ISD ChipCorder
ISD2360
Design Guide

Publication Release Date: Aug 3, 2016
Revision V1.15

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result in or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.

Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

Publication Release Date: Aug 3, 2016
Revision V1.15

Table of Contents-

List of Figures-

List of Tables-

1. General Description

Overview

The ISD2360 3-channel digital ChipCorder® provides single-chip storage and playback of high quality audio. The ISD2360 features digital de-compression, comprehensive memory management, Flash storage, an integrated audio signal path with up to 3 channel concurrent playback and a Class D speaker driver capable of delivering 1 W of power. The ISD2360 utilizes Flash memory, in 1-Kbyte sectors, to provide non-volatile audio playback for a single-chip audio playback solution for up to 64 seconds duration (based on 8 kHz/4-bit ADPCM compression). This eliminates the need for additional EEPROM/Flash devices.

The ISD2360 can be controlled and programmed through a Serial Peripheral Interface (SPI) or can operate in standalone mode by triggers applied to the device's six General Purpose Input/Output (GPIO) pins.

The ISD2360 includes an internal oscillator and requires no external clock sources or components other than a speaker to deliver quality audio prompts or sound effects to enhance user interfaces. The ISD2360 also provides higher sampling frequencies, improved Signal-to-Noise Ratios (SNR), reduced power consumption, fast programming time and integrated program verification.

Features

- Performance Enhancements
  - 3-channel mixing playback
  - GPIO parallel processing: supports dynamic change on GPIO output during playback
  - EMI, EFT, ESD improvements
  - 1 W power output at 5 V
- Duration—64 sec. based on 8 kHz/4-bit ADPCM in 2 Mbits of Flash storage (256 KB)
- Audio Management
  - Store pre-recorded audio (**Voice Prompts**) using high quality digital compression
  - Use simple index-based commands for playback – no address needed
  - Execute pre-programmed macro scripts (**Voice Macros**) designed to control the configuration of the device and playback Voice Prompts sequences
- Path and Playback Control
  - Audio streaming for up to 3 channels can be mixed and played back concurrently
  - Independent counters for each channel enable user micro-management on Voice Macro execution
  - Mask Jump for branch execution; based on internal register or external GPIO pin
- Control
  - Serial Peripheral Interface (SPI) for microprocessor control and programming
  - Standalone control when customized Voice Macro scripts are assigned to GPIO trigger pins
- Sample Rates
  - 8 sampling frequencies available: 4, 5.3, 6.4, 8, 10.67, 12.8, 16 and 32 kHz
  - Each Voice Prompt can have its own optimal sample rate
- De-Compression Algorithms
  - µ-Law: 6, 7 or 8 bits per sample.
  - Differential µ-Law: 6, 7 or 8 bits per sample
  - PCM: 8, 10 or 12 bits per sample

  - Enhanced ADPCM: 2, 3, 4 or 5 bits per sample
  - Variable bit-rate optimized compression allows best possible compression given a metric of Signal-to-Noise Ratios (SNR) and background noise levels
- Clock Source: Internal oscillator with internal reference, factory trimmed to ±1% deviation at room temperature
- Output
  - Pulse Wave Modulation (PWM): Class D speaker driver to direct drive an 8 Ω speaker or buzzer
  - Delivers 1W at 5V supply
  - Delivers 400mW at 3V supply
- I/Os
  - SPI interface: MISO, MOSI, SCLK, SSB for commands and digital audio data
  - 6 General Purpose I/O (GPIO) pins multiplexed with the SPI interface
- Flash Storage
  - 2 Mbits of storage for combined audio/data content
  - Fast programming time (20 µs/byte)
  - Erase sector size 1 Kbyte, sector erase time 2 ms
  - Integrated memory checksum calculation for fast verification
  - Endurance >100K cycles; retention > 10 years
- Operating Voltage: 2.4-5.5 V
- Packages, Green:
  - QFN 32-Lead
  - SOP 16-Lead 300 mil
- Temperature Range: Industrial: -40° C to 85° C

2. Pin Configurations

2.1 Pin Diagrams

The ISD2360 ChipCorder is available in QFN 32-Lead and SOP 16-Lead 300 mil packages, as shown in Figure 2-1 and Figure 2-2**.**

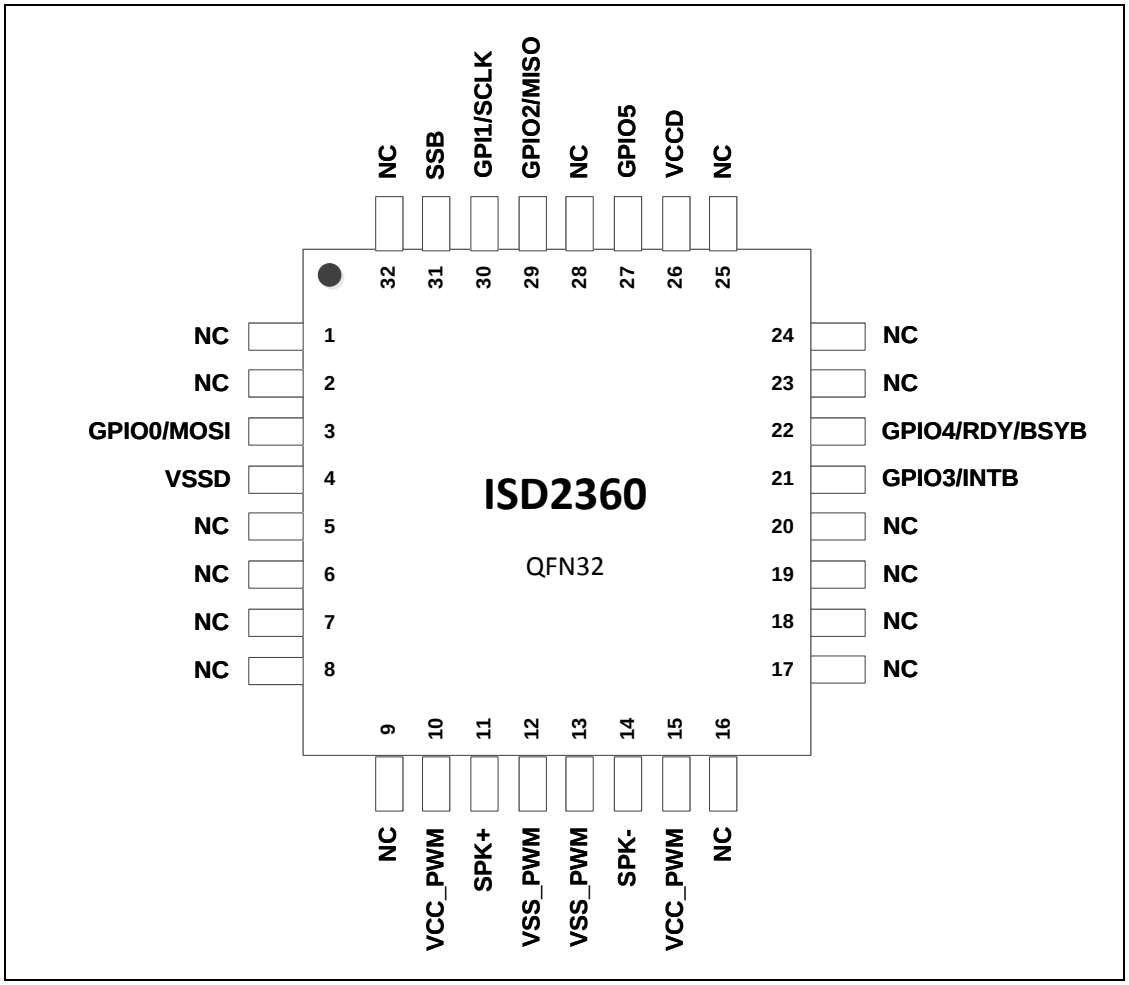


Figure 2-1 ISD2360 QFN 32-Lead Package

Note: The large center exposed pad under the QFN 32-Lead package should be connected to VSSD on the board to ensure good heat dissipation and mechanical stability.

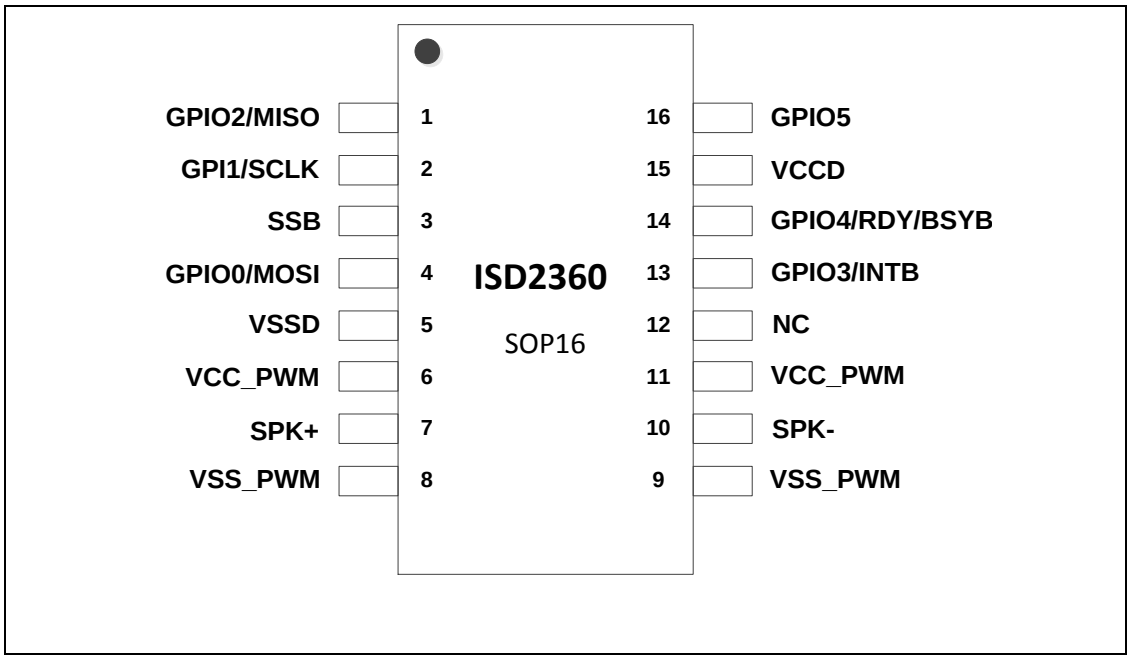


Figure 2-2 ISD2360 SOP 16-Lead 300 mil Package

2.2 Pin Descriptions

The pin descriptions for the ISD2360 in QFN 32-Lead package and in SOP 16-Lead 300 mil package are provided in Table 2-1 and Table 2-2, respectively.

Table 2-1 32-Lead QFN Pin Descriptions

Pin	Pin Name	I/O	Function
1	NC		This pin should remain Not Connected.
2	NC		This pin should remain Not Connected.
3	GPIO0/MOSI	I	Master-Out-Slave-In. Serial input to the ISD2360 from the host. Can be configured as a General Purpose I/O pin.
4	VSSD		Digital Ground.
5	NC		This pin should remain Not Connected.
6	NC		This pin should remain Not Connected.
7	NC		This pin should remain Not Connected.
8	NC		This pin should remain Not Connected
9	NC		This pin should remain Not Connected.
10	VCC_PWM	I	Digital Power for the PWM Driver. It can be from a separate power supply other than VCCD.
11	SPK+	O	PWM driver positive output. The SPK+ output and the SPK- pin provide a differential output to drive an 8 Ω speaker or buzzer. During power down, this pin is in tri-state.
12	VSS_PWM	I	Digital Ground for the PWM Driver.
13	VSS_PWM	I	Digital Ground for the PWM Driver.
14	SPK-	O	PWM driver negative output.

Pin	Pin Name	I/O	Function
			The SPK- output and the SPK+ pin provide a differential output to drive an 8 Ω speaker or buzzer. During power down, this pin is tri-state.
15	VCC_PWM	I	Digital Power for the PWM Driver. It can be from a separate power supply other than VCCD.
16	NC		This pin should remain Not Connected
17	NC		This pin should remain Not Connected
18	NC		This pin should remain Not Connected
19	NC		This pin should remain Not Connected
20	NC		This pin should remain Not Connected
21	GPIO3/INTB	O	Active low interrupt request pin. This pin is an open-drain output. Can be configured as a General Purpose I/O pin.
22	GPIO4/RDY/ BSYB	O	Output pin reports the status of data transfer on the SPI interface. “High” indicates the ISD2360 is ready to accept new SPI commands or data. Can be configured as a General Purpose I/O pin.
23	NC		This pin should remain Not Connected
24	NC		This pin should remain Not Connected
25	NC		This pin should remain Not Connected
26	VCCD	I	Digital Power. It can be from a separate power supply other than VCC_PWM.
27	GPIO5	I/O	General Purpose I/O pin
28	NC		This pin should remain Not Connected
29	GPIO2/MISO	O	Master-In-Slave-Out. Serial output from the ISD2360 to the host. This pin is in tri-state when SSB=1. Can be configured as a general purpose I/O pin.
30	GPI1/SCLK	I	Serial Clock input to the ISD2360 from the host. Can be configured as a General Purpose input pin.
31	SSB	I	Slave Select input to the ISD2360 from the host. When SSB is low, the device is selected and responds to commands on the SPI interface. When asserted, GPIO0/1/2 automatically configure to MOSI/SCLK and MISO respectively. SSB has an internal pull-up to VCCD.
32	NC		This pin should remain Not Connected.
-	center exposed pad		The center pad under the QFN 32-Lead package is connected to VSSD internally.

Table 2-2 SOP 16-Lead Pin Descriptions

Pin	Pin Name	I/O	Function
1	GPIO2/MISO	O	Master-In-Slave-Out. Serial output from the ISD2360 to the host. This pin is in tri-state when SSB=1. Can be configured as a general purpose I/O pin.
2	GPI1/SCLK	I	Serial Clock input to the ISD2360 from the host. Can be configured as a General Purpose input pin.
3	SSB	I	Slave Select input to the ISD2360 from the host. When SSB is low, the device is selected and responds to commands on the SPI interface. When asserted, GPIO0/1/2 automatically configure to MOSI/SCLK and MISO respectively. SSB has an internal pull-up to VCCD.
4	GPIO0/MOSI	I	Master-Out-Slave-In. Serial input to the ISD2360 from the host. Can be configured as a General Purpose I/O pin.
5	VSSD		Digital Ground.
6	VCC_PWM	I	Digital Power for the PWM Driver. It can be from a separate power supply other than VCCD.
7	SPK+	O	PWM driver positive output. The SPK+ output and the SPK- pin provide a differential output to drive an 8 Ω speaker or buzzer. During power down, this pin is in tri-state.
8	VSS_PWM	I	Digital Ground for the PWM Driver.
9	VSS_PWM	I	Digital Ground for the PWM Driver.
10	SPK-	O	PWM driver negative output. The SPK- output and the SPK+ pin provide a differential output to drive an 8 Ω speaker or buzzer. During power down, this pin is tri-state.
11	VCC_PWM	I	Digital Power for the PWM Driver. It can be from a separate power supply other than VCCD.
12	NC		This pin should remain Not Connected
13	GPIO3/INTB	O	Active low interrupt request pin. This pin is an open-drain output. Can be configured as a General Purpose I/O pin.
14	GPIO4/RDY/ BSYB	O	Output pin reports the status of data transfer on the SPI interface. “High” indicates the ISD2360 is ready to accept new SPI commands or data. Can be configured as a General Purpose I/O pin.
15	VCCD	I	Digital Power. It can be from a separate power supply other than VCC_PWM.
16	GPIO5	I/O	General Purpose I/O pin

3. Block Diagram

The major functional blocks of the ISD2360 ChipCorder, as shown in **Figure** 3-1 include:

- SPI and GPIO Interfaces
- Memory Management and Command Interpreters
- De-Compression
- Digital Signal Path Filtering, Mixing, Sampling and Volume Control
- PWM Control
- Internal Flash Memory
- Power Conditioning
- Flash Memory Controller

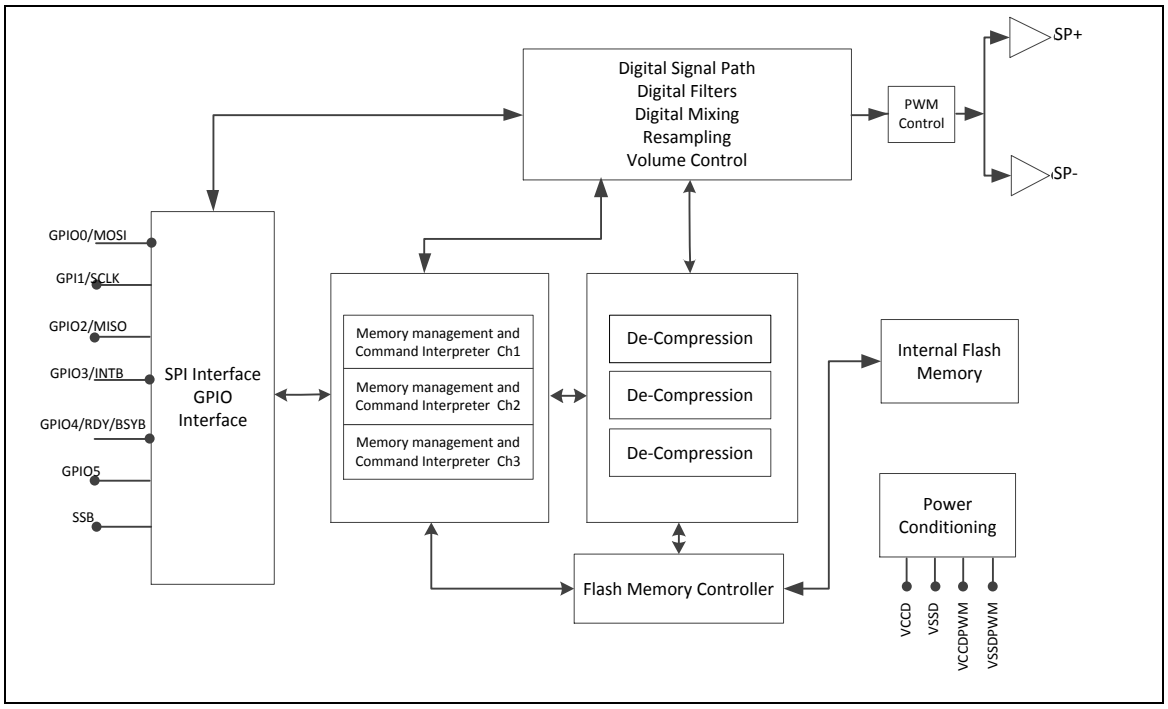


Figure 3-1 ISD2360 Block Diagram

The ISD2360 can be controlled and programmed through a Serial Peripheral Interface (SPI) or can operate in standalone mode by triggers applied to the device's six General Purpose Input/Output (GPIO) pins. Voice Prompt and Voice Macro commands facilitate fast programming.

The ISD2360 ChipCorder provides 2 Mbits of Flash storage for combined audio/data content. Memory is available in 1-Kbyte sectors, eliminating the need for additional serial EEPROM/Flash devices. The Flash memory provides non-volatile audio playback for a single-chip audio playback solution for up to 64 seconds duration (based on 8 kHz/4-bit ADPCM compression).

4. Device Status

The ISD2360 status can be acquired by querying the *Device Status Register* and the *Interrupt Status Register*. After receiving a READ_STATUS command, the ISD2360 ChipCorder continues to send back the device status byte and the interrupt status byte in turn, as long as the master provides the clock.

During an SPI transaction, the ISD2360 continually sends back its current status via MISO. When executing an SPI command or a Voice Macro command (refer to section 7.3 for details of Voice Macro) script, the ISD2360 continually updates its device status register bits. Upon completion of an SPI command or a Voice Macro command, the ISD2360 updates its interrupt status register bits.

4.1 Device Status Register

During an SPI transaction, for all commands except digital reading commands, the device status byte is sent back from device via MISO for every byte of data sent to the ISD2360.The details of the device status bits are shown in Table 4-1**.**

Table 4-1 Device Status Register Description

Status Byte							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
PD	DBUF_RDY	INT	-	CH2_BSY	CH1_BSY	CH0_BSY	DIG_BSY

The individual bits of the Device Status Register refer to the following conditions:

PD If this bit is set, the device is powered down. The DBUF_RDY bit will be low. When PD is high, only the READ_STATUS, READ_INT and PWR_UP commands are accepted. If any other command is sent, it is ignored and no interrupt for an error is generated.

DBUF_RDY In Power Down status, this bit is low indicating the device can only accept a PWR_UP (power up) command. When PD is low, this bit reflects the state of the RDY/BSY pin.

INT Indicates that an interrupt has been generated. The interrupt is cleared by the READ_INT command. The Interrupt type can be determined by the bits of the Interrupt Status Byte.

CHx_BSY When high, this bit indicates that channel [x] is in one of the following conditions:

- Processing a Voice Macro
- Processing a Voice Prompt
- Channel is waiting to process existing command in command buffer
- GPIO command is pending

Once set, Channel [x] will not respond to a new audio command until it returns low.

DIG_BUSY When high, this bit indicates that the Flash controller is still processing a digital memory access. For device erase commands, such as ERASE_MEM and CHIP_ERASE, the user can poll this bit to determine if the erasure is complete.

4.2 Device Interrupt Register

Whenever the ISD2360 generates an interrupt, the *Interrupt Status Register* holds flags that indicate the type of interrupt that was generated. The interrupt bits are shown in **Table** 4-2. These flags will remain set until a READ_INT command clears them and the hardware interrupt pin (INTB) is set. Some interrupts require further servicing to remove the condition generating the interrupt; for instance, a FIFO full or empty interrupts. If the condition is not serviced before a READ_INT, the device will immediately generate a new interrupt. To respond to an interrupt, use the following procedure:

- READ_STATUS to determine which interrupt flags are set.
- Service the interrupt appropriately.
- READ_INT to determine if a new interrupt has occurred during the service routines.
- If a new interrupt is detected, go to step 2. READ_INT will clear the interrupt status.

Table 4-2 Interrupt Status Register Description

Interrupt Status Byte							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
TALARM_INT	MPT_ERR	WR_FIN	CMD_ERR	OVF_ERR	CH2_FIN	CH1_FIN	CH0_FIN

The individual bits of the Interrupt Status Register refer to the following conditions:

INT An interrupt has been generated. The interrupt is cleared by the READ_INT command.

TALARM_INT Indicates that a temperature alarm has been set.

MPT_ERR Indicates a memory protection error. Digital access is attempted for protected memory.

WR_FIN Indicates a digital write command has finished writing to the Flash memory.

CMD_ERR An invalid command was sent to the device. The invalid command will be ignored because the command buffer was full, a Voice Macro has been active, or the device was not ready to respond to an erase command.

OVF_ERR This error is generated if the host illegally tries to read or write data while the RDY/BSYB pin is low. It is also generated if a digital read or write attempts to read or write past the end of memory.

CHx_FIN This bit indicates an interrupt was generated because a command finished executing on Channel X. A CHx_FIN interrupt will be generated each time a Voice Macro or Voice Prompt play finishes.

5. Device Configuration

The ISD2360 is configured by writing to a set of configuration registers. This can be accomplished either by sending an SPI command, such as WR_CFG_REG, or by executing Voice Macros that contain configuration commands.

All configuration registers are reset to their default values when there is a reset condition. When the ISD2360 is in Power Down Mode, a group of 3 V registers will retain their values while all other registers lose their content. Note the difference between Power Down and Power Off: Power Down means the device is entering the standby state with the PD bit set; Power Off means the device no longer has power

Refer to Section 9 *REGISTER OPERATIONS* for more detailed register information.

5.1 Device ID

The ISD2360 responds to an SPI READ_ID command by sending out its 4-byte identification data through the MISO line. Within these 4 bytes, the first byte is the ISD digital ChipCorder family ID which is 0x05 for ISD2360. The following three bytes are a JEDEC compliant code indicating the memory type; they are manufacturer ID, memory type ID and memory size byte with values of 0xEF, 0x20 and 0x60 respectively, for an ISD2360 device.

5.2 Clock Configuration

The ISD2360 has an internal oscillator trimmed at manufacturing that requires no external components to operate. This oscillator provides an internal clock source that operates the ISD2360 at a maximum audio sample rate F_{smax} of 32 kHz. The ISD2360 user should always use the clock from the internal oscillator with an internal reference that is the device default setting. By default, the ISD2360 device sets the clock register value as 0x00, which uses the internal oscillator with an internal reference. The user should not change the clock source.

5.3 GPIO Configuration

5.3.1 GPIO Pin Function Definition

The ISD2360 includes 6 GPIO pins. Each pin can be individually configured as an I/O pin with internal pull up/down capability, as an alternate function pin, or as a triggering pin. More specifically, the functional configuration of each GPIO pin may be one of the following four choices: an I/O pin, an alternate function pin, a falling-edge-triggering pin or a rising-and-falling-edge-triggering pin. Alternate function control registers, AF1 (0x1E) and AF0 (0x1F), define the specific function mode of each pin. The function mode of a GPIO pin is defined by AF1 bit0 and AF0 bit0, shown in Table 5-1**.** The configuration choices for each mode are identified in **Table** 5**-**2. The upper two bits of the AF1 and AF0 registers are don't care bits.

Table 5-1 AF1/AF0 Bit Combinations for GPIO Pin Function Mode

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Alternate Function Control 1 (0x1E)	-	-	GPIO5 Function Mode	GPIO4 Function Mode	GPIO3 Function Mode	GPIO2 Function Mode	GPI1 Function Mode	GPIO0 Function Mode
Alternate Function Control 0 (0x1F)	-	-						

Table 5-2 GPIO Pin Function Modes

AF1 Bit	AF0 Bit	Configuration
0	0	IO pin
0	1	Alternate Function
1	0	Falling-edge Trigger
1	1	Rising-and-falling edge Trigger

Notes: GPIO1 is permanently low to guarantee SPI operation. Do not use as an output. GPIO5 does not have an alternate function.

5.3.2 Special 3 V Registers

The ISD2360 device contains a group of special registers that can keep their value during power down. This special group includes registers in the range of 0x14-0x16 and 0x19-0x2F, and they are powered by an internal regulated 3V power supply.

During power down with 1µA typical standby current, the GPIO triggering configuration in those 3V registers is still in effect; therefore, the ISD2360 can continue detecting a GPIO trigger during power down.

5.3.3 GPIO Pin Structure

Once a GPIO pin is configured as an I/O pin, additional configuration of this I/O pin can be done by configuring the *Output Data Control Register*(0x19), *Output Enable Control Register* (0x1A), *Pull Enable Control Register*(0x1B), *Input Data Control Register*(0x1C) and *Pull Select Control Register*(0x1D). When the GPIO is selected for its alternate function, the DOUT and OE connections to the pin are driven by sources other than the DOUT and OE registers. The structure of the GPIO pads is shown in Figure 5-1.

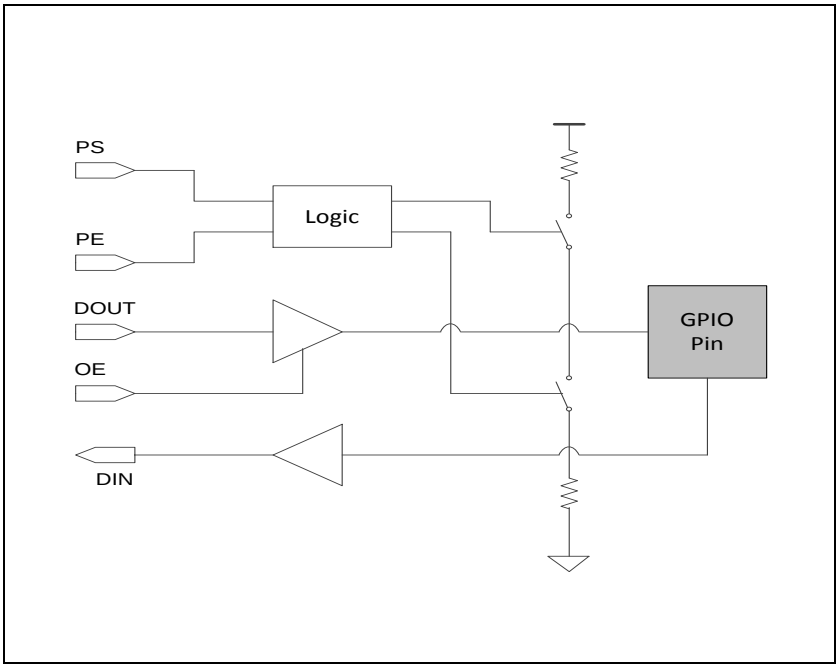


Figure 5-1 GPIO Pad Structure

Regardless of the configuration of GPIO pins 0,1 and 2, these pins turn into the SPI interface

whenever the SSB pin goes low and revert back to their former configurations when the SSB pin goes back to high. On the contrary, GPIO pins 3, 4 and 5 do not automatically change their configurations when the SSB pin goes low. Users must especially pay attention to the GPIO4 (RDY/BSYB) as RDY/BSYB pin is required when performing a digital-read or digital-write. This requires manual configuration of GPIO4 to the RDY/BSYB function before the digital operation is required. GPI03 can be configured as an INTB pin so the user can monitor this pin for task completion.

Transitions on the GPIO pins can generate a GPIO_INT interrupt. At the moment when the ISD2360 goes into a power down state, the status of the GPIO pins are latched. If the GPIO pin is configured as a triggering pin (interrupt enabled) and toggling on the pin is the valid triggering type indicated by AF1 and AF0, then when the toggling happens the ISD2360 device will execute a wake-up event.

5.4 Signal Path Configuration

The signal path involves filtering, sample rate conversion, volume control and decompression. A block diagram of the signal path is shown in **Figure** 5-2.The two PWM driver output pins SPK-and SPK+ provide a differential output to drive an 8 Ω speaker or buzzer. Note that during power down, these pins are in tri-state.

Pre-compressed audio signals transfer from memory or the SPI interface through the de-compressor block to the PWM driver or SPI out. The audio level is adjustable via VOLC before going out to the PWM driver path. As shown in **Figure** 5-2, the possible path combinations are:

- MEMORY → DECOMPRESS → SPKR (Playback to Speaker)
- MEMORY → DECOMPRESS → SPI_OUT (SPI Playback)
- SPI_IN → DECOMPRESS → SPKR (SPI Decode to Speaker)

To configure a signal path, such as a playback path from memory to the PWM output, the user should enable decompression and PWM (write 0x44 to register 0x02). Later, audio can be played through the PWM output by issuing a PLAY_VP command.

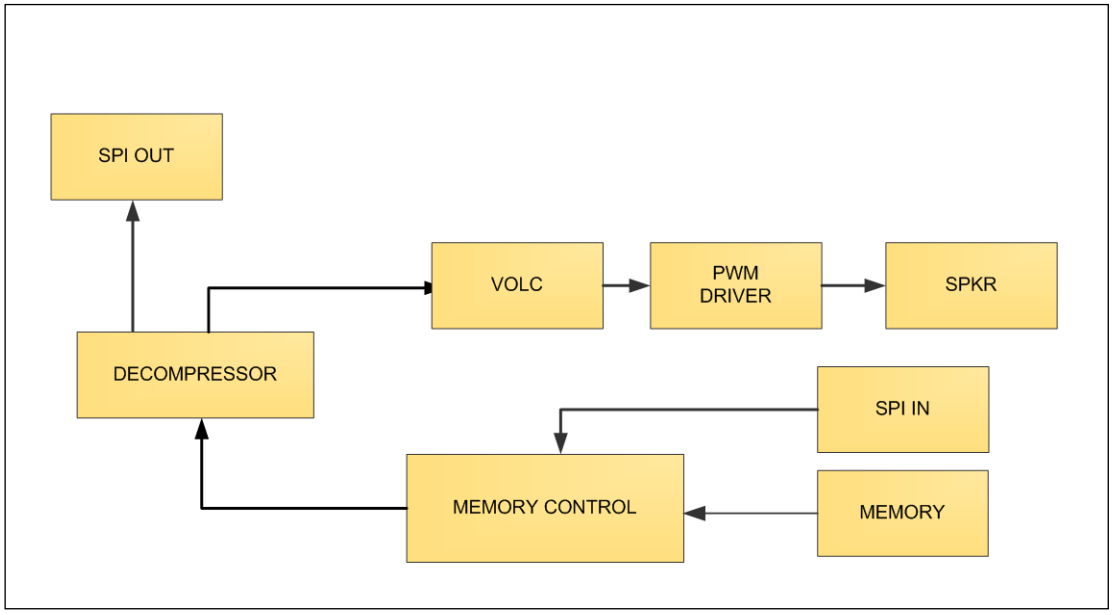


Figure 5-2 ISD2360 Signal Path

5.5 Device Checksum

The ISD2360 is able to calculate the hardware checksum from the beginning of memory to a specified end address. To start a checksum calculation, the user should first reset the circuit by writing one followed by a zero, to the RST_CHECKSUM bit in *Checksum Reset Register*, and then issue the SPI CHECKSUM command to initiate the calculation. The calculation is based on the Fletcher-32 algorithm, and the calculated checksum is stored in read-only *Checksum Register* 0x10-0x13, with the order of CHK_SUM1[7:0], CHK_SUM1[15:8], CHK_SUM2[7:0] and CHK_SUM2[15:8] respectively, as shown in Table 5-3.

Table 5-3 Checksum Register Result Data Storage

Address	Access Mode	Value At Reset	Nominal Value
0x10-0x13	R	0x0000	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x10	CHK_SUM1[7:0]							
0x11	CHK_SUM1[15:8]							
0x12	CHK_SUM2[7:0]							
0x13	CHK_SUM2[15:8]							

Note that the hardware Fletcher-32 algorithm calculation starts from the specified end address and continues backwards until the data address reaches 0. Therefore, the software checksum calculation should follow the same sequence fetching data. An example of the ISD2360 Fletcher-32 calculation routine is shown in Table 5-4:

Table 5-4 C Code Example Fletcher-32 Checksum Calculation

```
unsigned int fletcher32_RevrseCalculate(unsigned char *data, size_tlen)
{
	int temp;
	unsigned int sum1 = 0xffff, sum2 = 0xffff;

	while (len) {
		unsignedtlen = len> 360 ? 360 : len;
		len -= tlen;
		do {
			sum1 += *data;
			data--;
			sum2 += sum1;
		} while (--tlen);
		sum1 = (sum1 & 0xffff) + (sum1 >> 16);
		sum2 = (sum2 & 0xffff) + (sum2 >> 16);
	}
	// Second reduction step to reduce sums to 16 bits
	sum1 = (sum1 & 0xffff) + (sum1 >> 16);
	sum2 = (sum2 & 0xffff) + (sum2 >> 16);
	return sum2 << 16 | sum1;
}
```

5.6 Indirect Reference Registers

Eight 16-bit Indirect Reference Registers (R0-R7) can be used to store the following:

- Voice Prompt indexes for the SPI commands PLAY_VP@Rn and PLAY_VP_LP@Rn
- Voice Macro index for the SPI command EXE_VM@
- Voice Macro index associated with GPIO triggering

Table 5-5 shows the address locations of the R0-R7 registers.

Table 5-5 Indirect Reference Registers R0-R7

Access Mode	Value At Reset	Nominal Value
R/W	0x00xx	

0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
R0[7:0]	R0[15:8]	R1[7:0]	R1[15:8]	R2[7:0]	R2[15:8]	R3[7:0]	R3[15:8]
0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
R4[7:0]	R4[15:8]	R5[7:0]	R5[15:8]	R6[7:0]	R6[15:8]	R[7:0]	R1[15:8]

Normally when a GPIO pin is configured as a GPIO trigger pin, the entry index of its associated Voice Macro should be written into the corresponding Indirect Reference Register in POI Voice Macro, so the associated Voice Macro can be executed once the trigger happens. The association between a GPIO pin and its Indirect Reference Register is fixed by hardware, as shown in Table 5-6.

Table 5-6 GPIO Pin and Indirect Reference Register Association

	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	GPIO6	GPIO7
Associated Indirect Reference Register	R0	R1	R2	R3	R4	R5	R6	R7

5.7 GPIO4 Configuration for Digital Read/Write

The ISD2360 implements an internal FIFO with 4-byte depth which governs digital read/write operations flow throughput. For digital read operations, the host controller should only try to read from the SPI when the FIFO has data – when RDY/BSYB is high. For digital write operations, the host controller should only try to write into ISD2360 via SPI when the FIFO has vacant space – when RDY/BSYB is high.

The GPIO4 pin can be configured as RDY/BSYB pin to reflect this internal FIFO status. Although the host controller can always rely on polling the DBUF_RDY bit to achieve the flow control, utilizing a direct RDY/BSYB pin hardware feature provides a much easier and more efficient solution. To configure the GPIO4 pin to function as a dedicated RDY/BSYB pin, write AF1 bit4 as 0 and AF0 bit4 as 1.

Flow control through the RDY/BSYB pin, or through the DBUF_RDY bit using a pure software solution must be implemented for the following SPI digital operations:

DIG_READ Read from memory, for example.

DIG_WRITE Programming the ISD2360, for example.

SPI_PCM_READ De-compress the memory data, then read the de-compressed 16-bit PCM data from the SPI interface.

SPI_SND_DEC Send compressed data to ISD2360, allow the ISD2360 to de-compress the data and then play it out.

5.8 Fast De-Bounce for GPIO Trigger

The default value of FAST_DEB bit in the *De-bounce Time Control Register* is zero, which gives 20 ms de-bounce time for the ISD2360 GPIO trigger. Writing one to this bit enables the fast de-bounce feature, and the de-bounce time will be reduced to 8 ns. Fast de-bounce time should be used only in such situations in which fast speed is desired, and more importantly, the triggering signal is very clean without glitches – usually done by a MCU controlled I/O line. This allows a much faster and reliable IO controlled GPIO trigger to be achieved.

5.9 Thermal Shutdown

The ISD2360 device PWM driver can deliver a maximum output of nearly 1 Watt when operating at 5 V. With the increase of the delivered output power, the heat generated also increases. To prevent the accumulating heat from burning out the device, a thermal shutdown feature is implemented which is able to automatically shut down the PWM output when the temperature reaches the shutdown threshold.

The ISD2360 PWM output thermal shutdown feature is shown in **Figure** 5-3. When the device temperature reaches the threshold Talarm, the device automatically shuts down its PWM output while the rest of the device continues to operate. When the temperature of the ISD2360 drops to the level around (Talarm-10) C°, the PWM module resumes operation. For more details regarding the Talarm setting, refer to the *Thermal Control Register* description in Section 9 REGISTER OPERATIONS.

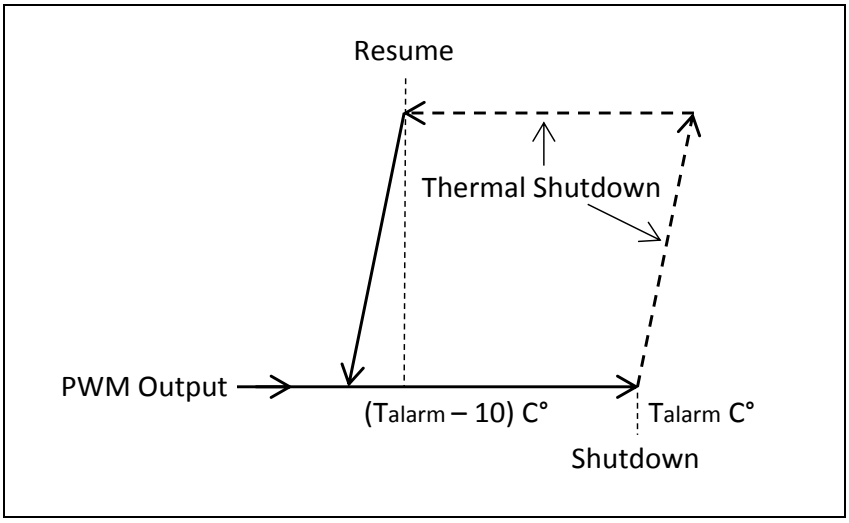


Figure 5-3 PWM Output Thermal Shutdown

6. Operational Description

6.1 Overview

In almost all cases, the ISD2360 needs to be programmed before being put into use in the field. A GUI software, *ISD-VPE2360* (Voice Prompt Editor for ISD2360 device) can be used to generate the application image file. Once the image file is ready, the user can choose to program the devices before populating them onto target systems, or to perform in-system programming in field.

In brief, the process for developing an ISD2360 application includes:

- Analyzing the application needs and decide on SPI or GPIO Trigger operation mode.
- Developing an *ISD-VPE2360* project; evaluate it on a demo system; and create the desired image file.
- Assembling the application target board using programmed devices, or using in-system programming in field.

The following sections describe the details involved in these procedures.

6.2 Audio Storage

ISD2360 has a built-in 2-Mbit flash memory, which consists of 256 sectors with a sector size of 1024-byte each; that in total can give 64 seconds playback duration, given the 8 kHz ADPCM4 compression algorithm to be used.

ISD2360 is a playback only device. For it to be functional, in most cases the ISD2360 device must first be programmed. Programming is done by using the SPI *DIG_WRITE* command, by writing an image file into the device Flash memory from the beginning of the Flash memory.

The GUI software, *ISD-VPE2360* enables users to create the application image file. Provided by Nuvoton Technology Corp., the *ISD-VPE2360* is available at no charge and can be downloaded from the Nuvoton website.

The *ISD-VPE2360* software generates the application image file based on the user's project. The image index/control data, called Memory Header, is stored from the first sector starting from address 0x00000. During power-on or power-up, the ISD2360 device automatically loads data from sector 0 for the initialization. Writing random data into the sector 0 has potential risk of causing the device to be unusable. Unless programming the device using the image file created by *ISD-VPE2360* Software, the user should avoid using sector 0.

For more details about memory organization, refer to Section *7 MEMORY MANAGEMENT.*

6.3 Sample Rates

The ISD2360 device is an audio de-compression device with a maximum supported sample rate at 32 KHz. The audio data compression is done by software, and the software should not use any compression method with a higher sample rate than 32 KHz.

By default (SRCFG=0), the ISD2360 generates the playback audio streaming according to the audio data header. To overturn the sample rate set by the audio data header and force the device to use the sample rate set by the *Sample Rate Overwrite Register*, the SRCFG bit in the *De-compres sion Control* **Register** needs to be set.

The forced sample rates available for playback are shown in Table 6-1

Table 6-1 Available Sample Rates

SR[2:0]	Ratio to F_{smax}	Sample Rate Fs (kHz)
0	8	4
1	6	5.44
2	5	6.4
3	4	8
4	2.5	12.8
5	2	16
6	1	32
7	3	10.67

6.4 Audio Compression and De-Compression

ISD2360 hardware performs only audio de-compression. Audio compression relies on the supplied software – ISD-VPE2360 or Voice Prompt Editor. This software takes a wave file as the input; converts it to Voice Prompt which is the basic usable audio element in a project. After the project image is burned into device, the audio sound effect can be re-produced by an SPI command, such as PLAY_VP.

When adding the wave file into a VPE project, the following algorithms are available:

- µ-Law: 6, 7 or 8 bits per sample.
- Differential µ-Law: 6, 7 or 8 bits per sample
- PCM: 8, 10 or 12 bits per sample
- Enhanced ADPCM: 2, 3, 4 or 5 bits per sample
- Variable-bit-rate optimized compression

Eight sampling frequencies (4, 5.3, 6.4, 8, 10.67, 12.8, 16 and 32 kHz) are available for each specified compression algorithm.

During playback, device hardware interprets the Voice Prompt header and de-compress the VP data accordingly. Unless overridden, ISD2360 hardware always plays the VP according to the sample rate specified in the VP header which is determined by the VPE project when adding wave files.

To override the setting in the VP header to enable play back of the audio streaming at a different sample rate, register 0x01 bit0 SRCFG must be set. Once the SRCFG bit is set, the device will play back audio streaming at the sample rate determined by register 0x00 bit5~7..

6.5 System Voice Macro Flow Chart

The ISD2360 reserves the first three Voice Macros for Power-On/Reset events, Power-UP initialization and GPIO triggering wake-up events with entry index 0x00, 0x01 and 0x02 respectively. The custom Voice Macro entry index starts from 0x03.

Whenever the ISD2360 detects a power-on reset condition, or receives a SPI RESET command, it begins a Power-On Initialization (POI) sequence and executes the POI Voice Macro, VM#0.

When the ISD2360 receives a Power Up command (PU) under power down state, it begins a power-up initialization (PU) sequence and executes PU Vocie Macro, VM#1. Note that if the device is already powered up, it will not execute PU Vocie Macro after receiving PU

command.

If the vectors for VM#0 or VM#1 do not exist in flash memory header, i.e. the POI Voice Macro and PU Voice Macro are not implemented, then a default routine for POI or PU is executed. The default sequence for POI is to power-down the ISD2360. The default PU sequence is to power up the device, assert the DBUF_RDY bit and clear the PD bit in status byte, and stay idle (powered up). **Figure** 6-1 shows the device initialization execution flow.

Once powered on, no matter if it is in PU or PD state, the ISD2360 device constantly detects edge transition on all GPIO pins. If an edge transition is detected and it meets the triggering condition defined by the AF1 and AF0 registers, then it a valid trigger occurs. If the edge transition does not match the triggering condition, or there is no trigger configuration defined at all, then the edge transition will be ignored.

When a valid trigger occurs, depending on if the trigger occurs in PD state, the ISD2360 device will or will not execute the Wakeup Voice Macro before executing the triggering pin associated Voice Macro, as shown in Figure 6-1. Please note that the Wakeup Voice Macro is shared by all triggering events so Wakeup Voice Macro can be used for common initialization when any triggering event awakes the device.

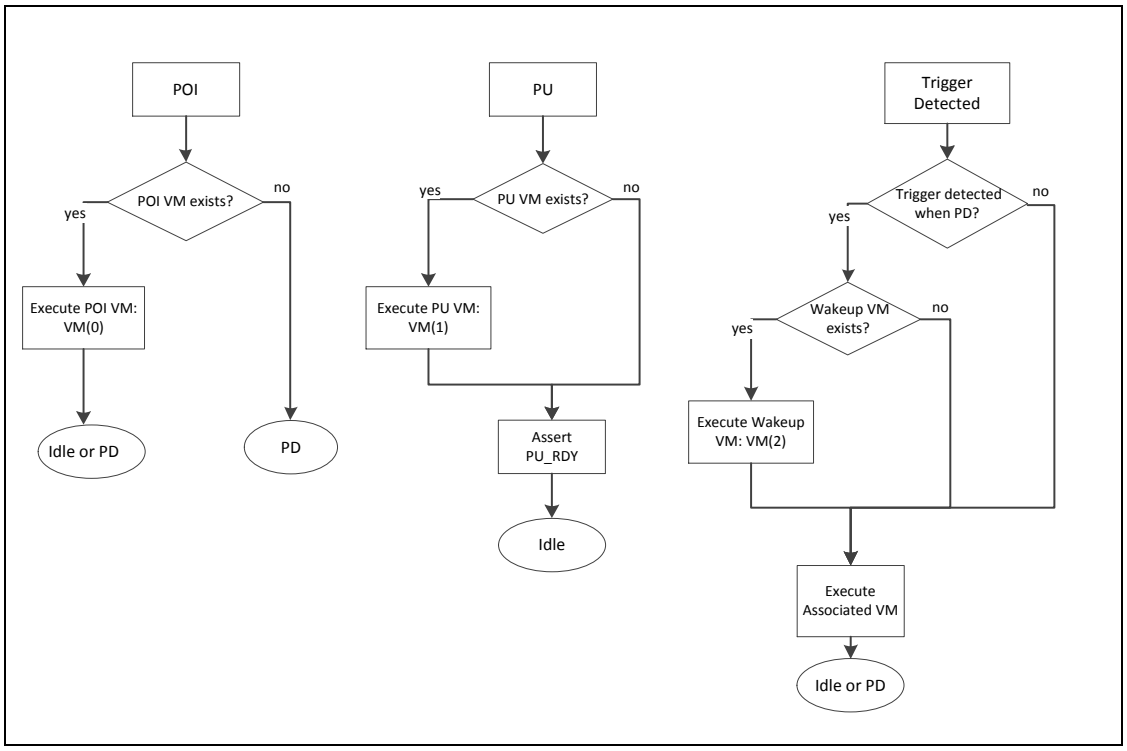


Figure 6-1 System Reserved Voice Macro Flow Chart

6.6 GPIO Trigger

6.6.1 GPIO Trigger Basics

Once power on, the ISD2360 continually detects the GPIO signal level transition. If an edge is detected, the ISD2360 hardware will further check if the edge matches the triggering condition defined in the *Alternate Function Control 1* and *Alternate Function Control 0* registers. If it matches, then this edge will cause a valid trigger and a sequence of operations will follow, otherwise the edge will be discarded.

Depends on whether the device was in PU or PD state when the trigger happened, the ISD2360 device operates differently. If the device was in PU state when the trigger happened, the device will only execute the GPIO pin associated Voice Macro; if the device was in PD state when the trigger happened, then device will first execute the Wakeup Voice Macro – no matter if it is implemented, then go to execute the GPIO pin associated Voice Macro. The Wakeup Voice Macro is a system reserved Voice Macro with fixed index 0x02; it always exists even if it is not implemented. If the Wakeup Voice Macro is not implemented, then the device will simply run default hardware initialization and power itself up, before executing the trigger associated Voice Macro.

The GPIO triggering pin to VM association is determined by hardware. That is: GPIO0 trigger will execute the VM whose index is in register R0, GPIO1 trigger will execute the VM whose index is in R1, and so on. If a trigger occurs but the associated VM index is not prepared in advance in the corresponding indirect reference register, the chip behavior is unknown.

During a triggering execution, the ISD2360 uses the associated VM index to locate the starting and end address of the Voice Macro script data in its memory, and then fetches the VM command data for execution.

To implement the GPIO triggering function, the user can follow the steps listed below; mostly the configuration is done in POI Voice Macro.

- Prepare the triggering condition:
  - Write *Output Enable Control Register* (0x1A) to configure the GPIO pin as an input pin.
  - Write *Output Enable Control Register* (0x1B) and *Pull Select Control Register* (0x1D). Configure the GPIO pin pull status according to the hardware feature. For example: if press-button pulls the GPIO low, and a falling edge trigger is desired, then the GPIO pin should be configured as an input with internal pulled high.
- Assign the channel for the Voice Macro:
  - Write *GPIO Trigger Channel Select 1* (0x14) and *GPIO Trigger Channel Select 2* (0x15) registers, so the associated VM can be assigned into the specified channel for execution.
- Configure the triggering mode:
  - Write *Alternate Function Control 1* (0x1E) and *Alternate Function Control 0* (0x1F) registers.
- Write the associated Voice Macro index into the Indirect Reference Register R0~R5:
  - Write CFG_REG 0x20~0x2B.
- Edit the associated VM in the ISD-VPE2360 GUI environment.

After creating and programming the application image into the device, , the ISD2360 device is ready for use in a triggering application.

6.6.2 SPI or GPIO Trigger

There are two ways to operate the ISD2360: via the SPI interface or through GPIO trigger. The SPI interface can be always available; whereas, GPIO trigger function mode is available only after the related registers are configured. The GPIO trigger is preemptive, where SPI operation is not preemptive.

6.6.2.1 *Pulling SSB low automatically claims the SPI interface*

Each of the six ISD2360 GPIO pins can be configured to work at the one of the following four function modes: GPIO pin, alternate function pin, falling-edge triggering pin or falling-and-rising-edge triggering pin. The GPIO pin function modes are defined by AF1 (0x1E) and AF0 (0x1F) registers.

Regardless of the AF1 and AF0 setting, the GPIO0/MOSI, GPI1/SCLK and GPIO2/MISO three pins will be automatically changed to alternate function mode so that SPI transactions can proceed whenever the SSB bar is pulled low. And these three pin's function modes will automatically recover to what are specified by the AF1 and AF0 registers once the SSB bar is pulled back to high. The mode change for these three pins triggered by SSB level changing is done by hardware and is transparent to the user.

6.6.2.2 *Pulling SSB Low does not affect the triggering capability*

Pulling SSB low has nothing to do with the triggering feature. In other words, if the SPI pins including MISO, MOSI and SCLK pins are configured as trigger pins, even after SSB is pulled low, edges on those pins can still trigger. Whether a pin can trigger or not solely depends on its setting in AF1 and AF0 registers.

6.6.2.3 *GPIO Trigger Execution is preemptive*

For SPI commands to be valid and action to be taken, certain conditions must be met. As a general rule, to ensure the success of an SPI operation, the user must check the device status before sending the SPI commands.

Unlike the SPI operation, in GPIO Trigger operation, the ISD2360 always executes the associated VM in the assigned channel, regardless the current status of the channel. That is, even if the current channel is busy playing a Voice Prompt or executing other Voice Macro commands, all the activities in the channel will be terminated and the device will start to execute the Voice Macro associated with the latest trigger.

6.6.2.4 *SPI vs. GPIO Triggering Reliability*

In most cases, especially in standalone applications, the setup of the GPIO triggering feature relies completely on the execution of the POI Voice Macro. In turn, having POI VM executed relies on the successful detection of the power-on reset condition, as in standalone applications, an SPI RESET command is not likely to be available.

In user applications, there is a risk that power supply glitches causing device POI VM not be executed and hence the device will stop responding to the GPIO trigger after power on. **Figure** 6-2 shows how power supply glitches might cause POR (Power On Reset) failure and therefore, cause the 3 V registers of the ISD2360 to lose their configuration.

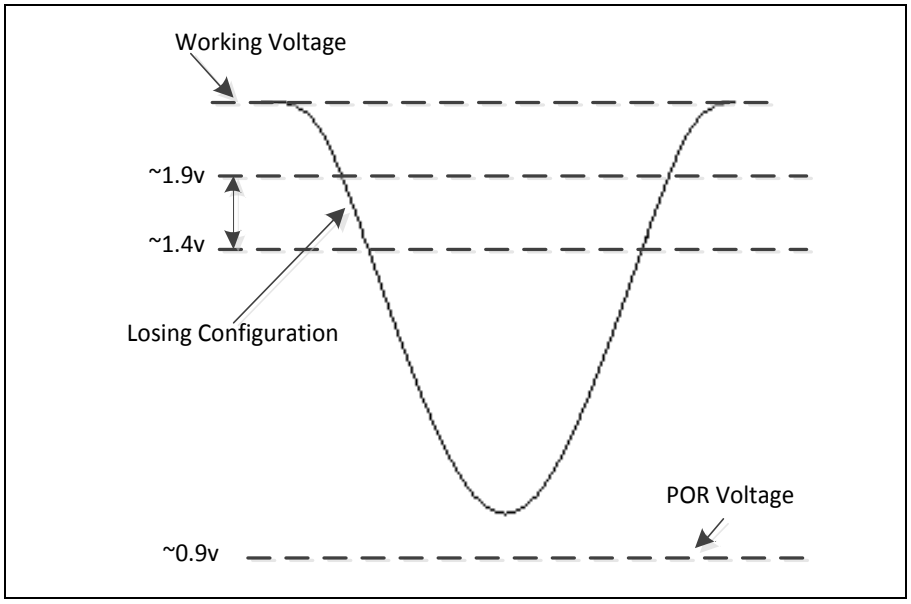


Figure 6-2 POR Reset Failure from Power Supply Glitches

When the power supply drops to ~1.9 V, the 3 V registers of the ISD2360 start to lose their values. If the power supply continues to drop to a level lower than the POR (Power On Reset) voltage, which is about 0.9 V, then once the power supply rises back to normal working voltage, POI VM will be triggered to run and the register settings can be recovered. However, if the power supply lowers to a level where it causes the device 3V registers to lose value, but not low enough (lower then ~0.9 V) to cause POR event, then once the power supply rise back to working voltage, the POI VM will not get triggered and hence the device lose desired triggering settings and may malfunction as a result. Once this has occurred, the only way to recover is to cycle the power again.

A clean power supply is very much desired in trigger application; because that can greatly reduce or even eliminate the chances for device losing its configurations.

On the other hand, the SPI interface is guaranteed to be responsive in any circumstances. For applications which require extreme reliability, the user should give SPI operation preferred consideration.

6.6.3 Volume Control via GPIO Trigger

By configuring the *Trigger Volume Control Register*, the user can use a GPIO Trigger to raise or lower the volume of the PWM output. GPIO trigger volume control provides 8 steps of attenuation, from maximum volume to maximum attenuation -63.75 dB. Note that the content of the *Volume Control Register* does not belong to the group of 3 V registers and the volume level is not retained at power down. When a GPIO trigger is configured as the volume up/down trigger, the ISD2360 device no longer executes its associated Voice Macro.

6.7 Multi-Channel Feature

The ISD2360 is a 3-channel device. The 3-channel mixer allows the user to mix audio data on any or all of the three channels. By first filtering and up-sampling the data from the individual channel(s) to an intermediate frequency of 64 kHz, the ISD2360 allows the user to mix audio data on the three channels with independent sample rates. Channel mixing is performed following the up-sampling and prior to volume control.

6.7.1 SPI Multi Channel

For audio playback operation started by an SPI command, the channel in which the playback is performed is determined by SPI_CMD_CH bits in the *Channel Control* Register. At any time, either one channel (0, 1, or 2) or all three channels may be in operation. Note that when all channels are selected, it is invalid to issue SPI play commands such as PLAY_VP or PLAY_VM. Otherwise, the command will be ignored and the CMD_ERR bit will be set. However, when all channels are selected, issuing the STOP command is a valid operation, and all activities in all three channels will be stopped.

To mix multiple channels for mixed playback, the user should first choose one channel and start the first playback, then select another channel to start the second playback. The two playbacks streaming will be automatically mixed and sent to the PWM output if enabled. For SPI play command, the device always checks whether the current channel is available – as reflected by the CHn_BSY bits in the Status Register. Issuing a SPI play command when the current channel is busy causes the play command to be ignored and the CMD_ERR bit to be asserted. The SPI STOP command stops the audio activity in the current channel, but has no effect on other channels. When all channels are selected, a STOP command stops activities in all three channels.

6.7.2 Multi Channel GPIO Trigger

A GPIO pin associated Voice Macro executes in its assigned channel. The two registers *GPIO Trigger Channel Select 1* and *GPIO Trigger Channel Select 2* specify the channel in which each of the six GPIO trigger-associated Voice Macros executes.

Multiple GPIO pins can be associated with the same VM; and multiple VMs can be assigned to the same channel. If two GPIO trigger Voice Macros are to be executed in the same channel, the later triggered playback will preempt the earlier triggered playback.

A GPIO triggering associated VM preempts only its own channel; the execution in other channels is not affected. In total, the ISD2360 device can have 3 channels running in parallel. If a trigger VM is assigned to “all channels”, i.e. 0x11 being written into the corresponding *GPIOn_TRIG_VH_SEL* bits, the result is that the VM will preempt all 3 channels when triggered; execution will begin in Channel 0. The status bit *CH0_BSY* will be set, which can in turn be preempted by another Channel 0 associated VM.

6.8 VM Jump and Channel Counter Commands

The ISD2360 has several special VM commands added to its VM command set. They are the Voice Macro Branch and Channel Counter commands. VM Branch commands include the unconditional jump (GOTO) and conditional jump (MASK_GOTO) commands. The Channel Counter uses the WAIT_CHN_CNT commands.

6.8.1 VM Branch Commands

GOTO is an unconditional absolute jump command. Similar to the absolute jump command commonly available to a microcontroller instruction set, it branches from the current execution and jumps to execute another valid VM command. The next to be executed VM command does not need to belong to the current VM from which the device jumps; it can be any valid VM command anywhere within the device image.

MASK_GOTO is a conditional jump. Refer to the *Mask Jump Control* register 0x08 definition; the jump condition includes JBSY and JGPIOn (0≤n≤5). Setting JBSY causes the VM execution to branch if the current channel holds an active playback operation; if it is not busy in the current channel, then the VM continues its execution. Setting JGPIOn bit(s) to one causes the device to determine if the respective pin(s) has an input high. If the input is high, the VM execution branches; otherwise it continues. Having more than one bit set to one in the *Mask Jump Control* register is allowed.

The address to which the branch command jumps must be a valid VM start address in memory. The GUI software ISD-VPE2360 provides the interface that allows the user to insert a label in front of VM commands; so instead of using an absolute memory address (which is very difficult to use because of memory address calculation), the user can choose to jump to a label. This provides the necessary means for the user to implement a branch command in an application project.

6.8.2 Channel Counter Command

It is sometimes desirable to have the capability to pause the VM execution for a certain amount of time. The WAIT_CHN_CNT command serves that purpose. When WAIT_CHN_CNT is executed, the device stops VM execution and waits until the current channel counter counts down to zero. After the counter reaches zero, the ISD2360 device continues the VM execution.

The *Chn_CNT (0≤n≤2)* register 0x0D-0F should be initialized before the execution of WAIT_CHN_CNT. The valid value for the channel counter is 0x01~0xFF; do not use the value 0x00. The pause time is (*Chn_CNT*+1)*12 ms.

7. Memory Management

The internal Flash memory of the ISD2360 provides 2 Mbits of storage space. Its 2, 097, 152 bits of memory are organized into 256-sectors of 1024-bytes each.

The ISD2360 is a playback-only device and usually must be pre-programmed before use. A GUI software *ISD-VPE2360*, provided by Nuvoton, can be used to generate the ISD2360 Flash image file. The binary image file should be programmed into the ISD2360 Flash memory from address 0x00.

After programming, the Flash memory can contain 4 types of data, as shown in **Figure** 7-1:

- Memory Header
- Voice Prompts
- Voice Macros
- User Data

Note that an ISD2360 application image always has a memory header and Voice Prompt data, but may not have Voice Macros and user data.

After being programmed, the *memory header* (which is stored from sector 0) contains the critical information needed for device power up. Therefore, the user should avoid writing random data into sector 0 during test, to prevent the potential deadlock that device can no longer be powered up because of bad data in sector 0. The user should always write sector 0 using only the image generated by *ISD-VPE2360*.

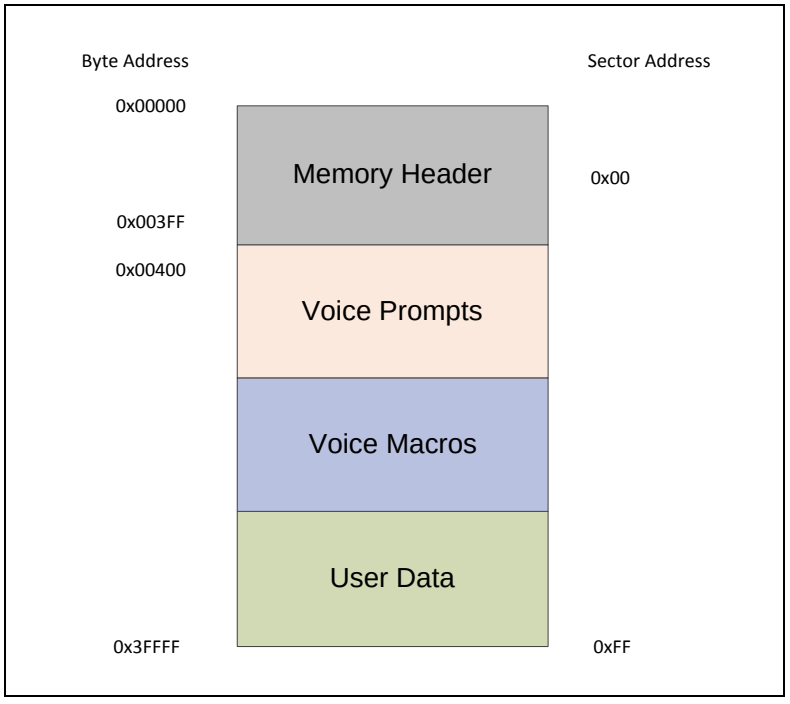


Figure 7-1 ISD2360 Memory Organization

7.1 Memory Header

7.1.1 Memory Format

The memory header is located from address 0x000000 and holds all the critical information about the memory organization upon which the device depends for all kinds of operations. **Table** 7-1 shows the typical memory data organization, including the memory header format, when there is no user data involved.

Table 7-1 Memory Data Sequence without Reserved User Data

Byte Address	Symbol	Description
0x00	MP_CFG	Memory protection scheme configuration byte.
0x01 – 0x02	-	Reserved.
0x03 – 0x04	PMP	Protected memory pointer.
0x05 – 0x0A	POI_VM	Start and end addresses of POI VM, 6 bytes in total.
0x0B – 0x10	PU_VM	Start and end addresses of PU VM, 6 bytes in total.
0x11 – 0x16	Wakeup_VM	Start and end addresses of Wake-up VM, 6 bytes in total.
0x17 – A1 = 0x17+6*VMs	Custom VM	Start and end addresses for all the custom VMs. In total 6*(number of custom VMs) bytes.
A1 – A2 = A1 + 6*VPs	VP	Start and end addresses for all the VPs. In total 6*(number of VPs) bytes.
A2 – A3	VP data	Holds all the VP data continuously.
A3 – A4	VM data	Holds all the VM data continuously.
A4 – end of the memory	Empty	Blank space, not used; all 0xFF.

Note: Byte order for PMP and VP/VM addresses conforms to the Little Endian convention.

The Memory Header contains at least 17 bytes located at the beginning of the memory space. Byte 0 determines the memory protection scheme. Bytes 3-4 store the PMP pointer. Byte 5 onwards is the Voice Macro/Voice Prompt index table. This table consists of six byte entries that are the start and end addresses of Voice Macros or Voice Prompts. Bytes 0x05-0x0A, bytes0x0B-0x10 and bytes 0x11-0x16 are reserved for the POI, PU and Wake-up Voice Macros, respectively. If the functions of these system-reserved Voice Macros are not implemented, these entries should be erased (i.e. 0xFFFFFF repeat six times).

After the system reserved VMs are the custom VMs index entries (starting from the 18th byte), then the VP index entries. After the VP index entries, the VP data and VM data are filled.

If there is no user data sector reserved, the data are to be filled continuously. In this case, memory space is occupied seamlessly without a single byte unused, from the beginning of the memory to the end of Voice Macro data (which also is the end of image data).

If there is user data, the VP data and VM data combine with user data and get their address assigned. For more details on data filling inside an image file when there is a reserved user data sector, please refer to Section 7.4 User Data.

7.1.2 Memory Protection

It may be desirable to protect portions of the internal memory from write/erase or interrogation (read) operations. The ISD2360 allows this by setting a Protection Memory Pointer (PMP) with which the user can protect an address range from the beginning of memory to the sector containing the PMP pointer. There are three types of protection; the type is set by bits in the memory header *MP_CFG* byte as shown in **Table** 7-2. Memory protection is activated on power-up of the ISD2360. Whenever the user changes the settings for memory protection, the new setting will take effect when the chip is reset.

Table 7-2 Memory Header MP_CFG Byte

Memory Header Byte							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
1	1	0	0	1	RP	WP	CEP

Chip Erase Protect Writing zero to the **CEP** bit enables chip erase protection. This prevents a mass erase function so that the CHIP_ERASE command cannot proceed. However, enabling the chip-erase function alone does not block the sector erase function.

Write Protect Writing zero to the **WP** bit enables write protection of the internal memory within the range pointed by the PMP. Write protection means that digital write or erase commands such as DIG_WRITE and MEM_ERASE will not function in this memory area. Note that **WP** protection does not block the chip erase protection.

Read Protect Writing zero to the **RP** bit enables the read protection of the internal memory within the range pointed to by the PMP. Read protection means that digital read or audio playback commands over the protected memory area are forbidden and the error status bit will be set. Use this to ensure that memory contents cannot be digitally copied or read.

Memory protection is activated on power-up of the ISD2360. Whenever the user changes the settings for memory protection, the new setting will take effect when the chip is reset. **Table** 7-3 shows the memory protection scheme.

Table 7-3 Memory Protection Scheme

MP_CFG Value	Bit Values	Protection Scheme
0xCF	CEP=1, WP=1, RP=1	No protection.
0xCE	CEP=0, WP=1, RP=1	Chip Erase Protection only; individual sectors can still be erased.
0xCD	CEP=1, WP=0, RP=1	Write Protection only; device still can be chip-erased.
0xCC	CEP=0, WP=0, RP=1	Chip Erase Protection plus Write Protection. Device image can no longer be changed.
0xCB	CEP=1, WP=1, RP=0	Read protection only.
0xCA	CEP=0, WP=1, RP=0	Read Protection plus Chip Erase Protection.
0xC9	CEP=1, WP=0, RP=0	Read Protection plus Write Protection.
0xC8	CEP=0, WP=0, RP=0	Read protection, Write protection, plus Chip Erase Protection.

7.2 Voice Prompt

Voice Prompt (VP) is a basic audio element that can be stored and played back by the ChipCorder. A Voice Prompt consists of:

- An index entry in the memory header, which points to the starting address of the compressed audio data of this prompt
- Compressed audio data, which are stored continuously

ISD-VPE software can be used to generate a VP from a wave file. Only wave files that are in the 16-bit PCM mono format should be used to generate Voice Prompts. Once generated, the Voice Prompts exist as part of the VPE project and thus part of the ISD2360 Flash image data. After the project image is burned into Flash memory and with appropriate configuration, a PLAY_VP command can be issued to playback the desired sound effect.

7.3 Voice Macro

Voice Macro (VM) is a command script that can be executed by the VM state machine of the ISD2360. Once it gets running, the command data in the script will be read from memory and executed by the device. For the ISD2360, VM can be called upon to run either by an SPI command or by a GPIO trigger event.

Similar to Voice Prompt, a Voice Macro consists of

- An index entry in the memory header, which points to the starting address of the Voice Macro command data.
- VM command data

For example: a VM with index 0x04 can have the following data stored in memory, starting from 0x1000:

0x1000: B8 02 44 B8 03 00 A6 00 06 12

The meaning behind this script is:

- Write 0x44 into register Reg0x02 to enable decoder and PWM path
- Write 0x00 into register Reg0x03 to maximize the volume
- Play VP with index 0x0006
- Power down the device after the VP playback finishes

Voice Macro is a powerful feature of the ISD2360 digital ChipCorder. VM functionality ranges from simple tasks such as register configuration, playing a VP or playing a VM to more complicated tasks, such as playing a sentence.

Note that for applications that need to utilize the GPIO triggering feature, POI VM must be implemented so the triggering condition can be ready after system power on.

7.3.1 Voice Macro Commands

A Voice Macro (VM) consists of a sequence of Voice Macro commands. Unlike the SPI commands, a Voice Macro command must be stored in the Flash memory as a part(s) of a Voice Macro command script. During execution, the ISD2360 VM state machine fetches the command data from memory and then executes the instructions sequentially. Many SPI commands can be included in a VM. Besides There are also special commands, such as FINISH, WAIT_CNT, BRANCH, etc., which are only available in a VM. When editing a Voice Macro in an ISD-VPE2360 project, the user can rely on the GUI interface to fill in the Voice Macro commands. The length of most VM commands must be in multiples of 3, except for some one-byte command such as PD, FINISH etc. For example, for the SPI command “PLAY_VP@R2” whose SPI hex code is “0xAE 0x02”, its VM command counterpart has hex code “0xAE 0x02 0x00”. The tailing 0x00 is the dummy byte for VM command fulfilling the multiple-of-three rule. Table 7-4 provides the Voice Macro commands and the byte index.

Table 7-4 Voice Macro Commands

VM Command	Command Byte Index					
	1	2	3	4	5	6
PWR_DN	0x12	-	-			
SILENCE	0xA8	Data	0x00*			
Play_VP	0xA6	Index[15:8]	Index[7:0]			
Play_VP@Rn	0xAE	*n* = 0 …7	0x00*			
Play_VP_LoopN	0xA4	Index[15:8]	Index[7:0]	N[15:8]	N[7:0]	0x00*
Play_VP_LoopN@Rn	0xB2	*n* = 0 …7	0x00*	N[15:8]	N[7:0]	0x00*
EXE_VM	0xB0	Index[15:8]	Index[7:0]			
EXE_VM@Rn	0xBC	*n* = 0 …7	0x00*			
WR_CFG_REG	0xB8	Reg_Addr	Data			
MASK_GOTO	0xE0	Addr[23:16]	Addr[15:8]	Addr[7:0]	0x00*	0x00*
GOTO	0xE1	Addr[23:16]	Addr[15:8]	Addr[7:0]	0x00*	0x00*
WAIT_CHN_CNT	0xEE	0x00*	0x00*			
FINISH	0xFF	-	-			

Note: Shaded areas with asterisk (*) represent dummy bytes.

Descriptions of the Voice Macro commands shown in Table 7-4 include:

PWR_DN Power down the device.

SILENCE (n) Play silence for the duration of 32*n ms, 0≤n≤0xFF.

PLAY_VP (i) Play VP with index i, 0≤ i ≤0xFFFF.

PLAY_VP@(Rn) Indirect Play VP whose index in currently in register Rn, 0≤n≤0x07.

PLAY_VP_LP (i, cnt) Loop Play cnt times the VP with index i, 0≤n≤0xFFFF, 0≤cnt≤0xFFFF.

PLAY_VP_LP@ (Rn, cnt) Indirect Loop Play cnt times the VP whose index is currently in Rn, 0≤n≤0x07, 0≤n≤0xFF.

EXE_VM (i) Execute Voice Macro with index i, 0≤ i ≤0xFFFF.

EXE_VM@ (Rn) Indirect Execute of the VM whose index is currently in register Rn, 0≤n≤0x07.

WR_CFG_REG (reg n) Set configuration register reg to value n, reg←register address, 0≤n≤0xFF.

MASK_GOTO (addr) Conditional branch to memory address addr, 0≤addr≤0xFFFFFF.

GOTO(n) Absolute jump to memory address addr, 0≤addr≤0xFFFFFF.

WAIT_CHN_CNT Pause VM execution until the current channel counter completes counting down to 0.

FINISH Finish the Voice Macro and exit.

The GUI software ISD-VPE2360 provided by Nuvoton can be used to generate the VM command scripts for a project. The user can rely on this GUI software to implement all necessary VMs when creating the device image file needed for pre-programming before the device is put into use.

7.3.2 System-Reserved Voice Macros

The ISD2360 has three system-reserved Voice Macros: Power On Initialization (POI VM) Power Up (PU VM) and Wakeup VM with index values of 0x00, 0x01 and 0x02, respectively. ISD2360 VPE automatically reserves space for these system-reserved VMs when creating a project. Custom VM is manually created and starts from index 0x03. If a system-reserved VM is not implemented in a project, its 3-byte VM data starting address and end address will all be 0xFFFFFF, which indicates an empty VM.

All VMs can be executed by sending the SPI PLAY_VM command or by a GPIO trigger. Certain system conditions can also trigger these VMs to execute.

POI VM Power-On Initialization VM (Index 0x00) executes after the device power on initiates, or after the device receives an SPI Reset command. In most cases, devices initialization is achieved in the POI VM. For standalone applications that utilize the GPIO triggering feature, the POI VM must be implemented to configure the operation mode of the GPIO pins, the VM channel assignment, the VM to GPIO pin association, and the assignment of the triggering pin, etc.

PU VM Power-Up VM executes only after the device receives a Power-Up command.

Wakeup VM Wakeup VM executes after the device detects a valid trigger and wakes up from Power-Down state.

For information about system-reserved VM execution flow, refer to Section 6.5 System Voice Macro Flow Chart**.**

7.3.3 Sample Project Voice Macros

This section describes sample projects to illustrate the use of Voice Macros to setup GPIO triggers, multiple-channel mixing and the multiple-channel counter with mask branch to achieve GPIO PWM control. The user should study the simplest project first because basic VM settings explained in simple projects will not be described in the more advanced projects.

Note: All sample projects can be found in the ISD-VPE2360 installation folder “C:\Program Files\ISD-VPE2360” after installation.

7.3.3.1 *Sample_1: A Simple Trigger-To-Play Project*

Goal: Create a GPIO trigger-to-play project so that device will play “one” when the GPIO0 button is pressed, play “two” when the GPIO1 button is pressed, and so on.

How to configure: Create a falling edge when a button is pressed, and let the ISD2360 respond to the button-press event by executing the GPIO trigger associated Voice Macro. Configure the device POI VM, Wakeup VM and associated VMs as follows:

- Configure the GPIO trigger pins as input pins (*Output Enable Control Register*), with pull-high (*Pull Select Control Register*) and pull-enable (*Output Enable Control Register*) select all enabled. So, initially the GPIO pin stays high and a button-press creates a falling edge.
- Write register *GPIO Trigger Channel Select 1* and/or GPIO *Trigger Channel Select 2* to assign audio channel for each GPIO trigger.
- Assign VM index to the GPIO pin to setup the association – the VM address is to be stored in the dedicated *R0_L* for each pin.
- Enable Falling Edge trigger mode by writing one to *Alternate Function Control 1* register bits and zero to *Alternate Function Control 0* register bits.
- In Wakeup VM, enable Decoder and PWM Output by writing to the *Path Control Register***Path Control RegisterPath Control RegisterPath Control RegisterPath Control RegisterPath Control RegisterPath Control Register.**

- In the associated VM, simply play a VP then power down.

See **Figure 7-2** for detailed configuration information.

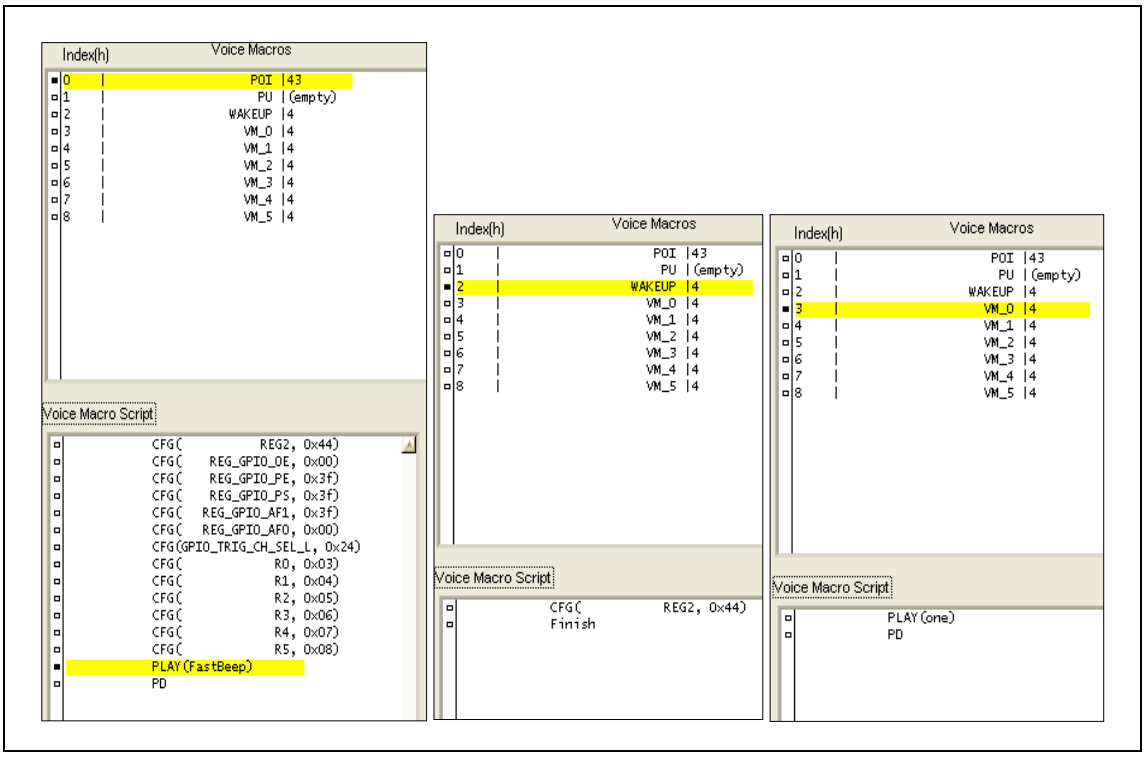


Figure 7-2 Sample_Project_1 Configuration

The following exlpains the POI VM:

CFG(REG2, 0x44) → write 0x44 into reg0x02; enable Decoder and PWM output path;
CFG(REG_GPIO_OE, 0x00) → write 0x00 into reg0x1A; configure all GPIO pins as input pins;
CFG(REG_GPIO_PE, 0x3F) → write 0x3F into reg0x1B; enable Pull-Enable for GPIO0-5;
CFG(REG_GPIO_PS, 0x3F) → write 0x3F into reg0x1D; enable Pull-High for GPIO0-5;
CFG(REG_GPIO_AF1, 0x3F) →write 0x3F into reg0x1E;
CFG(REG_GPIO_AF0, 0x00) →write 0x00 into reg0x1F; combined with reg0x1E to confiure GPIO0-5 as falling edge triggering pins.
CFG(GPIO_TRIG_CH_SEL_L, 0x24)→ write 0x24 into reg0x14, assign channel 0 to VM associated with GPIO0;
assign channel 1 to VM associated with GPIO1;
assign channel 2 to VM associated with GPIO2;
assign channel 0 to VM associated with GPIO3;
CFG(R0, 0x03) → write 0x03 into reg0x20; associate VM0x03 with GPIO0;
CFG(R1, 0x04) → write 0x04 into reg0x22; associate VM0x04 with GPIO1;
CFG(R2, 0x05) → write 0x05 into reg0x24; associate VM0x05 with GPIO2;
CFG(R3, 0x06) → write 0x06 into reg0x26; associate VM0x06 with GPIO3;
CFG(R4, 0x07) → write 0x07 into reg0x28; associate VM0x07 with GPIO4;
CFG(R5, 0x08) → write 0x08 into reg0x2A; associate VM0x08 with GPIO5;
PLAY(FastBeep) → Play VP “Fastbeep”
PD → power down device (to save power).

In POI VM, configuration to the *Output Enable Control Register*, *Pull Select Control Register* and *Output Enable Control Register* can be omitted because of their POR (Power-On-Reset) default values. Putting them there emphasizes that if button pressing pulls the GPIO pins low, then the initial pin status needs to be configured as input, pull enabled and pulled high to ensure a falling edge to be correctly generated.

Also note that since there is no assignment to the *GPIO Trigger Channel Select 2* register, the default channel to be assigned to GPIO4 and GPIO5 is Channel 0.

Once executed, at the end of POI VM, the device will play a sound effect “FastBeep” then power down. As long as there is power applied, all 3 V registers configured in this VM will retain their values except the *Path Control Register*. This enables the device to continue detecting a falling edge during PD. Once a falling edge is detected, the device will wake up and try to run the wakeup VM. In the Wakeup VM, common initialization for all trigger events can be done.

The following exlpains the Wakeup VM 0x02:

CFG(REG2, 0x44) → write 0x44 into reg0x02; enable Decoder and PWM output path;
Finish → device stays power on and exit

In power down mode, the *Path Control Register*. loses its value and the PWM path is disabled. Writing the value 0x44 into the *Path Control Register* enables the playback path for all GPIO triggers, since the Wakeup VM is shared by all triggering events. Note that the Wakeup VM should never end with a PD VM command; otherwise, the device will never be able to wake up.

The following explains the custom VM 0x03:

PLAY(one) → Play VP “one”
PD → power down device (to save power).

After system power on, i.e. after POI VM has executed, pressing the GPIO0 button plays “one” then the device will power down. Note that in this VM, it does not configure output path because the Wakeup VM has already done so.

7.3.3.2 *Sample_2: Channel Mixing with Volume Control*

Goal: Create a GPIO trigger-to-play project which can start and stop the playing of “Do ---“, “Re ---“, “Mi ---“ in Channel 0, Channel 1 and Channel 2, respectively, such that the mixed sound effect can be heard when all three channels play at the same time and also the volume can be increased or decreased..

Pressing the GPIO0 button once starts loop play of a tone “Do ----” in Channel 0; pressing twice stops the play. Repeating this “play-stop-play-stop …” pattern with the GPIO1 trigger for Channel 1 with the sound effect “Re ----”, and the GPIO2 trigger for Channel 2 with the sound effect “Mi ----” allows the mixed sound effect to be heard when multiple channels play at the same time. Pressing the GPIO4 button increases the volume in 8 steps. Pressing the GPIO5 button decreases the volume in 8 steps.

How to configure: Configure the device as follows:

- Configure GPIO0, 1, 2, 4 and 5 as falling-edge trigger pins.
- Assign Channel 0, 1, and 2 to GPIO0, 1 and 2 triggering respectively.
- Assign GPIO4 as the volume up trigger control pin.
- Assign GPIO5 as the volume down trigger control pin.

For each of the VM scripts associated with GPIO0, GPIO1 and GP102, making the change to the associated VM address for the next trigger causes the loop to use the pattern: play→ stop → play → stop.

See **Figure** 7-3 for detailed VM configuration information.

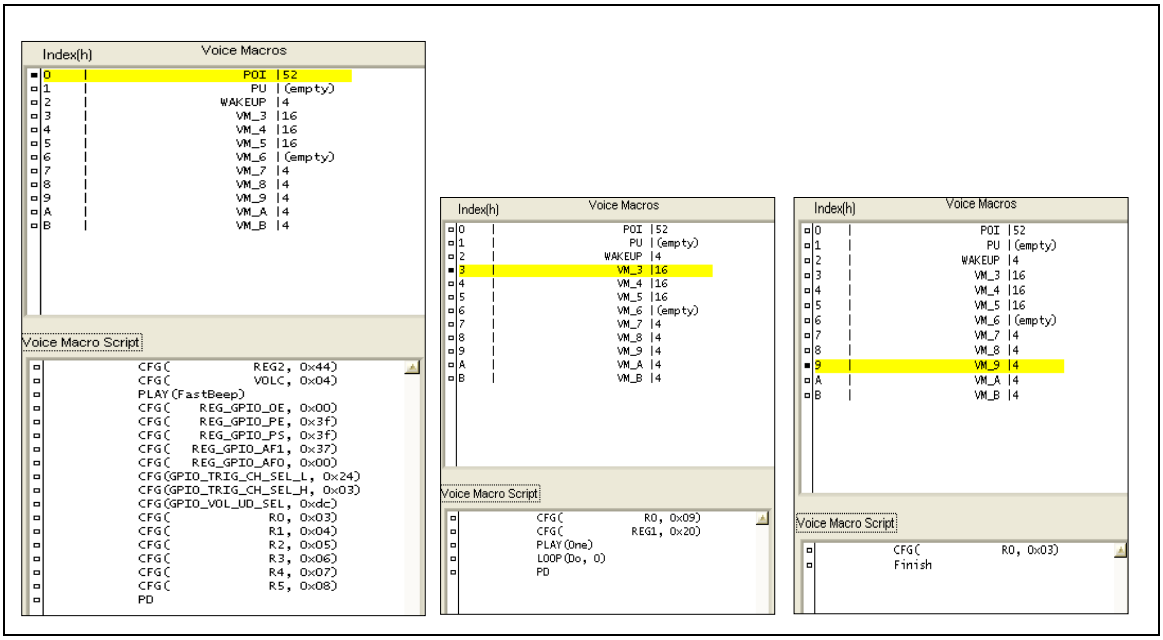


Figure 7-3 Sample_Project_2 Configuration

The following exlpains the POI VM:

CFG(REG2, 0x44) → write 0x44 into reg0x02; enable Decoder and PWM output path;
CFG(VOLC, 0x04) → write 0x04 into reg0x03; set attenuation at 4*(-0.25)dB value;
PLAY(FastBeep) → Play VP "Fastbeep"
CFG(REG_GPIO_OE, 0x00) → write 0x00 into reg0x1A; configure all GPIO pins as input pins;
CFG(REG_GPIO_PE, 0x3F) → write 0x3F into reg0x1B; enable Pull-Enable for GPIO0-5;
CFG(REG_GPIO_PS, 0x3F) → write 0x3F into reg0x1D; enable Pull-High for GPIO0-5;
CFG(REG_GPIO_AF1, 0x37) → write 0x37 into reg0x1E;
CFG(REG_GPIO_AF0, 0x00) →write 0x00 into reg0x1F; combined with reg0x1E to confiure GPIO0,1,2, 4 and 5 as falling edge triggering pins.
CFG(GPIO_TRIG_CH_SEL_L, 0x24)→ write 0x24 into reg0x14, assign channel 0 to VM associated with GPIO0; assign channel 1 to VM associated with GPIO1; assign channel 2 to VM associated with GPIO2; assign channel 0 to VM associated with GPIO3;
CFG(GPIO_TRIG_CH_SEL_H, 0x03)→ write 0x03 into reg0x15, assign all channels to VM associated with GPIO4; assign channel 0 to VM associated with GPIO5;
CFG(GPIO_VOL_UD_SEL, 0x03)→ write 0x03 into reg0x16, GPIO4 Trigger to volume up; GPIO5 trigger to volume down;
CFG(R0, 0x03) → write 0x03 into reg0x20; associate VM0x03 with GPIO0;
CFG(R1, 0x04) → write 0x04 into reg0x22; associate VM0x04 with GPIO1;
CFG(R2, 0x05) → write 0x05 into reg0x24; associate VM0x05 with GPIO2;
CFG(R3, 0x06) → write 0x06 into reg0x26; associate VM0x06 with GPIO3;
CFG(R4, 0x07) → write 0x07 into reg0x28; associate VM0x07 with GPIO4;
CFG(R5, 0x08) → write 0x08 into reg0x2A; associate VM0x08 with GPIO5;
PD → power down device (to save power).

As in Sample_Project_1, the initial status and trigger conditions are set in Sample_Project_2. Other functions are set in Sample_Project_2 as well: Input mode is set with internal pull high by the *Output Enable Control Register*, the *Pull Select Control Register* and the *Pull Enable Control Register*; GPIO pin trigger mode is set by the *Alternate Function Control 1* and *Alternate Function Control 0* registers; assignment of the trigger channel(s) is set by the *GPIO Trigger Channel Select 1* and *GPIO Trigger Channel Select 2* registers; and assignment of the associated VM indexes is performed.

POI VM in Sample_Project_2 also configures the *Trigger Volume Control Register* by assigning GPIO4 as the volume up trigger pin, and GPIO5 as the volume down trigger pin.

Note that trigger volume control overwrites the trigger play function. For example, in this sample project, because the GPIO4 and GPIO5 are configured as volume control pins, even if they have their associated VM assigned, the associated VMs are not to be executed after the trigger. The volume control applies to the audio streaming after mixer. There is no control of each individual channel.

The Wakeup VM 0x02 enables the playback path for all triggers.

The following explains the custom VM 0x03:

```
CFG(        R0, 0x09) → write 0x09 into reg0x20; so next time GPIO0 trigger executes VM 0x09;
CFG(      REG1, 0x20) → write 0x44 into reg0x02; No audo ramp down during loop play,
                        Thus prevents the clicks sound between each repeating play;
PLAY(one)             → Play VP “one”
Loop (      Do,   0) → Loop play VP “Do –” infinitely, in channel 0;
PD                    → power down device (to save power).
```

The following explains the custom VM 0x09:

```
CFG(        R0, 0x03) → write 0x03 into reg0x20; so next time GPIO0 trigger executes VM 0x03;
PD                    → power down device (to save power).
```

Because of the preemptive nature of the GPIO trigger, VM 0x09 stops the play in its channel which is channel 0. Then it reassigns the GPIO0 associated VM back to 0x03; thus, the play pattern of “play – stop – play – stop ….” is implemented.

7.3.3.3 *Sample_3: Driving GPIO Using the Channel Counter*

Goal: Create a GPIO trigger-to-play project which can start and stop the playing of a sound effect, increase and decrease the volume of the sound and turn an LED on and off.

Pressing the GPI1 button once starts loop play of a sound effect; pressing the button twice stops the play. This “play-stop-play-stop” pattern repeats. During play, the ISD2360 also drives the GP104 and GP105 LED on and off, as determined by Channel Counter 0. The GPI1 trigger play pattern is replicated on GP102, the volume is increase by pressing the GP100 button and the volume is decreased by pressing the HP104 button.

How to configure: Configure the device as follows:

- Configure GPIO0, 1, 2, and 3 as falling-edge trigger pins.
- Configure GPIO4 and GPIO5 as output pins.
- Assign Channel 0, 1, 2 and 0 to GPIO0, 1, 2 and 3 triggering, respectively.
- Assign GPIO0 as the volume up trigger control and GPIO3 as the volume down trigger control pin.
- Configure GPIO associated VM to implement “play – stop – play – stop …” pattern.

See Figure **7**-**4** for detailed VM configuration information.

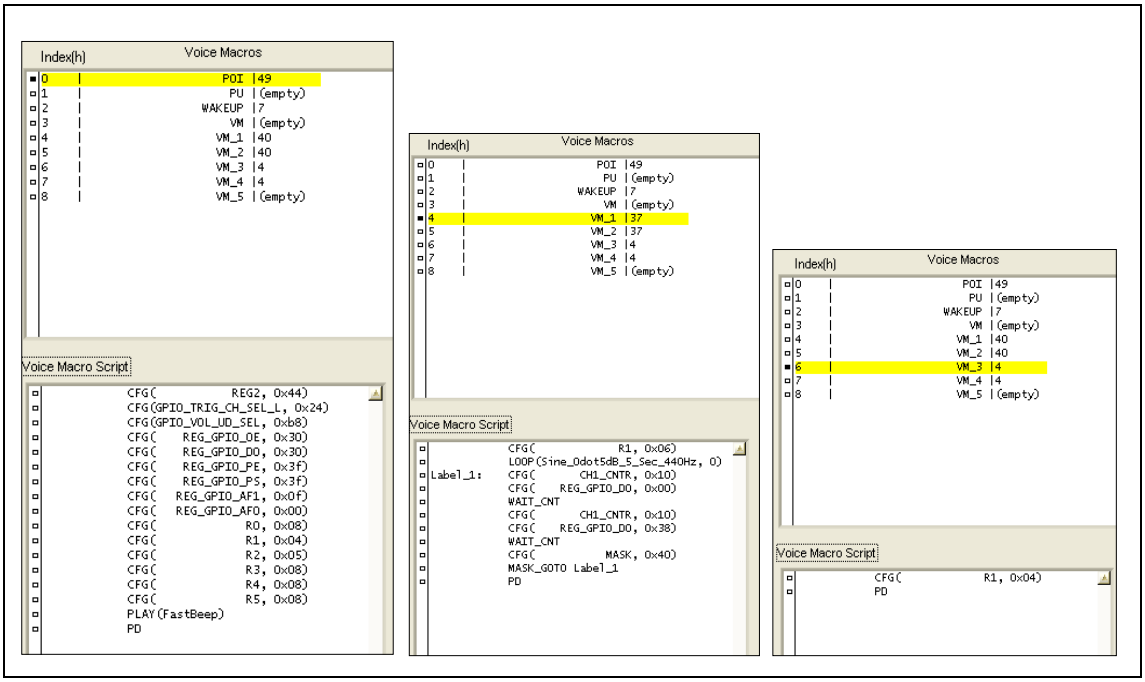


Figure 7-4 Sample_Project_3 Configuration

The following exlpains the POI VM:

CFG(REG2, 0x44) → write 0x44 into reg0x02; enable Decoder and PWM output path;
CFG(GPIO_TRIG_CH_SEL_L, 0x24)→ write 0x24 into reg0x14, assign channel 0 to VM associated with GPIO0;
assign channel 1 to VM associated with GPIO1;
assign channel 2 to VM associated with GPIO2;
assign channel 0 to VM associated with GPIO3;
CFG(GPIO_VOL_UD_SEL, 0xb8)→ write 0xb8 into reg0x16, GPIO0 Trigger to volume up;
GPIO3 trigger to volume down;
CFG(REG_GPIO_OE, 0x30) → write 0x00 into reg0x1A; configure GPIO 0,1,2 and 3 pins as input pins;
GPIO4, and GPIO5 as I/O output pins ;
CFG(REG_GPIO_DO, 0x30) → write 0x00 into reg0x19; GPIO4 and 5 output high;
CFG(REG_GPIO_PE, 0x3F) → write 0x3F into reg0x1B; enable Pull-Enable for GPIO0-5;
CFG(REG_GPIO_PS, 0x3F) → write 0x3F into reg0x1D; enable Pull-High for GPIO0-5;
CFG(REG_GPIO_AF1, 0x0F) → write 0x0F into reg0x1E;
CFG(REG_GPIO_AF0, 0x00) → write 0x00 into reg0x1F; combined with reg0x1E to confiure GPIO0-3 as
falling edge triggering pins; GPIO4 and 5 as output pins;
CFG(R0, 0x08) → write 0x08 into reg0x20; associate VM0x08 with GPIO0;
CFG(R1, 0x04) → write 0x04 into reg0x22; associate VM0x04 with GPI1;
CFG(R2, 0x05) → write 0x05 into reg0x24; associate VM0x05 with GPIO2;
CFG(R3, 0x08) → write 0x08 into reg0x26; associate VM0x08 with GPIO3;
CFG(R4, 0x08) → write 0x08 into reg0x28; associate VM0x08 with GPIO4;
CFG(R5, 0x08) → write 0x08 into reg0x2A; associate VM0x08 with GPIO5;
PLAY(FastBeep) → Play VP "Fastbeep"
PD → power down device (to save power).

The Wakeup VM 0x02 enables the playback path for all triggers.

The following explains the custom VM 0x04:

PLAY(one) → Play VP "one"
PD → power down device (to save power).
CFG(R1, 0x06) → write 0x06 into reg0x22; associate VM0x06 with GPI1;
LOOP(Sine_0dot5dB_5_Sec_440Hz,0) → Loop play a sound effect infinitely.
Label1: CFG(CH1_CNTR, 0x10)→ write 0x10 into reg0x0E, initialize channel counter 1 as 0x10;
CFG(REG_GPIO_DO, 0x06) →write 0x06 into reg0x19; drive GPIO output pins

```
WAIT_CNT                    → wait for the current channel counting down to 0;
CFG(CH1_CNTR,      0x10)    → write 0x10 into reg0x0E, initialize channel counter 1 as 0x10;
CFG(        REG_GPIO_DO, 0x38) →write 0x38 into reg0x19; drive GPIO output pins
WAIT_CNT                    → wait for the current channel counting down to 0;
CFG(              MASK,     0x40)→ write 0x40 into reg0x08, mask_goto command branches if VP playing;
MASK_GOTO_LABEL_1           → Mask_Branch, VM execution branches if VP playing.
PD                          → power down device (to save power).
```

The purpose of VM 0x04 is to loop play a sound effect and drive an LED on-and-off during play. The on-and-off time is determined by the current channel counter register *Channel 1 Counter Control*. The Mask_Goto command branches according to the setting defined in the *Mask Jump Control* register.

7.4 User Data

The *ISD-VPE2360* allows the user to reserve user data sectors for storage of application data other than audio. When working on a VPE project, an interface is available within the Reserved Memory panel for the user to specify the start address and the number of the sectors to be reserved.

When there are reserved user data sectors, the *ISD-VPE2360* compiles the image according to the following rule: the audio data (VP or VM data) before and after the reserved section must be complete. That is, for every Voice Prompt or Voice Macro script, the data chunk must be continuous and should not be separated by reserved sections. Thus, the ISD2360 device hardware can successfully fetch data and finish the play for audio operation or VM execution.

The user should only read/write application-specific data within the reserved memory; extending data beyond the reserved sectors will damage the audio data integrity.

Note that the reserved user data sector should not start from address 0x0000. Also be aware that the size of the reserved data chunk must be in multiples of 4 Kbytes.

8. Serial Peripheral Interface

8.1 SPI Features

A standard four-wire Serial Peripheral Interface (SPI) is used for communication between the ISD2360 and the host. The interface consists of an active low slave-select (SSB), a serial clock (SCLK), a data input (Master Out Slave In - MOSI), and a data output (Master In Slave Out - MISO). Also, for some transactions requiring data flow control, a RDY/BSYB signal (pin) is available. The ISD2360 supports **SPI Mode 3:** (1) SCLK must be high when SPI bus is inactive, and (2) data is sampled at the rising edge of SCLK. An SPI transaction begins on the falling edge of SSB and its waveform is illustrated in

Figure 8-1.

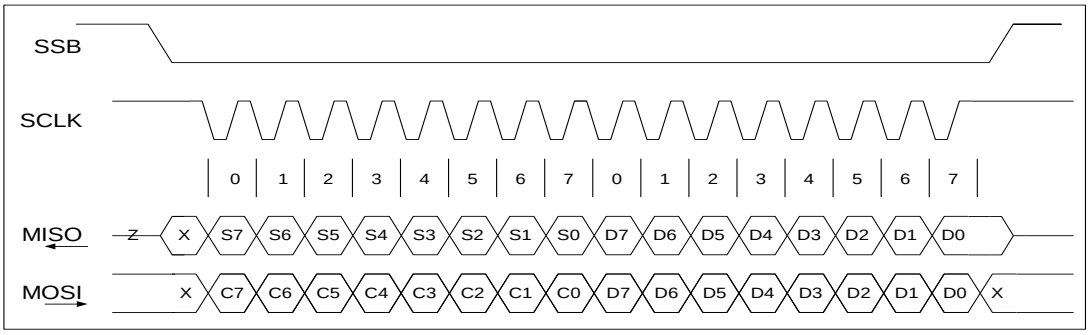


Figure 8-1 SPI Data Transaction Waveform

A transaction begins with a command byte (C7-C0) with the most significant bit (MSB – C7) first. During the byte transmission, the status (S7-S0) of the device is sent out via the MISO pin. After the byte transmission, depending upon the command sent, one or more bytes of data will be sent via the MISO pin.

RDY/BSYB pin is used to handshake data into or out of the device. Upon completion of a byte transmission, RDY/BSYB pin could change its state after the rising edge of the SCLK if the built-in 4-byte data buffer is either full or empty. At this point, SCLK must remain high until the RDY/BSYB pin returns to high, indicating that the ISD2360 is ready for the next data transmission. See

Figure 8-2 for the timing diagram.

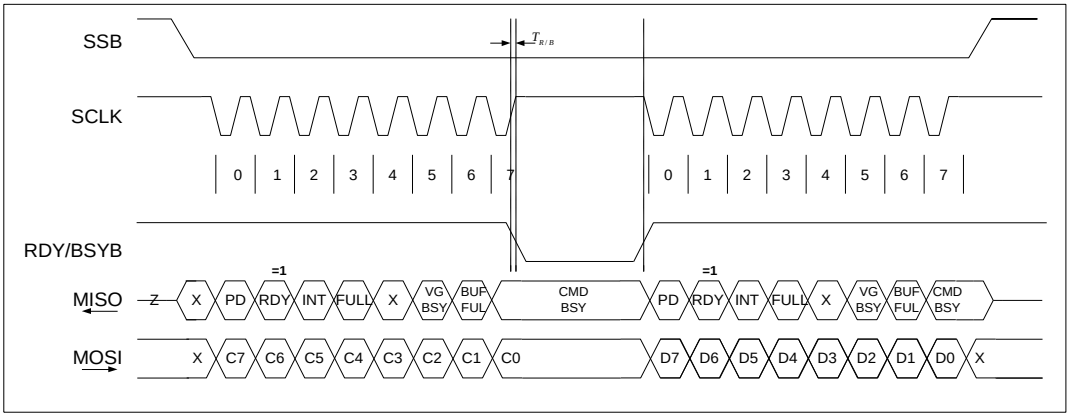


Figure 8-2 RDY/BSYB Timing for SPI Write Transactions

If the SCLK does not remain high, the RDY bit of the status register will be set to zero and will be reported via the MISO pin so the host can take the necessary actions (i.e., terminate SPI transmission and re-transmit the data when the RDY/BSYB pin returns to high).

For commands that read data from the ISD2360 device, (i.e., DIG_READ, SPI_PCM_READ), MISO is used to read the data. Therefore, the host must monitor the status via the RDY/BSYB pin and take the necessary actions. The INT pin will go low to indicate (1) data overrun/overflow when sending data to the ISD2360; or (2) invalid data from ISD2360. Refer to **Figure** 8-3 for the timing diagram in which RDY/BSYB is ignored.

The following conditions must be met to avoid RDY/BSYB polling for digital operations:

- Ensure the device is idle (CMD_BSY=0 in status) before operation.
- Digital Write: Send 32 bytes of data or less in a digital write transaction **or** ensure that there is a 24 µs period between each byte sent where SCLK is held high.
- Digital Read: Ensure a 2 µs period between the last address byte of a digital read command and first data byte where SCLK is held high.

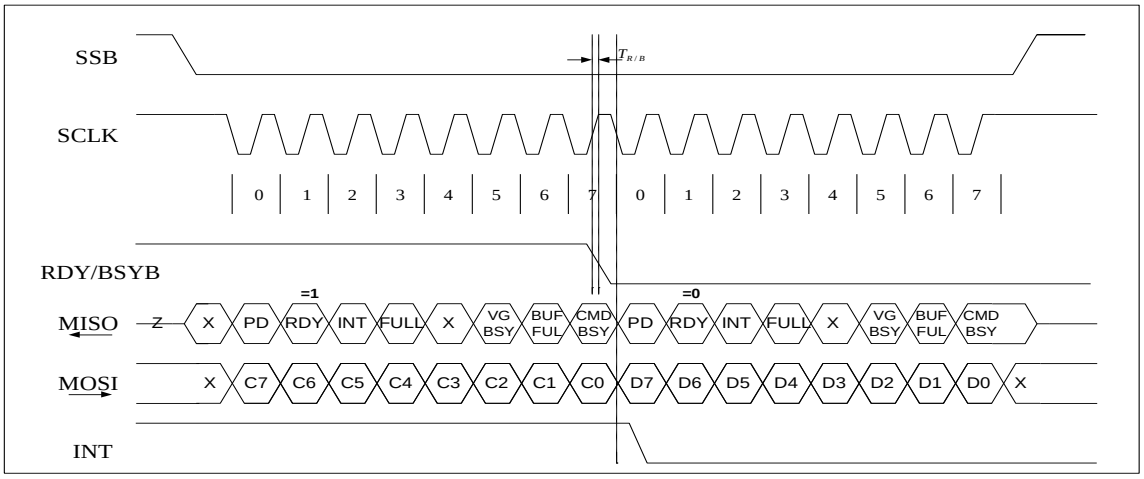


Figure 8-3 RDY/BSYB Ignored for SPI Transactions

8.2 SPI Commands

The ISD2360 provides SPI commands to play audio, query device status, perform digital memory operations and configure the device. Table 8-1 provides a list of all SPI commands and their function descriptions.

Table 8-1 SPI Commands

Instructions	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 … Byte n	Description
PLAY_VP	0xA6	INX[15:8]	INX[7:0]			Play Voice Prompt Index INX
PLAY_VP@Rn	0xAE	*n* = 0 …7				Play Voice Prompt; Index is value in register R*n*.
PLAY_VP_LP	0xA4	INX[15:8]	INX[7:0]	CNT[15:8]	CNT[7:0]	Loop Play Voice Prompt Index INX, CNT times
PLAY_VP@Rn_LP	0xB2	*n* = 0 …7	CNT[15:8]	CNT[7:0]		Loop Play Voice Prompt; Index in register R*n,* CNT times
EXE_VM	0xB0	Index[15:8]	Index[7:0]			Execute Voice Macro Index
PLAY_SIL	0xBC	*n* = 0 …7				Execute Voice Macro; Index contained in register R*n*
PLAY_SIL	0xA8	LEN[7:0]				Play silence for LEN*32ms
STOP	0x2A					STOP current playback operation
STOP_LP	0x2E					Stop Loop Play Voice Prompt
SPI_PCM_READ	0xAC	D0[7:0]	D0[15:8]	D1[7:0]	D1[15:8] …Dn[7:0] Dn[15:8]	Receive 16 bit PCM audio data [low-byte, high-byte] from ISD2360 via SPI interface.
SPI_SND_DEC	0xC0	D0[7:0]	D1[7:0]	D2[7:0]	D3[7:0] …Dn[7:0]	Send compressed audio data to ISD2360 via SPI interface for decoding.
READ_STATUS	0x40	XX	XX	XX	…	Query status of ISD2360.
READ_INT	0x46	XX	XX	XX	…	Query status and clear interrupt flags of ISD2360.
READ_ID	0x48	XX	XX	XX	XX	Read device ID of ISD2360.
DIG_READ	0xA2	A[23:16]	A[15:8]	A[7:0]	XX, … XX	Read digital data from address A.
DIG_WRITE	0xA0	A[23:16]	A[15:8]	A[7:0],	D0[7:0], … Dn[7:0]	Write digital data from address A.
ERASE_MEM	0x24	SA[23:16]	SA[15:8]	SA[7:0]		Erase 1 kByte sector of memory containing start address SA.
CHIP_ERASE	0x26	0x01				Initiate a mass erase of memory.
CHECKSUM	0xF2	EA[23:16]	EA[15:8]	EA[7:0]		Calculate checksum from 0x0000 to the specified end address EA.
PWR_UP	0x10					Power up ISD2360
PWR_DN	0x12					Power down ISD2360
SET_CLK_CFG	0X34	CFG_CLK[7:0]				Sets clock configuration register; default setting must be retained.

Instructions	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 … Byte n	Description
RD_CLK_CFG	0xB6	XX				Read clock configuration register.
WR_CFG_REG	0xB8	REG[7:0]	D0[7:0], …Dn[7:0]			Write data D0…Dn to configuration register(s) starting at configuration register REG.
RD_CFG_REG	0xBA	REG[7:0]	XX, …XX			Read configuration register(s) starting at configuration register REG.
RESET	0x14					Reset all the registers and initiate POI procedure.

8.3 SPI Command vs. Status

Each SPI command will be accepted only if certain conditions are met, as described in **Table 8-2**, or a CMD_ERR interrupt will be generated and the command will be ignored. It is a good practice to check the device status prior to sending commands to ensure the success of the SPI command.

Table 8-2 SPI Commands vs. Status

Instructions	Op Code	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		PD	DBUF_RDY	INT	-	CH2_BSY	CH1_BSY	CH0_BSY	DIG_BSY
PLAY_VP	0xA6	0	1	x	-	√	√	√	x
PLAY_VP@Rn	0xAE	0	1	x	-	√	√	√	x
PLAY_VP_LP	0xA4	0	1	x	-	√	√	√	x
PLAY_VP@Rn_LP	0xB2	0	1	x	-	√	√	√	x
STOP	0x2E	0	1	x	-	x	x	x	x
EXE_VM	0xB0	0	1	x	-	√	√	√	0
PLAY_SIL	0xBC	0	1	x	-	√	√	√	0
PLAY_SIL	0xA8	0	1	x	-	√	√	√	x
STOP	0x2A	0	1	x	-	x	x	x	x
STOP_LP	0x2E	0	1	x	-	x	x	x	x
SPI_PCM_READ	0xAC	0	1	x	-	x	x	x	x
SPI_SND_DEC	0xC0	0	1	x	-	0	0	0	0
READ_STATUS	0x40	x	x	x	-	x	x	x	x
READ_INT	0x46	x	x	x	-	x	x	x	x
READ_ID	0x48	0	1	x	-	x	x	x	x
DIG_READ	0xA2	0	1	x	-	0	0	0	0
DIG_WRITE	0xA0	0	1	x	-	0	0	0	0

Instructions	Op Code	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		PD	DBUF_RDY	INT	-	CH2_BSY	CH1_BSY	CH0_BSY	DIG_BSY
ERASE_MEM	0x24	0	1	x	-	0	0	0	0
CHIP_ERASE	0x26	0	1	x	-	0	0	0	0
CHECKSUM	0xF2	0	1	x	-	0	0	0	0
PWR_UP	0x10	1	0	x	-	x	x	x	x
PWR_DN	0x12	0	1	x	-	x	x	x	x
SET_CLK_CFG	0X34	0	1	x	-	0	0	0	0
RD_CLK_CFG	0xB6	0	1	x	-	x	x	x	x
WR_CFG_REG	0xB8	0	1	x	-	x	x	x	x
RD_CFG_REG	0xBA	0	1	x	-	x	x	x	x
RESET	0x14	x	x	x	-	x	x	x	x

Note: **"√"** indicates that the current channel in which the SPI operation runs must be idle for the SPI command to succeed. Concurrently, the other two channel CHx_BSY bits are don't care bits.

8.4 SPI Command Descriptions

8.4.1 Audio Play Commands

8.4.1.1 *PLAY_VP – Play Voice Prompt*

PLAY_VP				
Byte Sequence:	Host Controller	0xA6	Index[15:8]	Index[7:0]
	ISD2360	Status Byte	Status Byte	Status Byte
Description:	Play Voice Prompt Index			
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback is finished.			

This command initiates the play of a pre-recorded voice-prompt. Before execution of the command, a valid signal path must be set up and the device must have space in the audio command buffer. After completion of playback, the device will generate an interrupt. The command will be accepted if status bits PD=0, DBUF_RDY=1, CHx_BSY=0 and DIG_BSY=0. If any of these conditions are not met, a CMD_ERR interrupt will be generated and the command will be ignored. If the command is terminated after the command byte is sent, no interrupt will be generated. Once playback is finished, a CMD_FIN interrupt will be generated. This command will be ignored when SPI_CMD_CH=3.

8.4.1.2 *PLAY_VP@Rn – Play Voice Prompt @ Rn*

PLAY_VP@Rn				
Byte Sequence:	Host Controller	0xAE	Rn[7:0]	
	ISD2360	Status Byte	Status Byte	
Description:	Play Voice Prompt with Index stored in Rn register			
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback is finished.			

This command initiates the play of a pre-recorded voice-prompt. Before execution of the command, a valid signal path must be set up and the device must have space in the audio command buffer. After completion of playback, the device will generate an interrupt. The command will be accepted if status bits PD=0, DBUF_RDY=1, CHx_BSY=0 and DIG_BSY=0. If any of these conditions are not met, a CMD_ERR interrupt will be generated and the command will be ignored. If the command is terminated after the command byte is sent, no interrupt will be generated. Once playback is finished, a CMD_FIN interrupt will be generated. This command will be ignored when SPI_CMD_CH=3.

8.4.1.3 *PLAY_VP_LP – Play Voice Prompt Loop*

PLAY_VP_LP						
Byte Sequence	Host Controller	0xA4	Index[15:8]	Index[7:0]	Loop_Cnt[15:8]	Loop_Cnt[7:0]
	ISD2360	Status Byte	Status Byte	Status Byte	Status Byte	Status Byte
Description:	Play Voice Prompt Index with looping					
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback is finished.					

This command initiates the play of a pre-recorded voice-prompt. Before execution of the command, a valid signal path must be set up and the device must have space in the audio command buffer. After completion of playback, the device will generate an interrupt. The command will be accepted if status bits PD=0, DBUF_RDY=1, CHx_BSY=0 and DIG_BSY=0. If any of these conditions are not met, a CMD_ERR interrupt will be generated and the command will be ignored. If the command is terminated after the command byte is sent, no interrupt will be generated. Once playback is finished, a CMD_FIN interrupt will be generated. This command will be ignored when SPI_CMD_CH=3.

8.4.1.4 *PLAY_VP@Rn_LP – Loop Play Voice Prompt Referenced by RnCnt Times*

PLAY_VP@Rn_LP					
Byte Sequence:	Host Controller	0xB2	Rn[7:0]	Loop_Cnt[15:8]	Loop_Cnt[7:0]
	ISD2360	Status Byte	Status Byte	Status Byte	Status Byte
Description:	Play Voice Prompt with Index stored in Rn register with looping				
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback is finished.				

This command initiates the play of a pre-recorded voice-prompt. Before execution of the command, a valid signal path must be set up and the device must have space in the audio command buffer. After completion of playback, the device will generate an interrupt. The command will be accepted if status bits PD=0, DBUF_RDY=1, CHx_BSY=0 and DIG_BSY=0. If any of these conditions are not met, a CMD_ERR interrupt will be generated and the

command will be ignored. If the command is terminated after the command byte is sent, no interrupt will be generated. Once playback is finished a CMD_FIN interrupt will be generated. This command will be ignored when SPI_CMD_CH=3.

8.4.1.5 *EXE_VM –Execute Voice Macro*

EXE_VM				
Byte Sequence:	Host Controller	0xB0	Index[15:8]	Index[7:0]
	ISD2360	Status Byte	Status Byte	Status Byte
Description:	Play Voice Macro Index			
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback is finished.			

This command initiates the execution of a pre-recorded voice group. After completion of the Voice Macro, the device will generate a CMD_FIN interrupt. The command will be accepted if status bits PD=0, DBUF_RDY=1, CHx_BSY=0 and DIG_BSY=0. If any of these conditions are not met, a CMD_ERR interrupt will be generated and the command will be ignored. If the command is terminated after the command byte is sent, no interrupt will be generated. Once the Voice Macro execution is finished, a CMD_FIN interrupt will be generated. This command will be ignored when SPI_CMD_CH=3.

8.4.1.6 *EXE_VM@Rn – Execute Voice Macro Referenced by Rn*

EXE_VM@Rn				
Byte Sequence:	Host controller	0xBC	Rn[7:0]	
	ISD2360	Status Byte	Status Byte	
Description:	Play Voice Macro with index stored in Rn			
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when playback is finished.			

This command initiates the execution of a pre-recorded voice group. After completion of the Voice Macro, the device will generate a CMD_FIN interrupt. The command will be accepted if status bits PD=0, DBUF_RDY=1, CHx_BSY=0 and DIG_BSY=0. If any of these conditions are not met, a CMD_ERR interrupt will be generated and the command will be ignored. If the command is terminated after the command byte is sent, no interrupt will be generated. Once the Voice Macro execution is finished, a CMD_FIN interrupt will be generated. This command will be ignored when SPI_CMD_CH=3.

8.4.1.7 *PLAY_SIL – Play Silence*

PLAY_SIL				
Byte Sequence:	Host Controller	0xA8	LEN[7:0]	
	ISD2360	Status Byte 0	Status Byte 0	
Description:	Play silence for LEN*32 ms			
Interrupt Generation:	CMD_ERR if not accepted. CMD_FIN when silence playback complete.			

This command plays a period of silence to the signal path. Before execution of the command, a valid signal path must be set up and the device must have space in the audio command buffer. After completion, the device will generate an interrupt. The duration that silence played is determined by the data byte, LEN, sent. Silence is played in 32 ms increments (at the signal path sampling frequency of 32 kHz); the total silence played is LEN*32 ms.

The command will be accepted if status bits PD=0, DBUF_RDY=1, CHx_BSY=0 and DIG_BSY=0. If any of these conditions are not met, a CMD_ERR interrupt will be generated and the command ignored. If the command is terminated after the command byte is sent, no interrupt will be generated. Once silence play is finished, a CMD_FIN interrupt will be generated. This command will be ignored when SPI_CMD_CH=3.

8.4.1.8 *STOP – Stop the Play Operations*

STOP				
Byte Sequence:	Host Controller	0x2A		
	ISD2360	Status Byte		
Description:	Stop current audio command and flush command buffer.			
Interrupt Generation:	The command does not generate an interrupt; the command being stopped generates the interrupt.			

This command stops the audio command active on a given channel in the ISD2360. If a PLAY_MSG@, PLAY_VP, EXE_VM or PLAY_SIL command is active, playback is stopped immediately. The STOP command flushes the audio command buffer, so that any command queued in the buffer when a STOP is issued will not be executed. When the device has finished the active command, a CMD_FIN interrupt will be generated. STOP will not stop an ERASE_MEM operation. If there is no active command, STOP will have no effect. The STOP command applies to the channel set in SPI_CMD_CH. When SPI_CMD_CH=3 and a STOP command is issued, the STOP command applies to all three channels.

8.4.1.9 *STOP_LP – Stop Loop Play Operations*

STOP_LP				
Byte Sequence:	Host Controller	0x2E		
	ISD2360	Status Byte		
Description:	Stop current PLAY_VP_LP or PLAY_VP@Rn_LP			
Interrupt Generation:	The command does not generate an interrupt; the command being stopped generates the interrupt.			

This command only stops a currently active LP or PLAY_VP@Rn_LP. The STOP_LP command applies to the channel set in SPI_CMD_CH. When SPI_CMD_CH=3 and a STOP_LP command is issued, the STOP_LP command applies to loop commands on any of the three channels.

8.4.1.10 *SPI_PCM_READ – SPI Read De-Compressed PCM Data from Memory*

SPI_PCM_READ							
Byte Sequence:	Host controller	0xAC					
	ISD2360	Status Byte	D0[7:0]	D0[15:8]	….	Dn[7:0]	Dn[15:8]
Description:	Read audio data via the SPI interface.						
Interrupt Generation:	OVF_ERR if RDY/BSY violated.						

This command allows the user to receive audio data, in 16-bit PCM format, from the SPI interface. Before execution of the command, a valid signal path must be set up.

When receiving audio data from memory (SPI playback): (1) a signal path must be set up for SPI output from the de-compressor. (2) A valid play command is then sent; valid play

commands include PLAY_VP, PLAY_VP@Rn, PLAY_VP_LP, PLAY_VP_LP@Rn, EXE_VM and EXE_VM@Rn. (3) The SPI_PCM_READ command follows. Multiple SPI_PCM_READ commands can be sent. (4) To finish receiving data, a STOP command is sent and the device will generate a CMD_FIN interrupt.

When the end of the message is reached, a CMD_FIN interrupt will be generated and zero will be sent as data. If the valid play command in step (2) above is EXE_VM, a CMD_FIN interrupt will be generated at the end of the Voice Macro. See **Figure 8-4**.

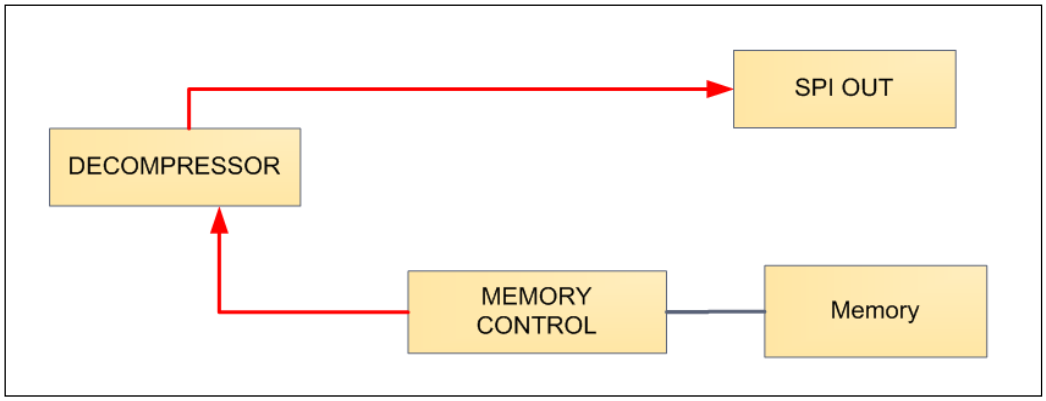


Figure 8-4 SPI Read De-Compressed Data (Playback)

The RDY/BSYB pin will go low whenever the internal FIFO is empty. If no path or playback operation is set up, the RDY/BSYB pin will remain low until the command is terminated. If RDY/BSYB is ignored, then an OVF_ERR interrupt will be generated.

8.4.1.11 *SPI_SND_DEC – SPI Send Compressed Data for Decoding*

SPI_SND_DEC						
Byte Sequence:	Host Controller	0xC0	D0[7:0]	D1[7:0]	….	Dn[7:0]
	ISD2360	Status Byte				
Description:	Write compressed audio data via SPI interface.					
Interrupt Generation:	OVF_ERR if RDY/BSYB violated.					

This command allows the user to send compressed audio data, in a byte-formatted bit stream, down the SPI interface to the de-compressor and signal path.

Before execution of the command, a valid signal path must be set up; valid paths are similar to a standard playback. Multiple SPI_SND_DEC commands can be issued to send data to the ISD 2100.

To finish decoding, a STOP command is sent and the device will respond with a CMD_FIN interrupt. The RDY/BSYB pin will handshake dataflow if the device cannot accept any further data for decompression. See **Figure 8-5**.

Note: If the host cannot keep up with data rate, the audio output will be corrupted.

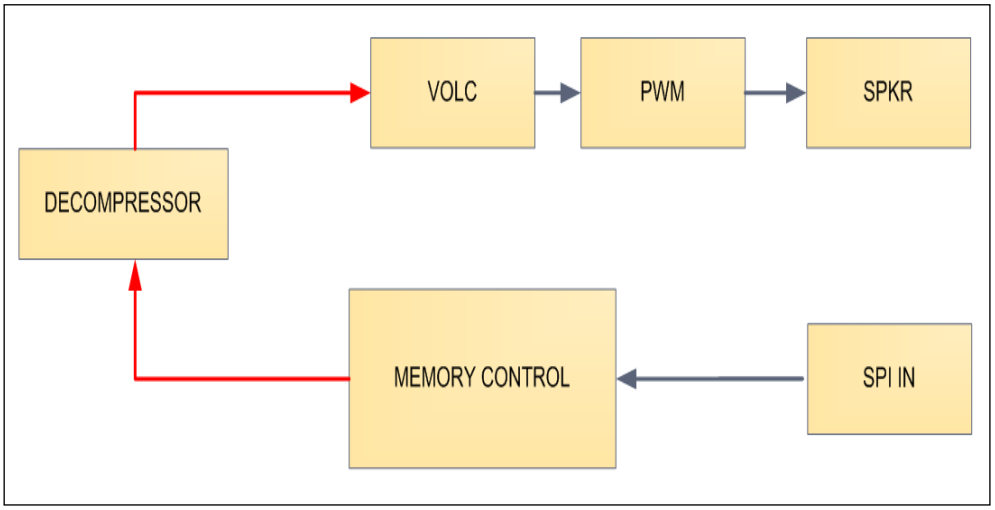


Figure 8-5 SPI Send Compressed Data to Decode

The RDY/BSYB pin will go low whenever the internal FIFO is full. If no path is set up to accept audio data, the RDY/BSYB pin will not return high until the command is terminated. If RDY/BSYB is ignored, an OVF_ERR interrupt will be generated. The SPI_SND_DEC command is accepted if no current play operation is active. If the command is not accepted, a CMD_ERR interrupt will be generated.

Note: It is possible to perform digital memory operations between SPI_SND_DEC operations; but, care must be taken to maintain the required data rate to avoid audio corruption

8.4.2 Device Status Commands

8.4.2.1 *READ_STATUS – Read Status*

READ_STATUS			
Byte Sequence:	Host Controller	0x40	0xXX
	ISD2360	Status Byte	Interrupt Status Byte
Description:	Query device status.		

This command queries the ISD2360 device status. For details, see **Section** 4 **Device Status**. If the device is powered up, the two status bytes will be repeated for each two dummy bytes sent to the SPI interface. If the device is powered down, only one status byte, 8oh, goes to the SPI interface at the same time the command is sent. The command is always accepted.

8.4.2.2 *READ_INT – Read Interrupt*

READ_INT				
Byte Sequence:	Host Controller	0x46	0xXX	
	ISD2360	Status Byte	Interrupt Status Byte	
Description:	Query device status and clear interrupt flags.			

This command queries the ISD2360 device status and clears any pending interrupts. After this command, the hardware interrupt line will return inactive. The INT bit of the status register, and any status error bits, will return inactive. This command is accepted whenever the device is powered up.

8.4.2.3 *READ_ID - Read Device ID*

READ_ID						
Byte Sequence:	Host Controller	0x48	0xXX	0xXX	0xXX	0xXX
	ISD2360	Status Byte	PART_ID	MAN_ID	MEM_TYPE	DEV_ID
Description:	Return ID of ISD2360					

This command queries the ISD2360 and returns four bytes to identify the ISD2360 part, the manufacturer, and the size and type of internal memory of the device. The bytes returned are:

- One byte ISD2360 Family ID, which is 0x05.
- Three bytes Flash JEDEC ID.
- MAN_ID=0xEF
- MEM_TYPE=0x20
- SIZE_ID = 0x60 (see **Device ID**.)

8.4.3 Digital Commands

This section describes the digital data commands that can be sent to the ISD2360. Digital commands are those that read, write or erase data in the Flash memory. Digital and Audio commands cannot occur concurrently. To prevent the possibility of data corruption, the user should disable any Voice Macro triggers and check the device status to ensure it is idle (CMD_BSY=0) before sending any digital memory commands.

8.4.3.1 *DIG_READ – Digital Read*

DIG_READ								
Byte Sequence:	Host Controller	0xA2	A[23:16]	A[15:8]	A[7:0]	0xXX	…	0xXX
	ISD2360	Status	Status	Status	Status	D0	…	Dn
Description:	Initiates a digital read of memory from address A [23:0].							
Interrupt Generation:	ADDR_ERR if memory is protected or if RDY/BSYB is violated. OVF_ERR if the read is past the end of the array.							

This command initiates a read of Flash memory from address A[23:0]. Following the three address bytes, data can be read out of memory in a sequential manner. The RDY/BSYB pin is used to control the flow of data. If the RDY/BSYB pin goes low, the transfer must be paused until the RDY/BSYB pin returns high. The user should check the RDY/BSYB pin before every byte is sent/read, including the command and address bytes. As many bytes of data as required can be read. The command is terminated by raising SSB high, finishing the SPI transaction. If an attempt is made to read past the end of memory, the status byte will be read back.

The command will always be accepted, and the RDY/BSYB pin will go low until any active digital memory command is complete. If a digital read is attempted in read-protected memory, the status byte will be read back and an ADDR_ERR interrupt will be generated. If a read past the end of memory is attempted, an OVF_ERR interrupt will be generated. If RDY/BSYB is violated, zero data will be read back and an OVF_ERR interrupt will be generated.

Digital and Audio commands cannot occur concurrently. To prevent the possibility of data corruption, the user should disable any Voice Macro triggers and check the status to ensure that the device is idle (CMD_BSY=0) before sending a digital read command.

8.4.3.2 *DIG_WRITE – Digital Write*

DIG_WRITE								
Byte Sequence:	Host Controller	0xA0	A[23:16]	A[15:8]	A[7:0]	D0	…	Dn
	ISD2360	Status	Status	Status	Status	Status	…	Status
Description:	Initiates a digital write to memory from address A[23:0].							
Interrupt Generation:	ADDR_ERR if memory is protected or if RDY/BSYB is violated. OVF_ERR if the write is past the end of the array.							

This command initiates a write to Flash memory from address A [23:0]. Following the three address bytes, data can be written to memory in a sequential manner. The RDY/BSYB pin is used to control the flow of data. If the RDY/BSYB pin goes low, the transfer must be paused until the RDY/BSYB pin returns high. The user should check the RDY/BSYB pin before every byte is sent, including the command and address bytes. As many bytes of data as required can be written. The command is terminated by raising SSB high, finishing the SPI transaction.

The command will always be accepted, and the RDY/BSYB pin will go low until any active digital memory command is complete. If a digital write is attempted in write-protected memory, the data will be ignored and an ADDR_ERR interrupt will be generated. If a write is attempted past the end of memory, an OVF_ERR interrupt will be generated. If RDY/BSYB is violated, the data will ignored and an OVF_ERR interrupt will be generated. Once the SPI transaction has ended, the ISD2360 will finish the Flash write operation. When this operation is complete, the ISD2360 will generate a WR_FIN interrupt. While the device is actively writing to Flash memory, the CMD_BSY bit will be active.

Digital and Audio commands cannot occur concurrently. To prevent the possibility of data corruption, the user should disable any Voice Macro triggers and check the status to ensure that the device is idle (CMD_BSY=0) before sending a digital write command.

8.4.3.3 *ERASE_MEM – Sector Erase Memory*

ERASE_MEM					
Byte Sequence:	Host Controller	0x24	SA[23:16]	SA[15:8]	SA[7:0]
	ISD2360	Status	Status	Status	Status
Description:	Erases one sector of memory starting from SA				
Interrupt Generation:	ADDR_ERR if memory is protected. CMD_ERR if the device is busy. CMD_FIN when the erase operation completes.				

This command erases memory from the sector containing start address SA. The minimum erase block is a 1-KByte sector of internal memory.

The command will be accepted if status bits PD=0, DBUF_RDY=1,CHx_BSY=0, DIG_BSY=0 and CMD_BSY=0. If any of these conditions are not met, a CMD_ERR interrupt will be generated and the command will be ignored. If the memory is write-protected, an ADDR_ERR interrupt will be generated. Upon completion of the erase, a CMD_FIN interrupt will be generated.

No other commands will execute while the device is erasing. If a PLAY command is sent, it is queued in the command buffer and will not execute until the erase is finished. If a DIG_RD or DIG_WR command is sent to the device, the RDY/BSYB pin will hold off any data transfer until the ERASE_MEM has completed.

When ERASE_MEM is in progress, the Status bit 0 CMD_BSY goes high. The user could poll the status to determine if the erasing is complete.

Digital and Audio commands cannot occur concurrently. To prevent the possibility of data corruption, the user should disable any Voice Macro triggers and check the status to ensure the device is idle (CMD_BSY=0) before sending a digital erase command.

8.4.3.4 *CHIP_ERASE – Erase Entire Memory*

CHIP_ERASE			
Byte Sequence:	Host Controller	0x26	0x01
	ISD2360	Status Byte	Status Byte
Description:	Initiate a mass erase of memory.		
Interrupt Generation:	CMD_ERR if the device is busy and cannot accept a command. CMD_FIN when the erase operation completes.		

This command erases the entire contents of the Flash memory.

The command will be accepted if the status bits PD=0, DBUF_RDY=1, CHx_BSY=0, DIG_BSY=0 and CMD_BSY=0. If any of these conditions are not met, a CMD_ERR interrupt will be generated and the command will be ignored. If memory is mass-erase-protected, an ADDR_ERR interrupt is generated. Upon completion of erase, a CMD_FIN interrupt will be generated.

While the device is erasing, no other commands will execute. If a PLAY command is sent, it is queued in the command buffer and will not execute until the erase is finished. If a DIG_RD or DIG_WR command is sent to the device, the RDY/BSYB pin will hold off any data transfer until the CHIP_ERASE has completed.

When CHIP_ERASE is in progress, the Status bit 0 CMD_BSY goes high. The user may poll the status to determine if the erasing is complete.

8.4.3.5 *CHECKSUM – Calculate Hardware Checksum*

CHECKSUM					
Byte Sequence:	Host Controller	0xF2	EA[23:16]	EA[15:8]	EA[7:0]
	ISD2360	Status Byte	Status	Status	Status
Description:	Initiate a checksum of memory.				
Interrupt Generation:	CMD_ERR if the device is busy and cannot accept command. CMD_FIN when erase operation is complete.				

This initiates a 4-byte checksum calculation from the very beginning to the specified end address. The calculated checksum is stored in configuration registers 0x10 – 0x013. To re-calculate the checksum with a different end address, the user must write register 1 followed by 0 to CFG_REG4 bit-4 to clear the registers 0x10 – 0x13.

The command is accepted if status bits PD=0, DBUF_RDY=1, CHx_BSY=0, DIG_BSY=0 and CMD_BSY=0. If any of these conditions are not met, a CMD_ERR interrupt will be generated and the command will be ignored. If the memory is mass-erase-protected, an ADDR_ERR interrupt will be generated. Upon completion, a CMD_FIN interrupt will be generated.

When CHECKSUM is in progress, the Status bit 0 CMD_BSY goes high. The user may poll CMD_BSY to determine if the checksum calculation is complete.

8.4.4 Device Configuration Commands

Six commands are used to configure the ISD2360. These commands are used to set up the clocking regime of the device (including the clock source and setting the master sample rate) and to configure the audio signal path, compression and sample rate. The signal path, compression and sample rate configuration are controlled by 48 bytes of the configuration register. These 48 bytes can be written individually or in a continuous sequential manner.

8.4.4.1 *PWR_UP – Power Up Device*

PWR_UP				
Byte Sequence:	Host Controller	0x10		
	ISD2360	Status		…
Description:	Powers up the device and initiates the power up sequence.			

This command powers up the ISD2360. If the device is already powered up, this command has no effect. If it is powered down, the internal power up sequence is initiated. If the Power Up (PU) Voice Macro is present, it is executed. Otherwise, the device defaults to power up the internal oscillator. When power up is complete, the PD bit of the status register will go low and the RDY bit will go high. Until this occurs, no other commands will be accepted.

The formal power-up procedure is as follows:

- Send PWR_UP command,
- Poll Status until bit-6 DBUF_RDY goes high, which means ready,
- If there is Power-Up Voice Macro implemented, poll the device status until CHx_BSY goes low, which means the Power-Up Voice Macro is finished.

8.4.4.2 *PWR_DN – Power Down Device*

PWR_DN				
Byte Sequence:	Host Controller	0x12		
	ISD2360	Status		…
Description:	Powers down the device after any active commands finish			

This command powers down the device. If the device is currently executing a command, the device will power down when the command finishes. If playing or executing a Voice Macro, the device will power down after playback is finished. The PWR_DN command will not generate an interrupt. PWR_DN has executed when the PD bit of status goes high.

8.4.4.3 *SET_CLK_CFG – Set Clock Configuration Register*

SET_CLK_CFG				
Byte Sequence:	Host Controller	0X34	CFG_CLK[7:0]	
	ISD2360	Status Byte	Status Byte	
Description:	Loads clock configuration register. Default setting must be retained.			

This sets the Clock Configuration Register. The part reconfigures the clock and PLL configuration and waits for stable clock conditions before accepting new commands. When the configuration is changed, CMD_BSY will go high until clock configuration is complete. No new commands should be sent when the device status shows that the device is busy.

Note: This command does not generate an interrupt.

8.4.4.4 *RD_CLK_CFG – Read Clock Configuration Register*

RD_CLK_CFG				
Byte Sequence: T	Host Controller	0xB6	0xXX	
	ISD2360	Status Byte	CFG_CLK[7:0]	
Description:	Reads the clock configuration register.			

This command reads the Clock Configuration Register.

8.4.4.5 *WR_CFG_REG – Write Configuration Register*

WR_CFG_REG						
Byte Sequence:	Host Controller	0xB8	REG[7:0]	D0	…	Dn
	ISD2360	STATUS0			…	
Description:	Loads configuration register CFG[REG] with D0. Data bytes 1..n can be sent to load CFG[REG+1] with D1 to CFG[REG+n] with Dn.					

This command loads configuration registers starting at the address specified. If multiple data bytes are sent, additional configuration registers are loaded.

8.4.4.6 *RD_CFG_REG – Read Configuration Register*

RD_CFG_REG						
Byte Sequence:	Host Controller	0xBA	REG[7:0]	X	…	X
	ISD2360	STATUS0		D0	…	Dn
Description:	Reads configuration register CFG[REG] and outputs to SPI as D0. Data bytes 1..n can be read sequentially from CFG[REG+1] to CFG[REG+n].					

This command reads the configuration register starting at the address specified. If multiple data bytes are sent, additional configuration registers are read.

8.4.4.7 *RESET – Reset Device*

RESET						
Byte Sequence:	Host Controller	0x14	X	…	X	
	ISD2360	Status		…		
Description:	Reset command resets all the registers and initiates the POI procedure.					

The Reset command is a 1-byte SPI command that resets all the registers and initiates a POI execution, if there is POI VM0 available. If the POI Voice Macro is empty, the device will go to power down.

9. Register Operations

Table 9-1 Register Operations

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
-	-	Clock Register										Reserved
			CLK_INP_SEL									Select system clock source. Use SET_CLK_REG and RD_CLK_REG to access this register. 00 = Internal Oscillator with internal Resistor 01 = Reserved 10 = Reserved 11 = Reserved
			Default	0	0	1	1	0	1	0	0	0x00 reset value. Read/Write.
-	-	Device Status Register	PD									Device Powered Up/Down Indicator. 1 = Device is powered down 0 = Device is powered up.
			DBUF_RDY									When the device is powered up, DBUF_RDY bit reflects the state of the RDY/BSYB pin.
			INT									This bit is set by hardware each time a playback operation completes. Must be cleared by software by a SPI READ_INT operation. If GPIO3/INTB pin is configured as INTB function pin, when INT bit is set, an active low interrupt is generated on INTB pin; and a SPI RED_INT operation can set INTB pin high.
			-									Reserved
			CH2_BSY									Set by hardware when Channel 2 is playing. Cleared by hardware when Channel 2 is idle.
			CH1_BSY									Set by hardware when Channel 1 is playing. Cleared by hardware when Channel 1 is idle.
			CH0_BSY									Set by hardware when Channel 0 is playing. Cleared by hardware when Channel 0 is idle.
			DIG_BSY									Set by hardware when the memory controller is busy processing memory access. Cleared by hardware when the memory controller is idle.
			Default	0	1	0	0	0	0	0	0	0x40 reset value. Read only. Note: POI VM may change the device Status register value after reset.
-	-	Interrupt Status Register	-									Reserved
			MPT_ERR									Set when digital access violates the memory protection scheme. Must be cleared by software READ_INT operation.
			WR_FIN									Set when digital write operation successfully completes. Must be cleared by software READ_INT operation.
			CMD_ERR									Set when the device receives an invalid command. Must be cleared by software READ_INT operation.
			OVF_ERR									Set when there is an invalid digital read/write operation when RDY/BSYB pin is low. Must be cleared by software READ_INT operation.
			CH2_CFIN									Set when a playback completes in Channel 2. Must be cleared by software READ_INT operation.
			CH1_CFIN									Set when a playback completes in Channel 1. Must be cleared by software READ_INT operation.
			CH0_CFIN									Set when a playback completes in Channel 0. Must be cleared by software READ_INT operation.
			Default	0	0	0	0	0	0	0	0	0x00 reset value. Read only. Note: POI VM may change the device Interrupt Status register value after reset.

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
g0	00	**Sample Rate Overwrite Register**	SR									Sample Rate: 000 = 4 KHz 001 = 5.33 KHz 010 = 6.4 KHz 011 = 8 KHz 100 = 12.8 KHz 101 = 16 KHz 110 = 32 KHz 111 = 10.67 KHz
			-									Reserved
			Default >>	0	1	1	0	0	1	0	0	0x64 reset value. Read/Write.
1	01	**De-compression Control Register**	FIFO_FULL									FIFO over-run indicator. Set by hardware when FIFO is full while more data keeps coming from ADC path, which causes loss of data. 1 = FIFO is full 0 = FIFO is not full
			FIFO_EMPTY									FIFO under-run indicator. Set by hardware when FIFO is empty while DAC path requests data. 0 = FIFO is not empty 1 = FIFO is empty
			NLRAMP									Auto ramp control during the loop play. 0 = default value. Device will auto ramp during the loop play. 1 = forbid auto ramp during the loop play
			CFG0_READ									Register 0x00 read out option control 0 = read register 0x00 returns its register value 1 = read register 0x00 returns the current effective sample rate
			Reserved									Reserved
			NRMP									Overturn the auto ramp control at the end of a playback. 0 = default value. Device will ramp down at the end of playback. 1 = device does not ramp down at the end of a playback.
			NSRSIL									Overturn the automatic silence insertion between two VPs with different sample rates. 0 = default value. Device will play silence between consecutive playbacks of 2 Voice Prompts with different sample rates. 1 = Device does not play silence when sample rate changes.
			SRCFG									Overturn the audio data memory header sample rate. 0 = default value. Device uses the sample rate set by audio memory header for playback. 1 = device will always use the sample rate set by register 0x00 for playback.
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
2	02	**Path Control Register**	-									Reserved
			DECODE									De-Compression control. 0 = device playback path is not enabled. 1 = enable de-compression block, Ready for playback.
			SPI_IN									SPI input control. Needs to be set for SPI playback function. 0 = SPI input path is disabled 1 = SPI input path is enabled
			-									Reserved
			PWM_OUT									PWM control 0 = PWM output is disabled. 1 = PWM output is enabled.
			-									Reserved
			SPI_OUT									SPI_Output control. Needs to be set for SPI memory read function. 0 = SPI_Output path is disabled. 1 = SPI_Output path is enabled.
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
3	03	Volume Control Register	VOLC									VOLC[7:0] sets the PWM output attenuation. 0.25dB for each step. 0000 0000 = 0dB attenuation. Maximum volume. 0000 0001 = -0.25dB. 0000 0010 = -.50 dB. …. …. 1111 1111 = -63.75dB attenuation. Minimum volume.
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
4	04	Checksum Reset Register	-									Reserved
			RST_CHKSUM									Write a 1 followed by a 0 to this bit to reset the checksum calculation.
			-									Reserved
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Write only.
5	05	Thermal Control Register	-									Reserved
			TALARM_INT									If set to 1, temperature alarm generates interrupt.
			TALARM_SHTDN									If set to 1, shut down PWM output when temperature reaches threshold.
			TALARM_SEL									Set temperature alarm approximate triggering temperature, i.e. the Talarm threshold. “x” for don’t care bit. 0000 = 105ºC 0001 = 115ºC 001x = 125ºC 01xx = 135ºC 1xxx = 145ºC
			Default >>	0	0	0	1	0	0	0	0	0x10 reset value. Read/Write.
6	06	-										Reserved
7	07	-										
8	08	Mask Jump Control Register	-									Reserved
			JBSY									VM execution branches if there is an active VP playback operation.
			JGPIO5									VM execution branches if GPIO5 input is high.
			JGPIO4									VM execution branches if GPIO4 input is high.
			JGPIO3									VM execution branches if GPIO3 input is high.
			JGPIO2									VM execution branches if GPIO2 input is high.
			JGPIO1									VM execution branches if GPIO1 input is high.
			JGPIO0									VM execution branches if GPIO0 input is high.
			Default >>		0	0	0	0	0	0	0	0x00 reset value. Read/Write.
9	09	PWM Control Register	PWM_FREQ									Sets nominal PWM carrier frequency. 000 = 287 KHz 001 = 420 KHz 010 = 862 KHz 011 = 1.26 MHz 100 = 84 KHz 101 = 125 KHz 110 = 166 KHz 111 = 245 KHz
			EN_DITHER									Spread the PWM carrier frequency by dithering. 00 = No dithering … 11 = highest dithering
			-									Reserved
				0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
10	0A											Reserved
11	0B											

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
12	0C	Channel Control Register	TDM_OFF									Time Division Multiplexing control 0 = enable. enable multi- channel feature. 1 = disable multiple-channel feature.
			-									Reserved
			SPI_CMD_CH									Channel control via SPI interface 00 = Channel 0 is selected. 01 = Channel 1 is selected. 10 = Channel 2 is selected. 11 = All three channels are selected.
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. 0x000 reset value. Read/Write.
13	0D	Channel 0 Counter Control Register	CH0_CNT									Sets Channel 0 counter reload value. VM execution in Channel 0 will be blocked until Channel 0 counter counts down to 0. Total delay time = (CH0_CNT+1)*12ms.
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
14	0E	Channel 1 Counter Control Regiser	CH1_CNT									Sets Channel 1 counter reload value. VM execution in Channel 1 will be blocked until Channel 1 counter counts down to 0. Total delay time = (CH1_CNT+1)*12ms.
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
15	0F	Channel 2 Counter Control Register	CH2_CNT									Sets Channel 2 counter reload value. VM execution in Channel 2 will be blocked until Channel 2 counter counts down to 0. Total delay time = (CH2_CNT+1)*12ms.
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
16	10	Checksum Register	CHK_SUM**1**_LB									Holds checksum value chk_sum1[7:0] after checksum calculation. Write a 1 then 0 to register 0x04 bit 4 resets all Checksum registers 0x10~0x13 to 0.
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read only
17	11		CHK_SUM**1**_HB									Holds checksum value chk_sum1[15:8] after checksum calculation. Write a 1 then 0 to register 0x04 bit 4 resets Checksum registers 0x10~0x13 to 0.
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read only.
18	12		CHK_SUM**2**_LB									Holds checksum value chk_sum2[7:0] after checksum calculation. Write a 1 then 0 to register 0x04 bit 4 resets Checksum registers 0x10~0x13 to 0.
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read only.
19	13		CHK_SUM**2**_HB									Holds checksum value chk_sum2[15:8] after checksum calculation. Write a 1 then 0 to register 0x04 bit 4 resets Checksum registers 0x10~0x13 to 0.
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read only.
20	14	GPIO Trigger Channel Select 1	GPIO**3**_TRIG_CH_SEL									Assign a channel(s) in which the GPIO3 trigger VM executes. 00 = Channel 0 01 = Channel 1 10 = Channel 2 11 = Channel 0
			GPIO**2**_TRIG_CH_SEL									Assign a channel(s) in which the GPIO2 trigger VM executes. 00 = Channel 0 01 = Channel 1 10 = Channel 2 11 = Channel 0
			GPIO**1**_TRIG_CH_SEL									Assign a channel(s) in which the GPIO1 trigger VM executes. 00 = Channel 0 01 = Channel 1 10 = Channel 2 11 = Channel 0
			GPIO**0**_TRIG_CH_SEL									Assign a channel(s) in which the GPIO0 trigger VM executes. 00 = Channel 0 01 = Channel 1 10 = Channel 2 11 = Channel 0
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
21	15	GPIO Trigger Channel Select 2 Register	-									Reserved
			GPIO5_TRIG_CH_SEL									Assign a channel(s) in which the GPIO5 trigger VM executes. 00 = Channel 0 01 = Channel 1 10 = Channel 2 11 = Channel 0
			GPIO4_TRIG_CH_SEL									Assign a channel(s) in which the GPIO4 trigger VM executes. 00 = Channel 0 01 = Channel 1 10 = Channel 2 11 = Channel 0
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
22	16	Trigger Volume Control Register	VOL_DOWN_EN									0 = disable GPIO trigger to volume down feature 1 = enable GPIO trigger to volume down feature
			VOL_DOWN_GPIO_SEL									000 = GPIO0 trigger to volume down 001 = GPIO1 trigger to volume down 010 = GPIO2 trigger to volume down 011 = GPIO3 trigger to volume down 100 = GPIO4 trigger to volume down 101 = GPIO5 trigger to volume down 110, 111 = Reserved
			VOL_UP_EN									0 = disable GPIO trigger to volume up feature 1 = enable GPIO trigger to volume up feature
			VOL_UP_GPIO_SEL									000 = GPIO0 trigger to up down 001 = GPIO1 trigger to up down 010 = GPIO2 trigger to up down 011 = GPIO3 trigger to up down 100 = GPIO4 trigger to up down 101 = GPIO5 trigger to up down 110, 111 = Reserved
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
23	17	De-bounce Time Control Register										Reserved
			FAST_DEB									0 = 20 ms de-bounce time for GPIO trigger 1 = fast de-bounce time (8 ns) for GPIO trigger
												Reserved
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
24	18	-	-									Reserved
			Default >>	-	-	-	-	-	-	-	-	N/A
25	19	Output Data Control Register	-									Reserved
			GPIO_DOUT[5]									0 = GPIO5 output 0; 1 = GPIO5 output 1
			GPIO_DOUT[4]									0 = GPIO4 output 0; 1 = GPIO4 output 1
			GPIO_DOUT[3]									0 = GPIO3 output 0; 1 = GPIO3 output 1
			GPIO_DOUT[2]									0 = GPIO2 output 0; 1 = GPIO2 output 1
			GPIO_DOUT[1]									0 = GPIO1 output 0; 1 = GPIO1 output 1
			GPIO_DOUT[0]									0 = GPIO0 output 0; 1 = GPIO0 output 1
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
26	1A	Output Enable Control Register	-									Reserved
			GPIO_OE[5]									0 = disable GPIO5 output; 1 = enable GPIO5 output
			GPIO_OE[4]									0 = disable GPIO4 output; 1 = enable GPIO4 output
			GPIO_OE[3]									0 = disable GPIO3 output; 1 = enable GPIO3 output
			GPIO_OE[2]									0 = disable GPIO2 output; 1 = enable GPIO2 output
			GPIO_OE[1]									0 = disable GPIO1 output; 1 = enable GPIO1 output
			GPIO_OE[0]									0 = disable GPIO0 output; 1 = enable GPIO0 output
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
27	1B	Pull Enable Control Register	-									Reserved
			GPIO_PE[5]									0 = disable GPIO5 pull ; 1 = enable GPIO5 pull
			GPIO_PE[4]									0 = disable GPIO4 pull; 1 = enable GPIO4 pull
			GPIO_PE[3]									0 = disable GPIO3 pull; 1 = enable GPIO3 pull
			GPIO_PE[2]									0 = disable GPIO2 pull; 1 = enable GPIO2 pull
			GPIO_PE[1]									0 = disable GPIO1 pull; 1 = enable GPIO1 pull
			GPIO_PE[0]									0 = disable GPIO0 pull; 1 = enable GPIO0 pull
			Default >>	1	1	1	1	1	1	1	1	0xFF reset value. Read/Write.
28	1C	Input Data Control Register	-									Reserved
			GPIO_DIN[5]									0 = GPIO5 input data 0 ; 1 = GPIO5 input data 1
			GPIO_DIN[4]									0 = GPIO4 input data 0 ; 1 = GPIO4 input data 1
			GPIO_DIN[3]									0 = GPIO3 input data 0 ; 1 = GPIO3 input data 1
			GPIO_DIN[2]									0 = GPIO2 input data 0 ; 1 = GPIO2 input data 1
			GPIO_DIN[1]									0 = GPIO1 input data 0 ; 1 = GPIO1 input data 1
			GPIO_DIN[0]									0 = GPIO0 input data 0 ; 1 = GPIO0 input data 1
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
29	1D	Pull Select Control Register	-									Reserved
			GPIO_PS[5]									0 = pull low on GPIO5; 1 = pull high on GPIO5
			GPIO_PS[4]									0 = pull low on GPIO4; 1 = pull high on GPIO4
			GPIO_PS[3]									0 = pull low on GPIO3; 1 = pull high on GPIO3
			GPIO_PS[2]									0 = pull low on GPIO2; 1 = pull high on GPIO2
			GPIO_PS[1]									0 = pull low on GPIO1; 1 = pull high on GPIO1
			GPIO_PS[0]									0 = pull low on GPIO0; 1 = pull high on GPIO0
			Default >>	1	1	1	1	1	1	1	1	0xFF reset value. Read/Write.
30	1E	Alternate Function Control 1 Register	-									Reserved
			AF1[5]									Combined with register 0x1F to define GPIO pins functions. See Table 5-1 for GPIO function definition.
			AF1[4]									
			AF1[3]									
			AF1[2]									
			AF1[1]									
			AF1[0]									
			Default >>		0	0	0	0	0	0	0	0x00 reset value. Read/Write.
31	1F	Alternate Function Control 0 Register	-									Reserved
			AF0[5]									Combined with register 0x1E to define GPIO pins functions. See **Table** 5-2 for GPIO function definition.
			AF0[4]									
			AF0[3]									
			AF0[2]									
			AF0[1]									
			AF0[0]									
			Default >>		0	0	0	1	1	1	1	0xFF reset value. Read/Write.
32	20	R0_L	R0_L									Indirect Reference Register R0 low byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
33	21	R0_H	R0_H									Indirect Reference Register R0 high byte value
			Default >>		0	0	0	0	0	0	0	0x000 reset value. Read/Write.
34	22	R1_L	R1_L									Indirect Reference Register R1 low byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
35	23	R1_H	R1_H									Indirect Reference Register R1 high byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
36	24	R2_L	R2_L									Indirect Reference Register R2 low byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
37	25	R2_H	R2_H									Indirect Reference Register R2 high byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
38	26	R3_L	R3_L									Indirect Reference Register R3 low byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
39	27	R3_H	R3_H									Indirect Reference Register R3 high byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
40	28	R4_L	R4_L									Indirect Reference Register R4 low byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
41	29	R4_H	R4_H									Indirect Reference Register R4 high byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
42	2A	R5_L	R5_L									Indirect Reference Register R5 low byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
43	2B	R5_H	R5_H									Indirect Reference Register R5 high byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
44	2C	R6_L	R6_L									Indirect Reference Register R6 low byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
45	2D	R6_H	R6_H									Indirect Reference Register R6 high byte value
			Default >>		0	0	0	0	0	0	0	0x000 reset value. Read/Write.
46	2E	R7_L	R7_L									Indirect Reference Register R7 low byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
47	2F	R7_H	R7_H									Indirect Reference Register R7 high byte value
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
48	30	Reserved	-									Reserved
49	31	Reserved	-									Reserved
50	32	Reserved	-									Reserved
51	33	Reserved	-									Reserved
52	34	Reserved	-									Reserved
53	35	Reserved	-									Reserved

10. Application Diagrams

The following application examples are provided for reference only. Nuvoton makes no representation or warranty that such applications are suitable for the use specified. Each design must be optimized in its own system for the best performance in voice quality, current consumption, functionality etc.

The ISD2360 can be controlled and programmed through a Serial Peripheral Interface (SPI) or can operate in standalone mode by triggers applied to the device's six General Purpose Input/Output (GPIO) pins.

10.1 SPI Application under MCU Control

A standard four-wire Serial Peripheral Interface (SPI) is used for communication between the ISD2360 and the host. The interface consists of an active low slave-select (SSB), a serial clock (SCLK), a data input (Master Out Slave In - MOSI), and a data output (Master In Slave Out - MISO). The RDY/BSYB signal (pin) is available for some transactions requiring data flow control.

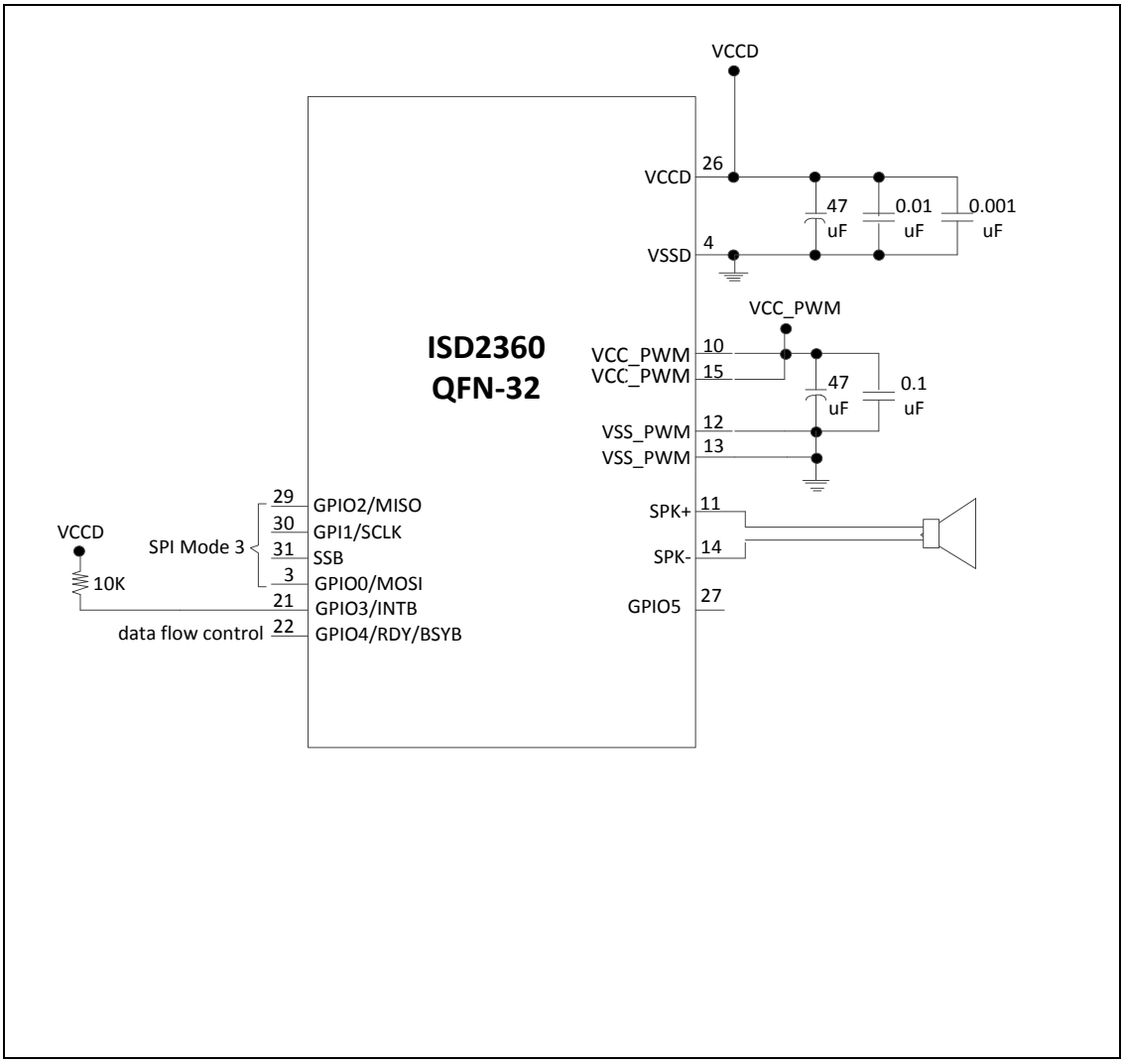


Figure 10-1 SPI Application under MCU Control

10.2 GPIO Trigger Standalone Application

The ISD2360 can operate in standalone mode by triggers applied to the device's six General Purpose Input/Output (GPIO) pins. Voice Prompt and Voice Macro commands facilitate fast programming.

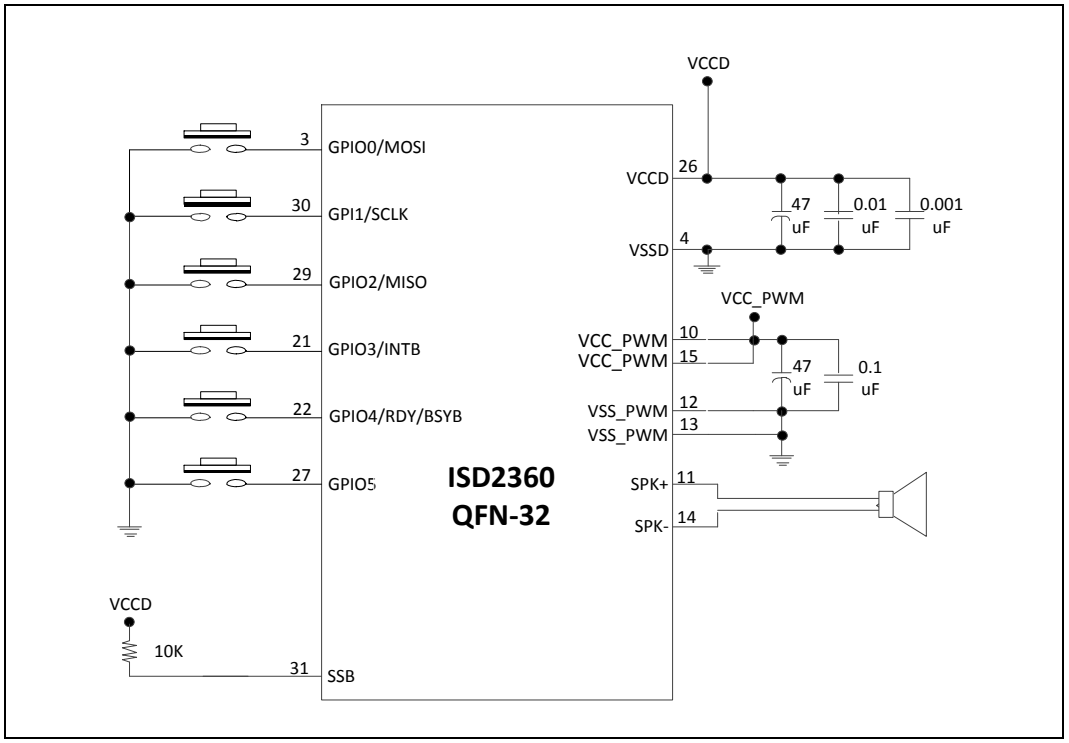


Figure 10-2 GPIO Trigger Standalone Application

11. Electrical Characteristics

11.1 Absolute Maximum Ratings

DESCRIPTION	SYMBOL	CONDITION	MIN	MAX	UNITS
DC Power Supply	V_{CCD}	$V_{CCD} – V_{SSD}$	-0.3	+6.0	V
	V_{CCPWM}	$V_{CCPWM} – V_{SSPWM}$	-0.3	+6.0	V
Digital Input Voltage	DV_{IN}	DV_{IN} - V_{SSD}	$V_{SSD} – 0.3$	$V_{CCD} + 0.3$	V
Junction Temperature	T_J	-	-40	+125	°C
Storage Temperature	T_{ST}	-	-65	+150	°C

CAUTION: Do not operate at or near the maximum ratings listed for extended period of time. Exposure to such conditions may adversely influence product reliability and result in failures not covered by warranty. These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.

11.2 Operating Conditions

Table 11-1 Operating Conditions (Industrial Packaging)

Condition	Value
Operating temperature range (Case temperature)	-40° C to +85° C
Supply voltage (V_{DD}) [1]	+2.4 V to +5.5 V
Ground voltage (V_{SS}) [2]	0 V
Digital input voltage (DV_{IN})	0 V to 5.5 V
Voltage applied to any pins	(V_{SS} –0.3 V) to (V_{DD} +0.3 V)

Notes: [1] $V_{DD} = V_{CCA} = V_{CCD}$
[2] $V_{SS} = V_{SSA} = V_{SSD}$

11.3 AC Paramaters

11.3.1 Internal Oscillator

Parameter	Symbol	Min	Typ	Max	Units	Conditions
Sample Rate with Internal Oscillator	F_{smax}	-1%	32 kHz	+1%	kHz	VDD = 3 V. at room temperature

11.3.2 Speaker Outputs

Parameter	Symbol	Min	Typ[1]	Max	Units	Conditions
Output Power	P_{OUT_SPK} VCC=5.0		1		W	Load 8 Ω [2]
THD, Memory to SPK+/SPK-	THD %		<1%			Load 8 Ω [2]
Minimum Load Impedance	$R_{L(SPK)}$	4	8		Ω	

Notes: [1] Conditions V_{CC}=5V, T_A=25°C unless otherwise stated.
[2] Based on 12-bit PCM.
[3] All measurements are C-message weighted.

11.3.3 DC Parameters

Parameter	SYMBOL	min	typ [1]	Max	Units	Conditions
Supply Voltage	**VDD**	**2.7**		**5.5**	**V**	
Input Low Voltage	VIL	VSS-0.3		0.3xVDD	V	
Input High Voltage	VIH	0.7xVDD		VDD	V	
Output Low Voltage	VOL	VSS-0.3		0.3xVDD	V	IOL = 1mA
Output High Voltage	VOH	0.7xVDD		VDD	V	IOH = -1mA
Pull-up Resistance	RPU		50		kΩ	
Pull-down Resistance	RPD		10		kΩ	
INTB Output Low Voltage	VOH1			0.4	V	
Playback Current	IDD_Playback		3		mA	No Load [2]
Standby Current	ISB		<1	10	µA	VDD= 3.6V
Input Leakage Current	IIL			±1	µA	Force VDD

Notes: [1] Conditions V_{DD}=3V, T_A=25°C unless otherwise stated
[2] To calculate total current, add load dissipation into application specific load.

11.3.4 SPI Timing

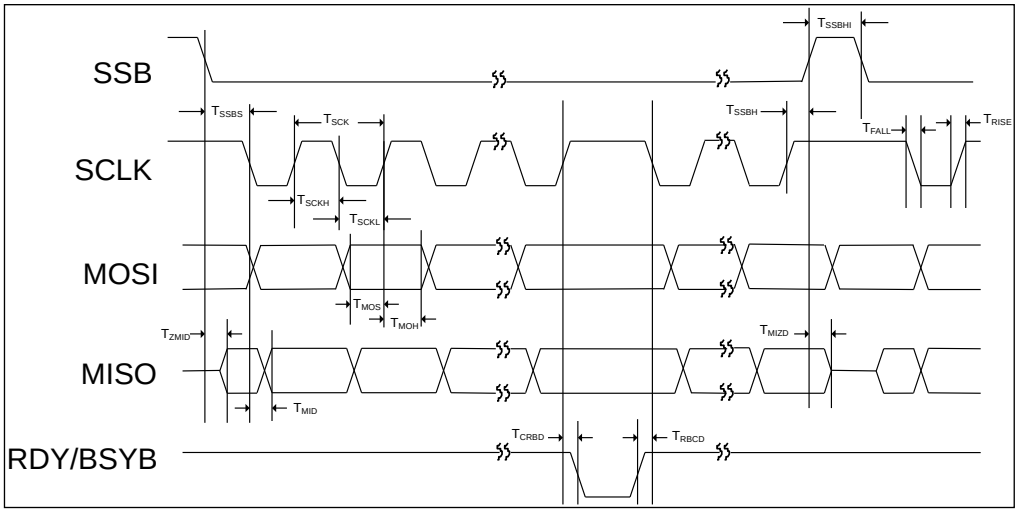


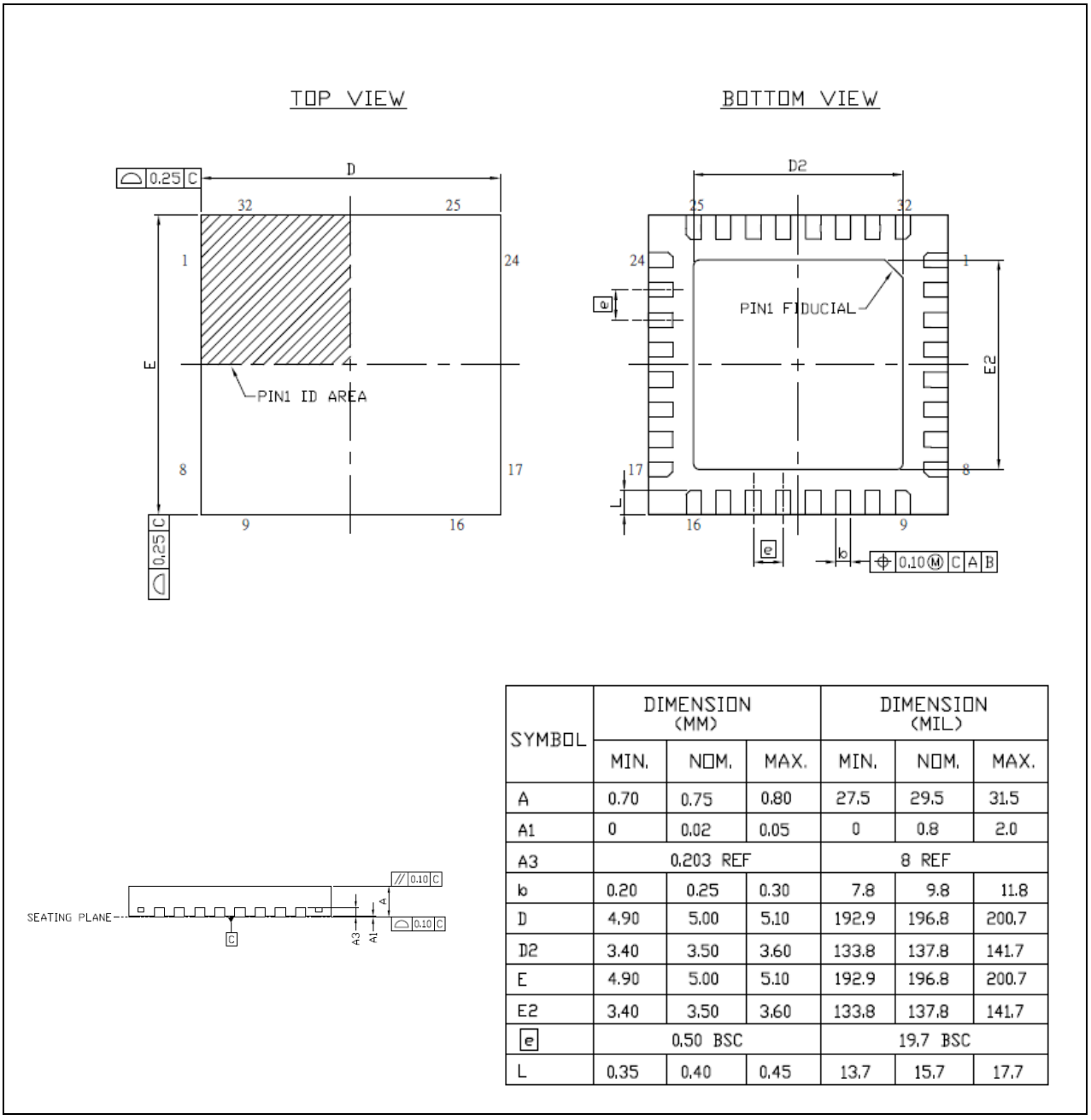
Figure 11-1 SPI Timing

Table 11-2 SPI Timing

Symbol	Description	Min	Typ	Max	Unit
T_{SCK}	SCLK Cycle Time	60	---	---	ns
T_{SCKH}	SCLK High Pulse Width	25	---	---	ns
T_{SCKL}	SCLK Low Pulse Width	25	---	---	ns
T_{RISE}	Rise Time for All Digital Signals	10	---	---	ns
T_{FALL}	Fall Time for All Digital Signals	10	---	---	ns
T_{SSBS}	SSB Falling Edge to first SCLK Falling Edge Setup Time	30	---	---	ns
T_{SSBH}	Last SCLK Rising Edge to SSB Rising Edge Hold Time	30 ns	---	50 µs	---
T_{SSBHI}	SSB High Time between SSB Lows	20	---	---	ns
T_{MOS}	MOSI to SCLK Rising Edge Setup Time	15	---	---	ns
T_{MOH}	SCLK Rising Edge to MOSI Hold Time	15	---	---	ns
T_{ZMID}	Delay Time from SSB Falling Edge to MISO Active	--	--	12	ns
T_{MIZD}	Delay Time from SSB Rising Edge to MISO Tri-state	--	--	12	ns
T_{MID}	Delay Time from SCLK Falling Edge to MISO	---	---	12	ns
T_{CRBD}	Delay Time: SCLK Rising Edge to RDY/BSYB Falling Edge	--	--	12	ns
T_{RBCD}	Delay Time: RDY/BSYB Rising Edge to SCLK Falling Edge	0	--	--	ns

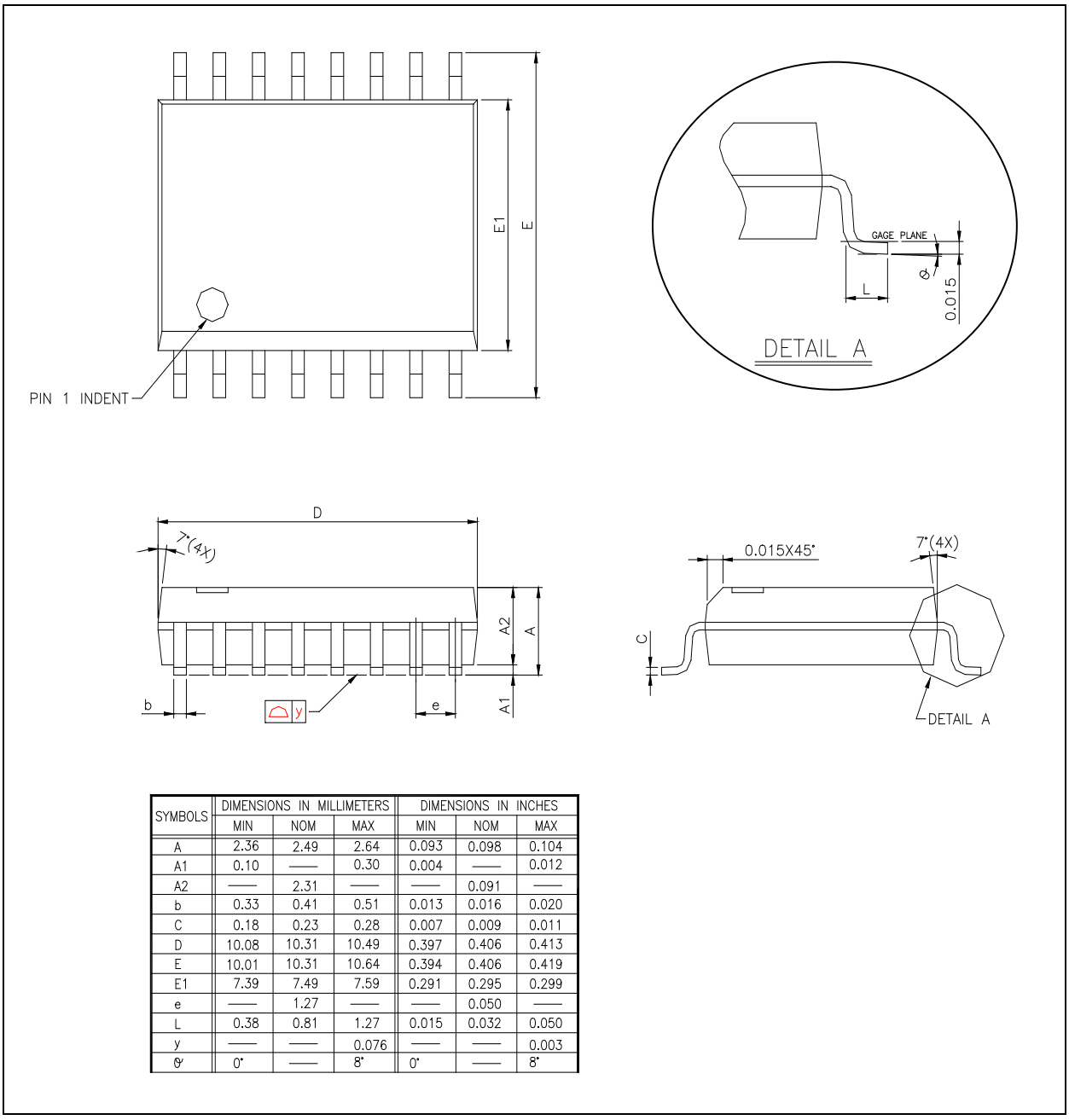
12. Package Dimensions

The ISD2360 is available in a QFN 32-Lead package, as shown in Figure 12-1 and an SOP 16-Lead package, as shown in Figure 12-2.



SYMBOL	DIMENSION (MM)			DIMENSION (MIL)		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.75	0.80	27.5	29.5	31.5
A1	0	0.02	0.05	0	0.8	2.0
A3	0.203 REF			8 REF		
b	0.20	0.25	0.30	7.8	9.8	11.8
D	4.90	5.00	5.10	192.9	196.8	200.7
D2	3.40	3.50	3.60	133.8	137.8	141.7
E	4.90	5.00	5.10	192.9	196.8	200.7
E2	3.40	3.50	3.60	133.8	137.8	141.7
e	0.50 BSC			19.7 BSC		
L	0.35	0.40	0.45	13.7	15.7	17.7

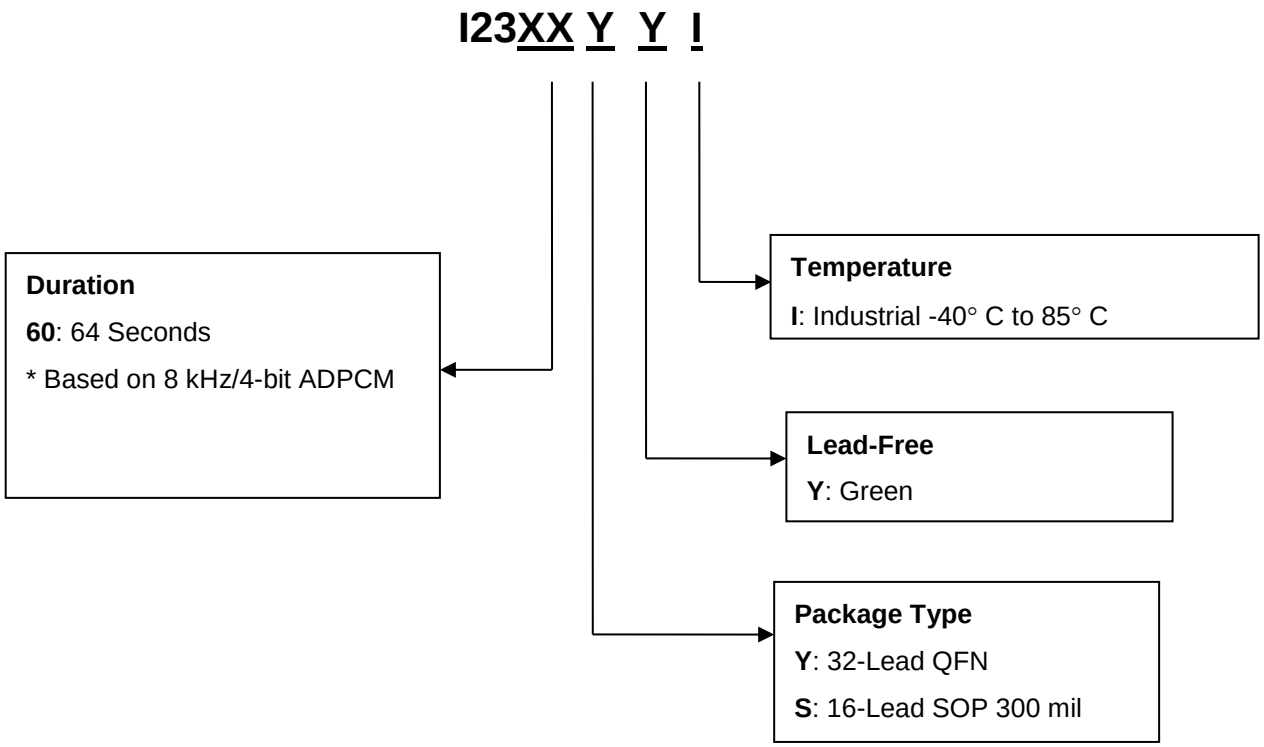
Figure 12-1 QFN 32-Lead Package



SYMBOLS	DIMENSIONS IN MILLIMETERS			DIMENSIONS IN INCHES		
	MIN	NOM	MAX	MIN	NOM	MAX
A	2.36	2.49	2.64	0.093	0.098	0.104
A1	0.10	——	0.30	0.004	——	0.012
A2	——	2.31	——	——	0.091	——
b	0.33	0.41	0.51	0.013	0.016	0.020
C	0.18	0.23	0.28	0.007	0.009	0.011
D	10.08	10.31	10.49	0.397	0.406	0.413
E	10.01	10.31	10.64	0.394	0.406	0.419
E1	7.39	7.49	7.59	0.291	0.295	0.299
e	——	1.27	——	——	0.050	——
L	0.38	0.81	1.27	0.015	0.032	0.050
y	——	——	0.076	——	——	0.003
θ	0°	——	8°	0°	——	8°

Figure 12-2 SOP 16-Lead Package

13. Ordering Information



14. Revision History

Version	Date	Description
1.0	December 21, 2011	Initial draft
1.1	July 05, 2012	First release
1.11	August 23, 2012	SOP 16-Lead device: Replaced pin diagram, added pin descriptions. Commercial Parts Operating Conditions removed. Linguistic and format changes.
1.12	March 05, 2013	SPI T_{rise}, T_{fall} moved to under Min category; SNR spec removed.
1.13	September 05, 2013	Fixed SOP package pin out order.
1.14	November 20, 2014	Added section 6.6.2.2: description on pulling CSB low .vs. trigger
1.15	August 3, 2016	Add Absolute Maximum Ratings. Fix QFN package dimension. Update VCCD, VCC_PWM description.