ISD ChipCorder®
ISD2361
Design Guide

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

Table of Contents-

List of Figures-

List of Tables-

1 General Description

Overview

ISD2361 is a playback only Digital ChipCorder® family product based on flash storage. The device features both internal and external flash support, digital de-compression, comprehensive memory management, integrated audio signal path with up to 3 channel concurrent playback and a Class D speaker driver capable of delivering 1W power output.

ISD2361 requires no external clock sources or components to operate. It can operate under SPI mode, or operate in stand-alone by detecting on level transition on its GPIO pins.

ISD2361YYI has 2Mbit non-volatile flash built-in, and can support external NOR serial flash up to 2G bit.

ISD2361SYI has 2Mbit non-volatile flash built-in only and cannot support external NOR serial flash.

Features

- Performance Enhancements
  - Support external flash **(ISD2361YYI)**
  - GPIO trigger reliability
  - Better DPWM output SNR
  - Audio output power efficiency
- Duration
  - 64 seconds based on 8kHz/4bit ADPCM in 2Mbit of internal flash storage
  - **ISD2361YYI with external memory** – total addressable space is 2Gbit (1092minutes)
- 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
  - Each channel has independent counter which enables user micro-management on VM execution
  - Mask Jump for branch execution; based on internal register or external GPIO pin
- Chip 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
- Compression Algorithms
  - µ-Law: 6, 7 or 8 bits per sample
  - Differential µ-Law: 6, 7 or 8 bits per sample
  - PCM: 8, 10, 12 or 16 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, ±1% deviation at room temperature
- Output
  - PWM: Class D speaker driver to direct drive speaker or buzzer
  - Delivers:
    - 4Ω load: 440mW@3.3V; 1.2W@5V; 1.5W@5.5V
    - 8Ω load: 330mW@3.3V; 800mW@5V; 1W@5.5V
- I/O
  - Total 10 General Purpose I/O (GPIO) pins
  - 6 out of the 10 GPIO pins multiplexed with SPI interface and capable of trigger play
  - 4 GPIO pins available when no external flash connected
  - SPI interface: MISO, MOSI, SCLK, SSB for commands and digital audio data
- Internal Flash Storage
  - Built-in 2Mbit of storage
  - Fast programming time (20µs/byte)
  - Erase sector size 512-byte, sector erase time 5ms
  - Integrated memory checksum calculation for fast verification
  - Endurance >100K cycles; retention >10 years
- External Flash Storage
  - **ISD2361YYI can support external flash**
    ISD2361SYI cannot support external flash
  - Supports NOR serial flash command set (ISD2361YYI)
  - Addressable space up to 2Gbit (ISD2361YYI)
- Operating Voltage: 2.4V-5.5 V
- Packages, Green:
  - QFN 32-Lead
  - SOP 16-Lead 300 mil
- Temperature Options:
  - Industrial: -40°C to 105°C[1]

[1] Note: TBD – maximum operating temperature to be further characterized.

2 Pin Configurations

2.1 Pin Diagrams

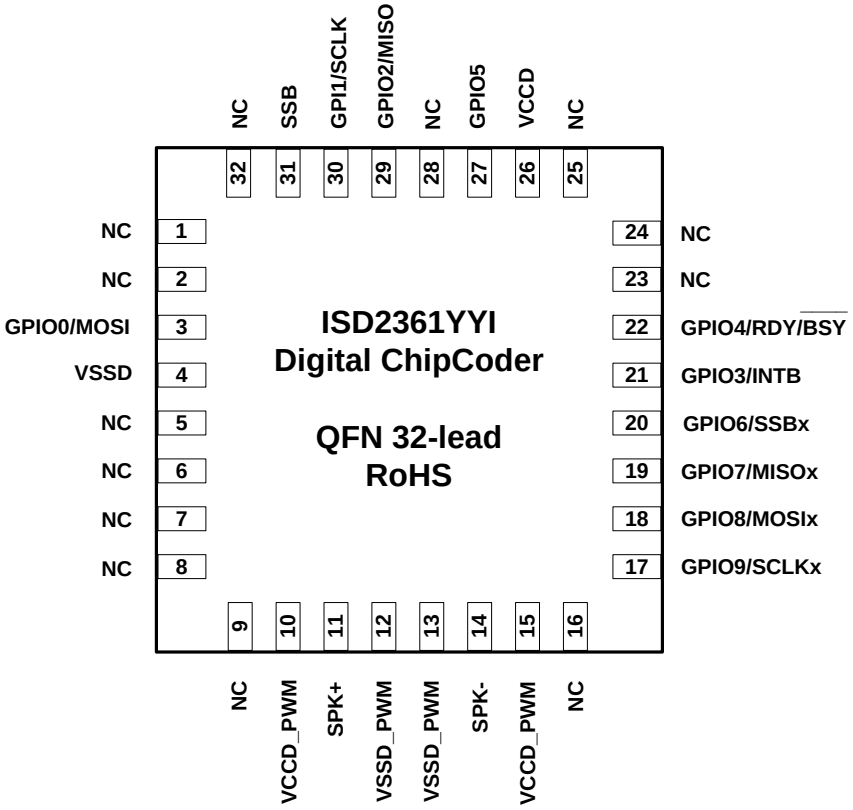


Figure 2-1 ISD2361 QFN 32-Lead Package

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.

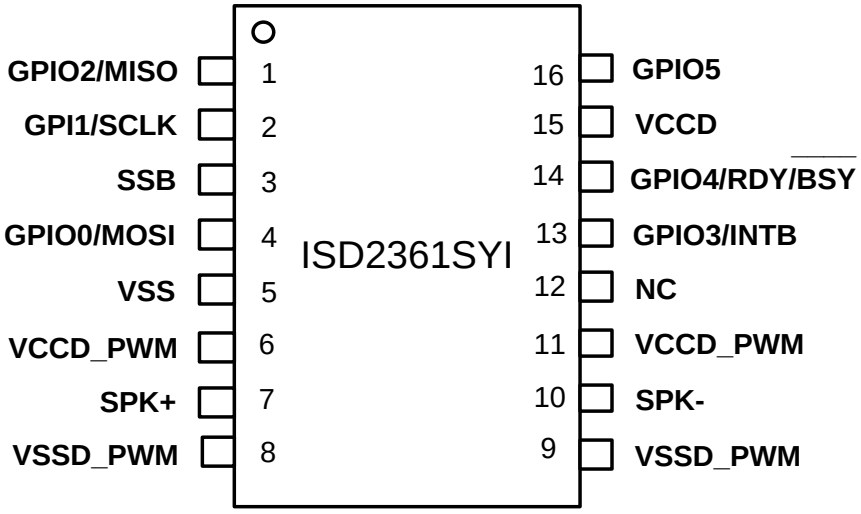


Figure **2-2** ISD2361 SOP 16-Lead 300 mil Package

2.2 Pin Descriptions

The ISD2361 pin description is shown in Table 2.2-1 below.

QFN Pin #	SOP Pin #	Pin Name	I/O	Function
1		NC		This pin should be left unconnected.
2		NC		This pin should be left unconnected.
3	4	GPIO0/MOSI	I/O	Master-Out-Slave-In. Serial input to the ISD2361 from the host. Can be configured as a general purpose I/O pin.
4	5	VSSD		Digital Ground.
5		NC		This pin should be left unconnected.
6		NC		This pin should be left unconnected.
7		NC		This pin should be left unconnected.
8		NC		This pin should be left unconnected.
9		NC		This pin should be left unconnected.
10	6	VCCD_PWM	I	Digital Power for the PWM Driver.
11	7	SPK+	O	PWM driver positive output. This SPK+ output, together with SPK- pin, provide a differential output to drive 8Ω speaker or buzzer. During power down this pin is in tristate.
12	8	VSSD_PWM	I	Digital Ground for the PWM Driver.
13	9	VSSD_PWM	I	Digital Ground for the PWM Driver.
14	10	SPK-	O	PWM driver negative output. This SPK- output, together with SPK+ pin, provides a differential output to drive 8Ω speaker or buzzer. During power down this pin is tristate.
15	11	VCCD_PWM	I	Digital Power for the PWM Driver.
16		NC		This pin should be left unconnected.
17		GPIO9/SCLKx	I/O	- External SPI interface SCLK pin if external flash is connected. I2361 is the master. - Can be configured as general purpose I/O pin if external flash is not connected.
18		GPIO8/MOSIx	I/O	- External SPI interface MOSI (DOUT) pin if external flash is connected. I2361 is the master. - Can be configured as general purpose I/O pin if external flash is not connected.
19		GPIO7/MISOx	I/O	- External SPI interface MISO (DIN) pin if external flash is connected. I2361 is the master. - Can be configured as a general purpose I/O pin if external flash is not connected.
20		GPIO6/SSBx	I/O	- External SPI interface Slave Select pin if external flash is connected. I2361 is the master. - Can be configured as general purpose I/O pin if external flash is not connected.

QFN Pin #	SOP Pin #	Pin Name	I/O	Function
21	13	GPIO3/INTB	I/O	Active low interrupt request pin. This pin is an open-drain output. Can be configured as a general purpose I/O pin.
22	14	GPIO4/RDY/$\overline{\text{BSY}}$	I/O	An output pin to report the status of data transfer on the SPI interface. “High” indicates that ISD2361 is ready to accept new SPI commands or data. Can be configured as a general purpose I/O pin.
23		NC		This pin should be left unconnected.
24		NC		This pin should be left unconnected.
25		NC		This pin should be left unconnected.
26	15	VCCD	I	Digital Power.
27	16	GPIO5	I/O	Can be configured as a general purpose I/O pin.
28		NC		This pin should be left unconnected.
29	1	GPIO2/MISO	I/O	Master-In-Slave-Out. Serial output from the ISD2361 to the host. This pin is in tristate when SSB=1. Can be configured as a general purpose I/O pin.
30	2	GPI1/SCLK	I	Serial Clock input to the ISD2361 from the host. Can be configured as a general purpose input-only pin.
31	3	SSB	I	Slave Select input to the ISD2361 from the host. When SSB is low 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 be left unconnected.

Table 2.2-1 ISD2361 pin description

*note: for QFN-32 package center pad underneath should be connected to VSSD.

3 Block Diagram

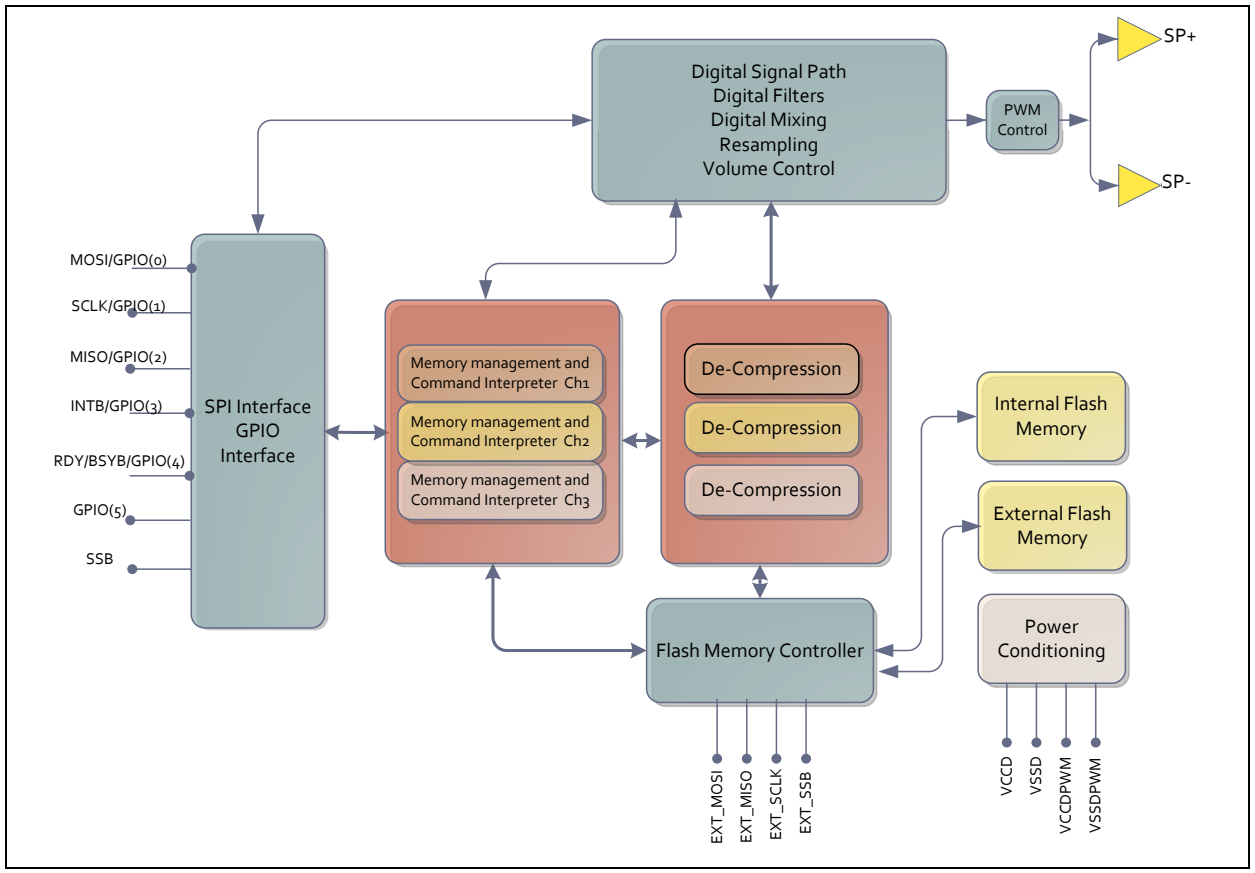


Figure 3-1 ISD2361YYI Block Diagram

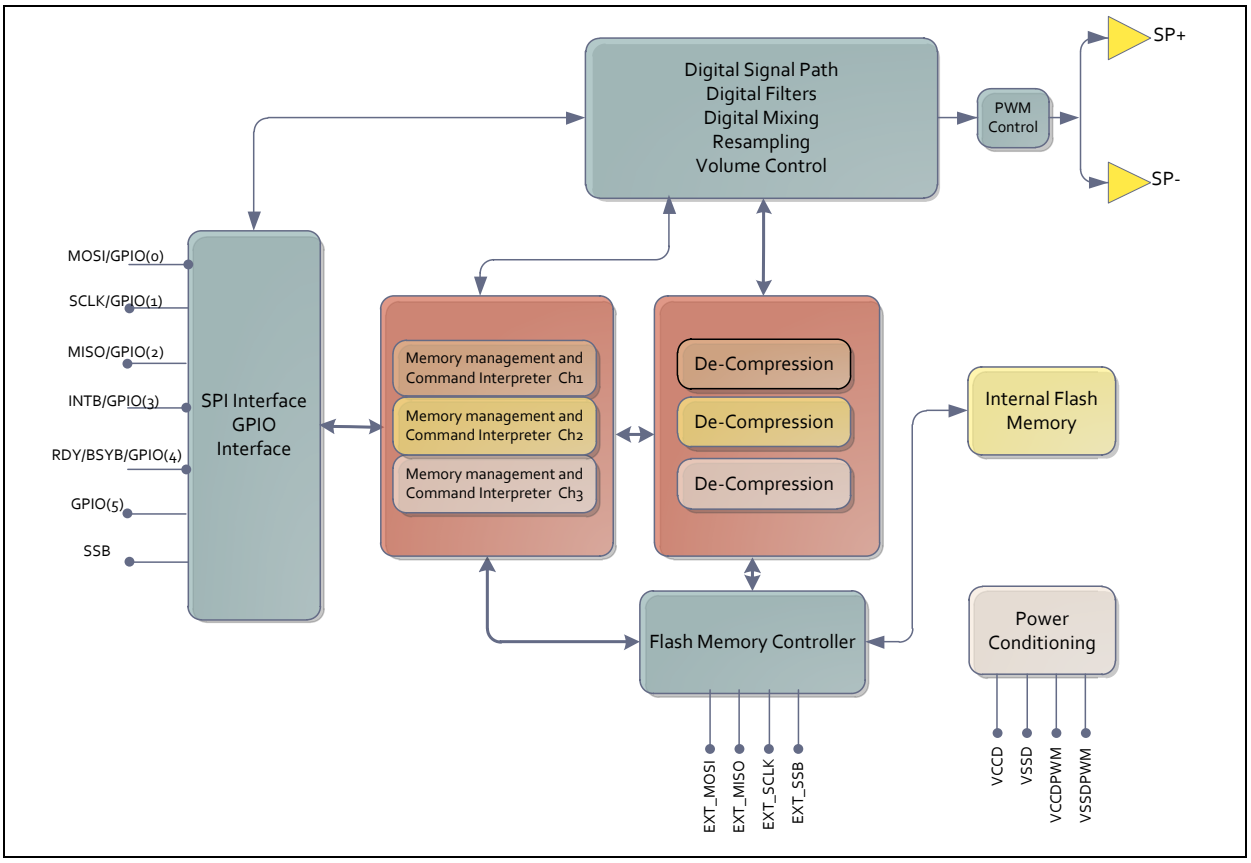


Figure 3-2 ISD2361SYI Block Diagram

4 Device Status

The ISD2361 status can be acquired by querying the *Device Status Register* and the *Interrupt Status Register*. After receiving a READ_STATUS command, the ISD2361 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 ISD2361 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 ISD2361 continually updates its device status register bits. Upon completion of an SPI command or a Voice Macro command, the ISD2361 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 ISD2361.The details of the device status bits are shown in Table 4.1-1**.**

Table 4.1-1 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 ISD2361 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 4.2-1.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 4.2-1 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 ISD2361 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 ISD2361 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 ISD2361's device ID is a 4-byte long value: *0x06 EF 20 60*. The ISD2361 device responds to SPI READ_ID command by sending out its 4-byte device identification value.

Of these 4-byte device ID, the first byte is the ISD digital ChipCorder family ID which is 0x06 for ISD2361. The following three bytes are a JEDEC compliant code indicating manufacturer ID, memory type ID and memory size byte, with values of 0xEF, 0x20 and 0x60 respectively for ISD2361 device.

5.2 GPIO Configuration

5.2.1 GPIO Pin Function Definition

ISD2361 has ten GPIO pins in total. GPIO[5:0] pins can be configured as four different function modes listed in Table 5.2-2, determined by register AF1 and AF0 pin combination listed in Table 5.2-1. By default, GPIO[9:6] are SPI pins interfacing with external flash. GPIO[7:6] can be set in tristate with pulled-high available, and GPIO[9:8] can be configured as GPIO pins if the external flash is not connected.

Regardless of GPIO/MOSI, GPI1/SCLK and GPIO2/MISO configured function modes, they will immediately become SPI pins whenever SSB pin becomes low. When SSB pin becomes high, they will resume their function mode determined by AF1 and AF0.

GPIO4 (RDY/BSYB) is required to be configured as a RDY/BSYB pin when read or write the internal flash. For example, before read out/write into internal flash user should write AF1[4] =0 and AF0[4] =1 in advance. For external flash read/write, there is no such requirement.

The Alternate Function Control Registers, AF1 (0x1E) and AF0 (0x1F), combine to define the specific function mode GPIO[5:0]. See Table 2.2-1 and Table 5.2-2

for GPIO0~6 pin function configuration.

Table 5.2-1 AF1/AF0 Bit Combination for GPIO0~6 Pin Function configuration

	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 5.2-2 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: GPI1 is input only when used as GPIO pin. GPIO5 does not have an alternate function.

See Register Operation for more detailed description on GPIO[9:0].

5.2.2 Special 3 V Registers

The ISD2361 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.

In power down typical current consumption is 2.5µA, and the 3v register contents will be kept in effect. Because the 3v registers always maintain the trigger condition as long as power is on, the ISD2361 keeps detecting a GPIO trigger during power down.

5.2.3 Release External Flash SSBx

The bit6 of register 0x1A configures the function mode of GPIO6/SSBx pin. By default, reg0x1A[6] is 0 which means this pin is controlled by SPIx module and functions as SSB control for external flash.

When power down, if reg0x1A[6] is 0 then SSBx will be pulled low. At this time if the external flash is still powered on, it will consume considerable idle current since it's under idle state with SSB pulled low. To avoid this situation, it is desired to release GPIO6/SSBx pin to tristate (with pulled high). To do so, user can first write reg0x1A[6] as 1 – to release GPIO6/SSBx pin as tristate pin, and write reg0x1B[6] as 1 – to enable SSBx internally pulled high, then power down the device.

Please note that register 0x1A and 0x1B are special 3V registers and can keep their values during power down. To recover the default function of SSBx pin, user can either reset device or write reg0x1A[6] as 0.

5.2.4 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 .

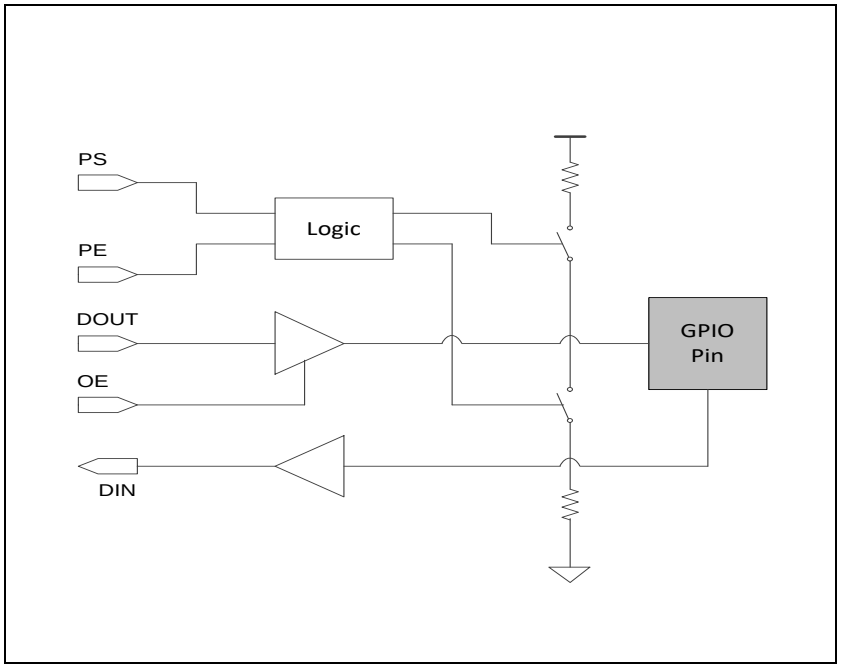


Figure 5-1 GPIO Pad Structure

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 ISD2361 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 ISD2361 device will execute a wake-up event.

5.3 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. The possible path combinations are:

- MEMORY → DECOMPRESS → SPKR (Playback to Speaker), see Figure 5.3-1;
- MEMORY → DECOMPRESS → SPI_OUT (SPI Playback);
- SPI_IN → DECOMPRESS → SPKR (SPI Decode to Speaker), see Figure 5.3-2.

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.

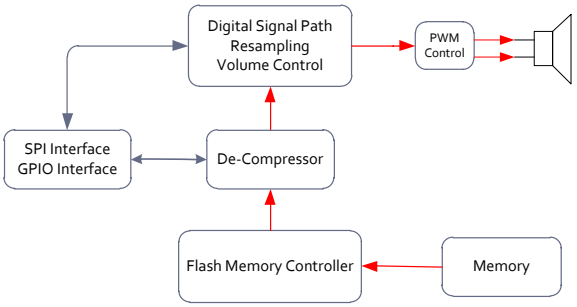


Figure 5-2 ISD2361 playback path

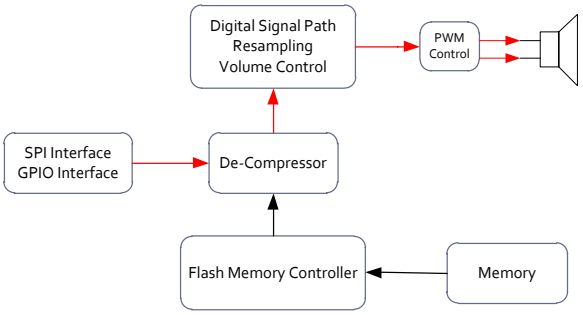


Figure 5-3 ISD2361 direct playback from SPI

The memory block in the diagram refers to both internal and external flash memory. If the external flash is presented, when memory addressing exceeds the internal flash memory boundary, the hardware will automatically switch to address external flash memory. This process is transparent to the user. User sees the total memory space as a whole seamless one. The total addressing space is 2Mbit of the embedded internal flash plus the size of the external flash.

5.4 Device Checksum

The ISD2361 has hardware checksum feature. That is the ISD2361 device can respond to SPI CHECKSUM command, calculate the checksum from the address 0x00 – beginning of the internal memory to specified end address within its internal flash memory.

The ISD2361 hardware checksum feature only calculates the checksum for its internal 2Mbit flash. The user is responsible for checking the external SPI flash content integrity.

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 5.4-1.

Table 5.4-1 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 ISD2361 Fletcher-32 calculation routine is shown in Table 5.4-2:

Table 5.4-2 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;
}
```

5.5 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 5.5-1 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 5.5-2.

Table 5.5-2 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

5.6 RDY/BYSB flow control for Digital Read/Write

The ISD2361 is equipped with an internal FIFO with 4-byte depth which governs digital read/write operations flow. For digital read operations, the host controller should only read from the SPI when the FIFO has data, i.e. when RDY/BSYB is high. For digital write operations, the host controller should only write into ISD2361 via SPI when the FIFO has vacant space, i.e. when RDY/BSYB is high.

The GPIO4/RDY/BSYB pin can be configured as RDY/BSYB pin to reflect this internal FIFO status. In Status Register bit[6] DBUF_RDY bit indicates the internal FIFO status as well. User can either poll DBUF_RDY bit or monitor the RDY/BSYB pin level to achieve the flow control for digital read/write operations. To configure GPIO4/RDY/BSYB pin as RDY/BSYB pin, write AF1 bit4 as 0 and AF0 bit4 as 1.

The following ISD2361 operations require flow control:

DIG_READ	Read from memory.
DIG_WRITE	Programming the ISD2361 internal flash.
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 from SPI interface to ISD2361, allow the ISD2361 to de-compress the data and then play it back through PWM output.

5.7 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 ISD2361 GPIO trigger. Writing one to this bit enables the fast de-bounce feature, and reduce the de-bounce time ~150 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.

5.8 Thermal Shutdown

The ISD2361 device PWM driver can deliver a maximum output power at about 1 Watt. With the increase of the output power, heat generated also accumulates. To prevent the accumulating heat from burning out the device, a thermal shutdown feature is implemented. When Thermal shutdown is enabled, the ISD2316 device automatically shuts down the PWM output when the temperature reaches the shutdown threshold.

The ISD2361 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 ISD2361 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 Figure 9-1 Register Operations.

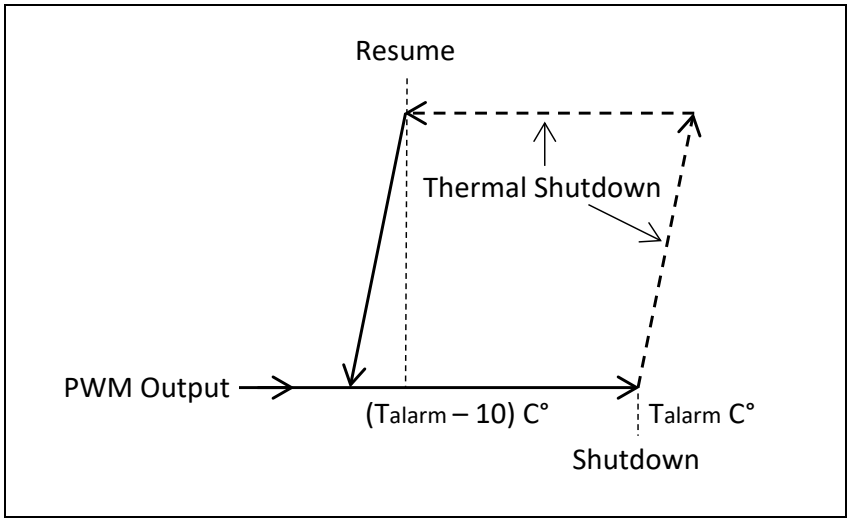


Figure 5-4 PWM Output Thermal Shutdown

6 Operational Description

6.1 Overview

Normally the ISD2361 device needs to be programmed before being put into use in the field. A GUI software, *ISD-VPE2361* (Voice Prompt Editor for ISD2361 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 perform in-system programming in field.

Typically, user can follow the steps below to develop an ISD2361 application system:

- Determine whether it is microcontroller application which relies on SPI to control the device, or standalone allocation which relies on GPIO Trigger without microcontroller involved.
- create an *ISD-VPE2361* project, import 16bit PCM wave files into the project, compile the project, burn the image into device internal flash memory - and external flash memory if needed, then evaluate the sound effect on a demo system.
- Integrate the device into target system. The ISD2361 device and its external flash can be either pro-programmed, or be programmed in the run.

6.2 Audio Storage

ISD2361 comes equipped with built-in 2-Mbit flash memory, which consists of 512 sectors with a sector size of 512-byte; that in total can give 64 seconds playback duration if using 8 kHz ADPCM4 compression.

ISD2361 also supports external flash up to 2Gbits. When 2Mbit internal flash is not bigger enough to hold application image, user can choose to connect external flash using GPIO6-9. Thus gives the total memory size: 2Mbit internal plus external flash size. If external flash is connected, both the ISD2361 internal flash and the external flash need to be programmed for system to function properly.

To program the internal flash, SPI command *DIG_WRITE* is to be used. *DIG_WRITE* command only programs the internal flash. To program the external flash, user should follow the format of OP_EXT + Flash SPI command to interface with external flash.

The *ISD-VPE2361* software allows user to generate the application image file based on the project. If the application image size is less or equal to 2Mbits, only internal image file will be generated. If the image size is greater than 2Mbits, then three image files are to be generated: one complete image file, one internal image file which contains the 2Mbit data from starting address 0x0000, and one image file for external flash memory which contains the application data starting from address 0x40000 – after the first 2Mbit, to the end.

For example, for a VPE project named as ***myProject***:

- If the image size is no greater than 2Mbit, then two image files will be generated: ***my_project.mem***, and ***my_project_Int.mem***, they are the same. And they should be programmed into ISD2361 internal flash.
- If the image size is greater than 2Mbit, then three images files will be generated: ***my_project.mem*** is the complete image file; ***my_project_Int.mem*** contains the image data for ISD2361 internal flash***; my_project_Ext.mem*** contains the image data after 2Mbit boundary and should be programmed into external flash memory.

These image files all reside in VPE project folder after compilation.

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 ISD2361 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-VPE2361* Software, the user should avoid using sector 0.

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

6.3 Programming ISD2361 Internal Flash

The ISD2361 device is a playback only device. The ISD2361 internal flash needs to be pre-programmed before put it can be put in use.

The ISD2361 internal flash is organized as 512 sectors with 512-byte each. User should erase the memory first before programming. DIG_WRITE command is to be used to program the internal flash.

During the programming, user need to monitor the RDY/BYSB pin status or poll the device status before write each byte, i.e. only write one byte when device is ready, either RDY/BSYB pin is high, or DBYF_RDY bit in Device Status register is set. However, polling Device Status register is not recommended because of the huge flow control overhead. To monitor the RDY/BSYB pin, GPIO4/RDY/BSYB pin needs to be configured as RDY/BSYB pin first, via its AF1 and AF0 bits.

If either monitoring RDY/BSYB pin or polling device status is not practicable, user can also apply write-then-wait strategy to program the device. For example: write in 4-byte data, then wait for 100-200ns to make sure device is ready to take next batch 4 bytes. The internal FIFO size is 4-byte.

6.4 Programming ISD2361 External Flash

The ISD2361 supports external flash up to 2Gbit. Commonly available NOR flash from vendors such as Windbond, Numonyx, Macronix etc., as long as the flash SPI command set are compatible[2], all can be used by ISD2361 for its external audio storage.

The ISD2361 implements a special SPI command OP_EXT to interface with external NOR serial flash. OP_EXT is a pass down command. During an SPI transaction, once OP_EXT command is received the ISD2361 literally connects the ISD2361 SPI interface with host controller to the SPI interface with the external flash, hence all the following command bytes after OP_EXT within the same SPI transaction will be sent to the external flash. This way gives the host controller the direct control of external flash.

User should follow the format below to construct a valid Pass Down command for ISD2361 to interface with its external flash:

OP_EXT + a valid NOR Serial Flash SPI command[3]

OP_EXT is the pass down command and it should be used in conjunction with a valid NOR serial flash SPI command. User should utilize the Pass Down command, send SPI commands directly to the external flash to achieve the external flash programming.

Table 6.4-1 listed some sample flash SPI commands and their corresponding ISD2361 Pass Down command are shown in Table 6.4-2.

[2] Compatible with Winbond NOR serial flash due to test coverage.

[3] Refer to NOR Serial SPI flash Datasheet for valid flash SPI commands.

SPI Flash Command	Byte1 (CODE)	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Write Enable	06h					
Write Disable	04h					
Read Status Register -1	05h	(S7-S0)				
Page program	02h	A23-A16	A15-A8	A7-A0	(D7-D0)	
Sector Erase(4KB)	20h	A23-A16	A15-A8	A7-A0		
Read Data	03h	A23-A16	A15-A8	A7-A0	(D7-D0)	

Table 6.4-1 Sample NOR Serial SPI commands

ISD2361 External Flash SPI Command	Byte1 (Pass Down OP Code)	Byte 2 (SPI Code)	Byte 3	Byte 4	Byte 6	Byte 7
Write Enable	38h	06h				
Write Disable	38h	04h				
Read Status Register -1	38h	05h	(S7-S0)			
Page program	38h	02h	A23-A16	A15-A8	A7-A0	(D7-D0)
Sector Erase(4KB)	38h	20h	A23-A16	A15-A8	A7-A0	
Read Data	38h	03h	A23-A16	A15-A8	A7-A0	(D7-D0)

Table 6.4-2 Sample ISD2361 SPI command for interfacing external flash

6.5 3-Byte vs 4-Byte Addressing Mode

To correctly address the external flash, the ISD2361 and its external flash need to be in synch with the same address mode. The ISD2361 default configuration is 3-byte address mode, which gives maximum 128Mbit addressing capability. When the external flash size is greater than 128Mbit, the ISD2361 needs to be configure as 4-byte mode to address the whole range of memory: internal 2Mbit plus external 256Mbit or above.

User can set ISD2361 CFG7[7] FOUR_BYTE_ADDR bit to enable the 4-byte addressing mode, or clear FOUR_BYTE_ADDR bit to switch to 3-byte addressing mode.

Please be aware that typically NOR SPI flash by default is at 3-byte address mode. User can change external flash to 4-byte mode by either sending change-to-4byte-mode SPI flash command, or modify the SPI flash non-volatile configuration bytes so to enter 4-byte addressing mode when power on.

6.6 Audio Compression and De-Compression

ISD2361 hardware performs only audio de-compression. Audio compression relies on the supplied software – ISD-VPE2361 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, 12 or 16 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, ISD2361 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..

6.7 System Voice Macro Flow Chart

The ISD2361 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 ISD2361 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 ISD2361 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 ISD2361. 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 ISD2361 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 ISD2361 device will or will not execute the Wakeup Voice Macro before executing the triggering pin associated Voice Macro, as shown in Figure 6.7-1. 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.

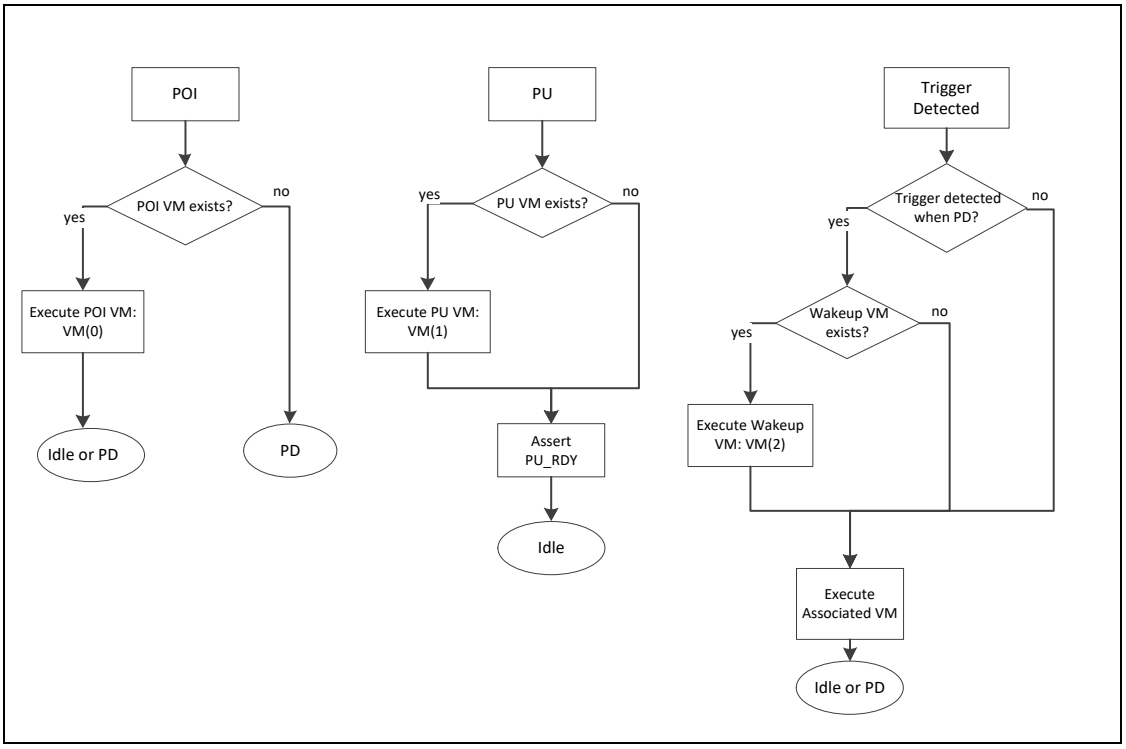


Figure 6-1 System Reserved Voice Macro Flow Chart

6.8 GPIO Trigger

6.8.1 GPIO Trigger Basics

Once power on, the ISD2361 continually detects the GPIO signal level transition. If an edge is detected, the ISD2361 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 ISD2361 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 ISD2361 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-VPE2361 GUI environment.

After creating and programming the application image into the device, , the ISD2361 device is ready for use in a triggering application.

6.8.2 SPI or GPIO Trigger

There are two ways to operate the ISD2361: 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.

6.8.2.1 Pulling SSB low automatically claims the SPI interface

Each of the six ISD2361 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. This mode switching caused by SSB level change is done by hardware automatically and is transparent to the user.

6.8.2.2 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.

6.8.2.3 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 ISD2361 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.

6.8.3 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 ISD2361 device no longer executes its associated Voice Macro.

6.9 Multi-Channel Feature

The ISD2361 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 ISD2361 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.

6.9.1 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.

6.9.2 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 ISD2361 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.

6.10 VM Jump and Channel Counter Commands

The ISD2361 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.

6.10.1 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-VPE2361 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.

6.10.2 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 ISD2361 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 ISD2361 integrates 2Mbit internal Flash memory, which contains 512 sectors with sector size of 512-byte each. Also, the ISD2361 series supports external NOR serial flash up to 2Gbit.

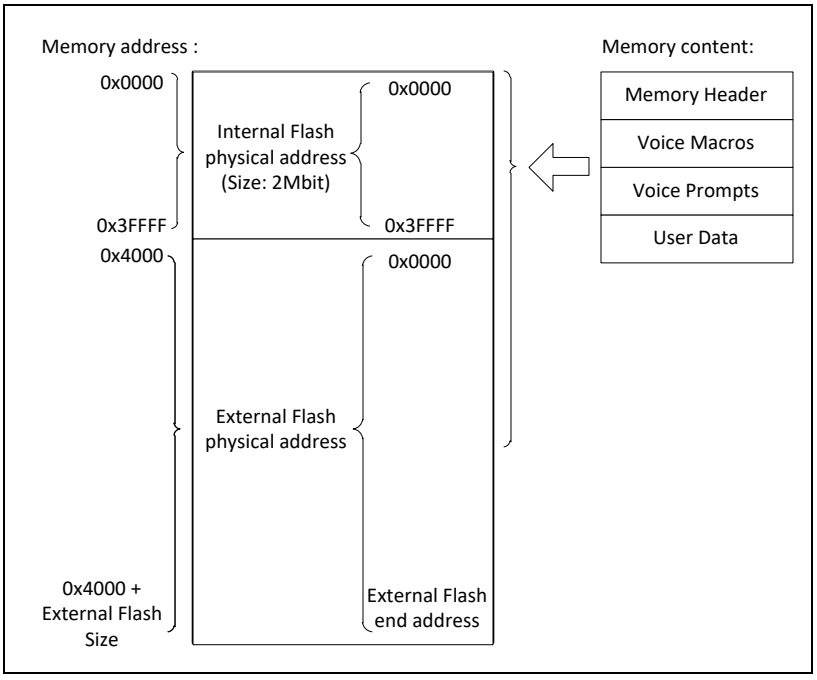


Figure 7-1 ISD2361 Internal Memory Organization

Figure 7-1 shows the ISD2361 memory organization. The internal and external flash together construct the system memory addressing space. When external flash is populated, starting from internal flash address 0x0000, the ISD2361 continuously addresses the memory into external flash once the address is greater than 0x3FFFF.

The ISD2361 needs to be pre-programmed before put into use. A GUI software, ISD-VPE2361 can be used to create the application image. If the application image is within 2Mbit, it will be programmed into ISD2361 internal flash only. If the image size is greater than 2Mbit, the ISD_VPE2361 software will generate two separate image files, one for internal and one for external flash. The address scope for internal image is 0x00-0x3FFFF. The address scope for external flash memory starts from address 0x40000.

An ISD2361 image consists of 4 parts:

- **Memory Header**: mainly consists of index table for Voice Macros and Voice Prompts, and also contains project configuration info.
- **Voice Macros**: data section for command scripts.
- **Voice Prompts**: audio data for all the sound effects.
- **User Data**: optional user reserved section for application data.

Note that user data is optional. An application image will always have memory header, VM data and VP data. At least the three system VMs will be included – if not generated by user, the VPE software will generate these three VMs by default.

7.1 Memory Header

7.1.1 Memory Format

The memory header is essentially important for device to function properly. Starting from address 0x000000, the memory header is programmed continuously from the first sector. Table 7.1.1-1 shows the memory header data organization.

Table 7.1-1 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	Protect Memory Pointer. Byte 0x04 is a reserved byte for ISD2361.
0x05 – 0x0C	POI_VM index	Start and end addresses of POI VM, 8 bytes in total.
0x0D – 0x14	PU_VM index	Start and end addresses of PU VM, 8 bytes in total.
0x15 – 0x1C	Wakeup_VM index	Start and end addresses of Wake-up VM, 8 bytes in total.
0x1D – A1 = 0x1D+8*VMs	Custom VM index	From Address 0x1D to A1, holding the start and end addresses for all other custom VMs. In total 8*(number of custom VMs) bytes.
A1 – A2 = A1 + 8*VPs	VP index	From address A1 to A2, holding the start and end addresses for all the VPs. In total 8*(number of VPs) bytes.
A2 – A3 = 15	VPE Info	From Address A2 to A3, are 15 bytes holding VPE project info
A3 - A4	VM data	Holds all the VM data continuously.
A4 – A5	VP data	Holds all the
A5 – end of the memory	empty	Blank space all 0xFF

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

The Memory Header is organized as:

- Byte 0 determines the memory protection scheme. Typically user should avoid accidentally writing 0's to its three lower bits. Because writing 00 to this byte bit[2:0] will cause the device being locked: the device can no longer be read, chip erase and sector erase.
- Bytes 1-2: reserved;
- Byte 3 store the PMP pointer for ISD2361; Byte 4 is reserved.
- Byte 5 onwards is the Voice Macro index table: each of the 8 byte are the VM's 4-byte starting address and 4-byte end address;
- From address A1 – A2 are VP index, which holds all Voice Prompts' starting address and end address, each of the starting address or end address is 4byte long;
- After VP index table there are 15 bytes reserved for VPE info;
- From address A3-A4 are all Voice Macros' script data;

- From address A4- A5 are all Voice Prompts' audio data;
- From address A5 to the end are non-used memory, all 0xFF.

7.1.2 Memory Protection

The ISD2361 provides three types of memory protection functions; they are Chip Erase Protect, Write Protect (or Sector Erase Protect) and Read Protect. For Write Protect and Read Protect, the memory protection range is determined by PMP -- Memory Protect Byte. PMP has no effect on Chip Erase Protect. The ISD2361 memory protection function only applies on its internal flash; it does not protect the external flash.

7.1.2.1 Memory Protection Range

For ISD2361 PMP value is the byte value stored in internal flash address 0x03. The memory protection range can be calculated by formula below:

Protection scope = PMP * 0x400 + 0x400, in byte address.

For example, if the byte value in 0x03 is

- 0x00, then the protection range is 0x00 – 0x3FF;
- 0x10, then the protection range is 0x00 – 0x43FF;
- 0xFF, then the protection range is 0x00 – 0x3FFFF, which covers the whole internal flash 2 Mbit.

7.1.2.2 Memory Protection Range

The three types of memory protection are defined in *MP_CFG* byte as shown in Table 7.1.2.2-1**.** MP_CFG byte is the first byte located in internal flash byte address 0x00.

Memory protection is activated via power-up from system power on, or Reset. After powered up whenever the user changes the settings for memory protection in byte 0x00, the new setting will only take effect after the chip is reset.

Table 7.1-2 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 disables CHIP_ERASE command. Note that chip-erase protection has nothing to do with sector erase function (or Write Protect).

Write Protect Writing zero to the **WP** bit enables write protection of the internal memory within the range pointed by the PMP. Write protection disables the digital write or erase commands such as DIG_WRITE and MEM_ERASE within PMP pointed memory area. Note that **WP** protection does not block the Chip Erase command.

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 disables digital read command for the protected memory range. Once activated, the protected memory cannot be read.

ISD2361 memory protection scheme is shown in **Table 7.1-3**.

Table 7.1-3 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.

7.1.2.3 Errata sequence for protection scheme 0xC9

Please note when protection scheme is 0xc9, and when user wants to Chip Erase device, user should follow the following errata sequence to execute chip erase function:

- Issue Chip Erase command;
- Then Wait INT – so to make sure Chip Erase function finishes;
- Then Read byte 0x03 FF F0;
- Then issue Chip Erase command again;
- Then Wait INT – again make sure Chip Erase function finishes, i.e. wait until device is no longer busy.

This sequence can guarantee Chip Erase successful and it is only needed when protection scheme is 0xC9.

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. For ISD2361, the starting address and end address for a VP are all 4-byte long.
- 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 ISD2361 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 ISD2361. Once it gets running, the command data in the script will be read from memory and executed by the device. For the ISD2361, 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 ISD2361 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 ISD2361 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-VPE2361 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 7.3.1-1 provides the Voice Macro commands and the byte index.

Table 7.3-1 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[31:24]	Addr[23:16]	Addr[15:8]	Addr[7:0]	0x00*
GOTO	0xE1	Addr[31:24]	Addr[23:16]	Addr[15:8]	Addr[7:0]	0x00*
WAIT_CHN_CNT	0xEE	0x00*	0x00*			
FINISH	0xFF	-	-			

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

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; addr is a 32bit bit address.

GOTO(n) Absolute jump to memory address addr, addr is a 32bit address.

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

FINISH Finish the Voice Macro and exit.

The user can use GUI software ISD_VPE2361 to create, edit Voice Macros. For details operation, user can check demo kits start page, or search Youtube ISD ChipCorder videos for operation tutorial.

7.3.2 System-Reserved Voice Macros

The ISD2361 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. ISD2361 VPE automatically reserves space for these system-reserved VMs when creating a project. Custom VM is manually created and starts from index 0x03.

All VMs can be executed by sending the SPI EXE_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 6.8 System Voice Macro Flow Chart.

Note: for ISD2361 the three system reserved VMs should not be left empty. If the user does need to any configuration in these VMs, the user can imply insert PD in POI VM, insert FINISH in PU VM, and insert FINISH in Wakeup VM. VPE software will safeguard this to ensure these three VMS are not empty when generating system image.

7.3.3 Sample Project Voice Macros

ISD_VPE2361 software installation comes with 3 example projects, they reside in “C:\Program Files\ISD-VPE2361” after installation. The user can use these example projects to play with device, playback audio to verify demo board, and study VM script editing basics.

This section uses these sample projects to illustrate how the Voice Macros works: play voice prompts in a sequence, setup GPIO triggers so a button press can trigger device to play certain audio sound, achieve multiple-channel mixing for playback, etc.

7.3.3.1 Sample_1: A Simple playback Project

Example_Project_1 is a simple playback only project.

Function: demo simple VP and VM playback function.

Figure 7-2 shows the screenshot taken from VPE Voice Macro tab, it show that the project has

- in total 7 Voice Prompts,
- 3 system reserved VMs. They do not configure anything, and all simply return.
- 2 task VMs.
  - VM3 plays VP sound effect “One” to “Five”.

- VP4 plays beep sound, “Alarm is ON”, and end with two beep sound.

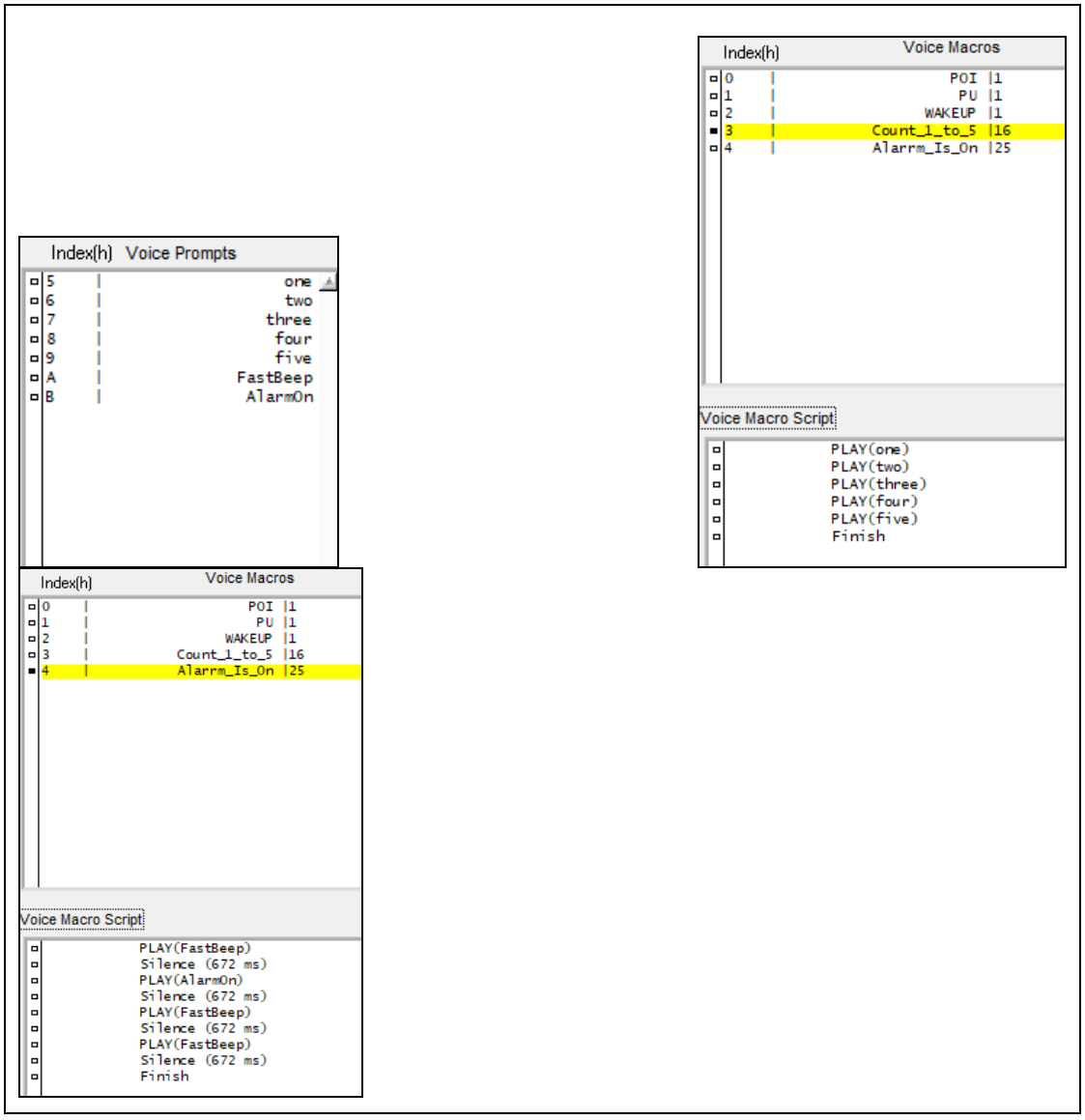


Figure 7-2 Example_Project_1 Configuration

7.3.3.2 Sample_2: One pin trigger to loop sound effect one to Six

Example_Project_2 is a one pin trigger playback only project.

Function: after power on or reset, press GPIO3/INTB button will cycle through sound effect “One” to “Six”, and then start over. That is: press once, play “ONE”; press again, play “Two”; and so on, until reach “Six” then press again it will play “One” again.

How to configure:

- In POI VM configure GPIO3/INTB pin as falling-edge trigger pin. GPIO3 associated VM is VM3, i.e. assign R3 = 03;
- In Wakeup VM, enable decode path and enable DPWM so to enable playback path.

- VM3 – VM8: configure a playback loop from “one” to “Six”.
  - VM3 plays back sound “one”, then assign R3= 04;
  - VM4 plays back sound “Two”, then assign R3= 05;
  - VM5 plays back sound “Three”, then assign R3= 06;
  - VM6 plays back sound “Four”, then assign R3= 07;
  - VM7 plays back sound “Five”, then assign R3= 08;
  - VM8 plays back sound “Six”, then assign R3= 03;

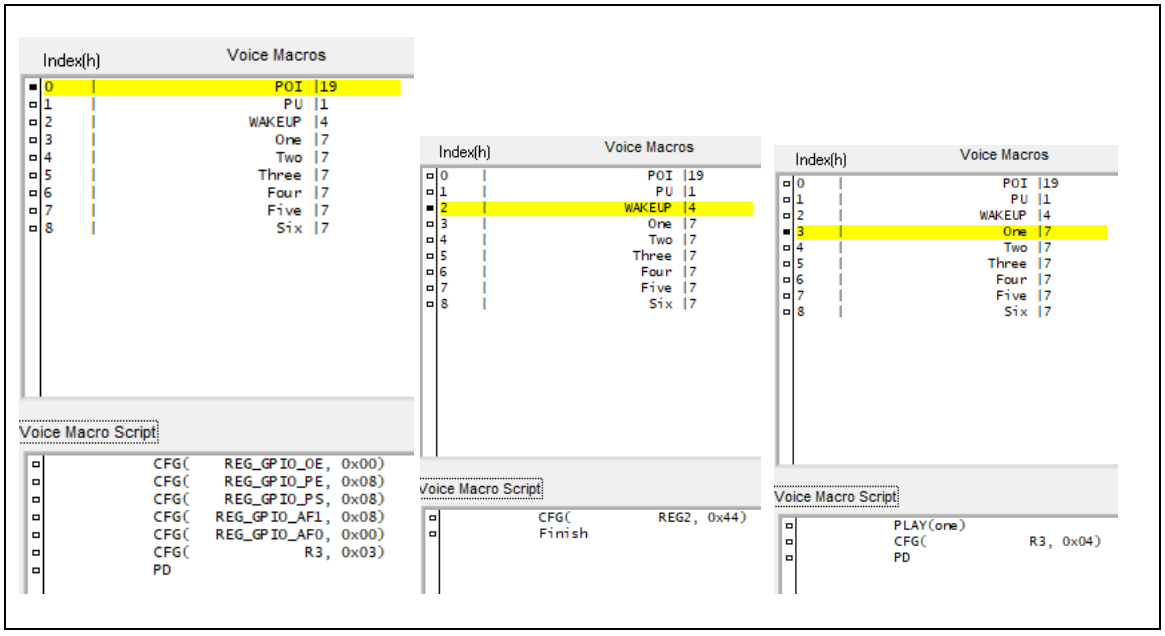


Figure 7-3 Example_Project_2 Configuration

The following exlpains the POI VM:

CFG(REG_GPIO_OE, 0x00) → write 0x00 into reg0x1A; configure all GPIO pins as input pins;
CFG(REG_GPIO_PE, 0x08) → write 0x08 into reg0x1B; enable Pull-Enable for GPIO3 only;
CFG(REG_GPIO_PS, 0x08) → write 0x08 into reg0x1D; enable Pull-High for GPIO3;
CFG(REG_GPIO_AF1, 0x08) → write 0x08 into reg0x1E;
CFG(REG_GPIO_AF0, 0x00) → write 0x00 into reg0x1F; configure GPIO3 as falling edge trigger pin.
CFG(R3, 0x03) → write 0x03 into reg0x26; associate VM0x04 with GPIO3 trigger;
PD → power down device (to save power, and wait for trigger).

The Wakeup VM 0x02 enables the playback path after trigger detected and system wakeup from PD state.

CFG(REG2, 0x44) → write 0x44 into reg0x02; enable Decoder and PWM output path;

Task VM VM03:

PLAY(one) → Play VP “one”
CFG(R3, 0x04) → now chanege associated VM for GPIO3 to VM 0x04;
PD → power down device.

7.3.3.3 Sample_3: Trigger for Channel mixing and volume control

Example_Project_3 function:

- Press GPIO0 to continuously play “Do ---“ in channel 0; press again will stop playback

in channel 0. This cycle repeats.

- Press GPIO1 to continuously play “Re ---“ in channel 1; press again will stop playback in channel 1. This cycle repeats.
- Press GPIO2 to continuously play “Mi ---“ in channel 2; press again will stop playback in channel 2. This cycle repeats.
- Press GPIO3 to stop the current channel playback, then assign GPIO3 its own trigger playback channel to next channel. So keeping pressing this button eventually stops playback in all channels.
- Press GPIO4 to increase playback volume by 1 level, total 8 levels;
- Press GPIO5 to decrease playback volume by 1 level, total 8 levels.

How to configure: Configure the device as follows:

- In POI VM
  - Configure GPIO 0-5 all as falling edge trigger pins;
  - Assign channel 0-2 to GPIO0-2 respectively;
  - Assign GPIO4 as volume up trigger pin;
  - Assign GPIO5 as volume down trigger pin;
  - Assign channel 0 to GPIO3.

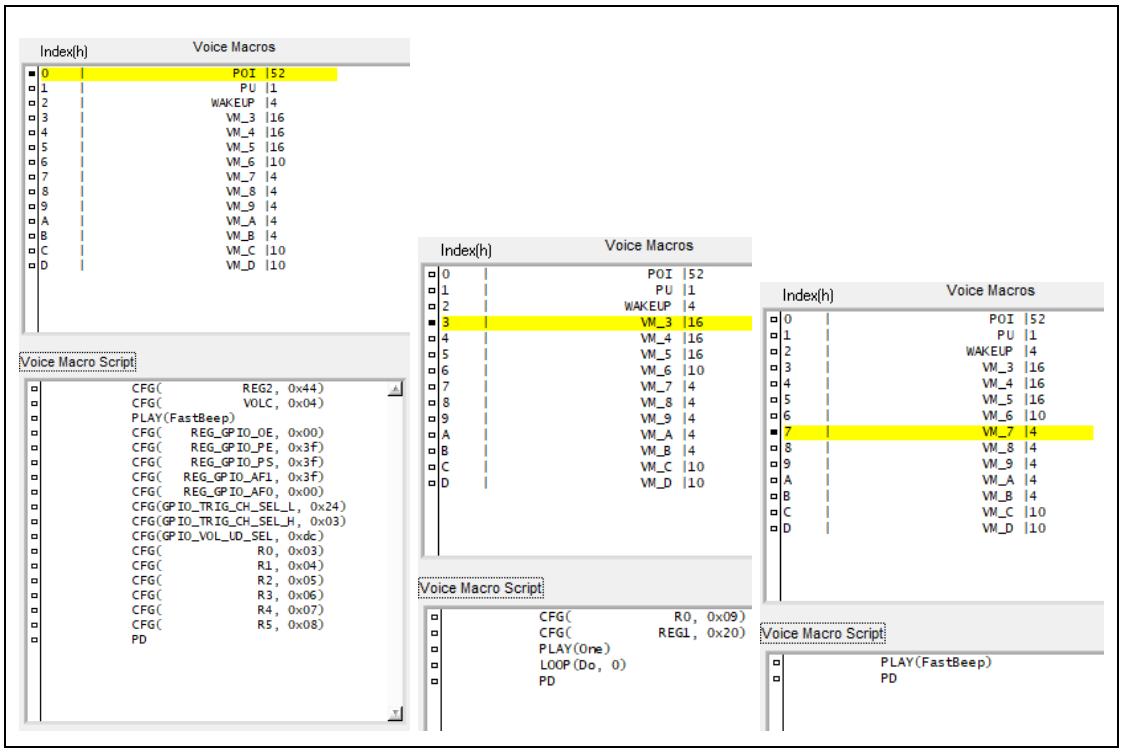


Figure 7-4 Example_Project_3 Configuration

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 GPIO0;
                                    assign channel 1 to GPIO1;
                                    assign channel 2 to GPIO2;
                                    assign channel 0 to GPIO3;
CFG(GPIO_TRIG_CH_SEL_H, 0x03)→ write 0x03 into reg0x15, assign all channels to VM associated with GPIO4;
                                    assign channel 0 to GPIO5;
CFG(GPIO_VOL_UD_SEL,      0xdc) → write 0xdc into reg0x16, enable both volume up/down via trigger;
                                    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).
```

Wakeup VM: enables playback path.

```
CFG(                 REG2, 0x44) → write 0x44 into reg0x02; enable Decoder and PWM output path;
FINISH                           → End Wakeup VM. Device stays at PU state.
```

Task VM_3: Continuously play “Do ---’ in channel 0, and assign R0=0x09

```
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.
```

Similarto VM_3, task VM_4 loop play “Re ---” in channel 1 and VM_5 loop play “Mi ---” in channel 2. Multiple trigger on GPIO0, 1 and 2 can make device loop play “Do ---” “Re---” “Mi ---” in different chanel simultanuously, thus achieved channel mixing playback.

Task VM_9: stop channel 0 playback, and assign R0=0x03

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

VM_3 and VM_9 combine to make GPIO0 trigger to play, and to stop repeatedly. Similarily, VM_4 and VM_B, VM_5 and VM_B combine to achieve the same function on GPIO1 and GPIO2, in channel 1 and 2 respectively.

Task VM_6: stops the playback in current channel, then assign GPIO3 playback channel to channel 1.

```
Silence (32ms)                   → play silence for 32 ms.
CFG(GPIO_TRIG_CH_SEL_L, 0x64) → write 0x64 into reg0x14, so assign GPIO3 trigger channel as channel 1
CFG(                    R3, 0x0C) → assign R3=0x0c, so next GPIO3 trigger executes VM_C
PD                               → power down device (to save power).
```

Task VM_C: stops the playback in current channel, then assign GPIO3 playback channel to channel 2.

```
Silence (32ms)                   → play silence for 32 ms.
CFG(GPIO_TRIG_CH_SEL_L, 0xA4) → write 0x64 into reg0x14, so assign GPIO3 trigger channel as channel 2
CFG(                    R3, 0x0d) → assign R3=0x0c, so next GPIO3 trigger executes VM_D
PD                               → power down device (to save power).
```

Task VM_D: stops the playback in current channel, then assign GPIO3 playback channel to channel 0.

Silence (32ms) → play silence for 32 ms.
CFG(GPIO_TRIG_CH_SEL_L, 0x24) → write 0x64 into reg0x14, so assign GPIO3 trigger channel as channel 0
CFG(R3, 0x06) → assign R3=0x0c, so next GPIO3 trigger executes VM_6
PD → power down device (to save power).

Task VM_7: GPIO4 is configured as volume-up trigger pin. Each trigger automatically increases volume, utill reaches 0 attenuation, i.e. max volume.

PLAY(FastBeep) → Play VP “Fastbeep” with increased volume.
PD → power down device (to save power).

Task VM_8: GPIO5 is configured as volume-down trigger pin. Each trigger automatically decreases volume, utill volume off.

PLAY(FastBeep) → Play VP “Fastbeep” once with decreased volume.
PD → power down device (to save power).

7.3 User Data

The *ISD-VPE2361* 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-VPE2361* 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 ISD2361 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 ISD2361 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 ISD2361 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 8.1-1 SPI Data Transaction Waveform.

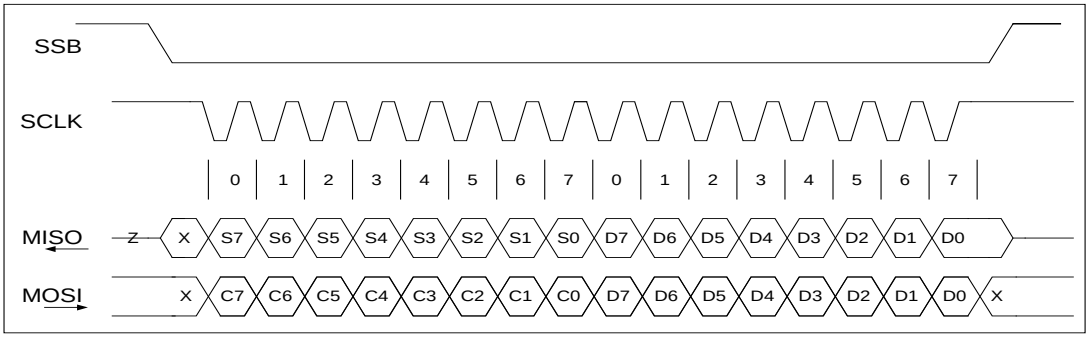


Figure 8-1 SPI Data Transaction Waveform

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 ISD2361 is ready for the next data transmission. See Figure 8-2 for the timing diagram.

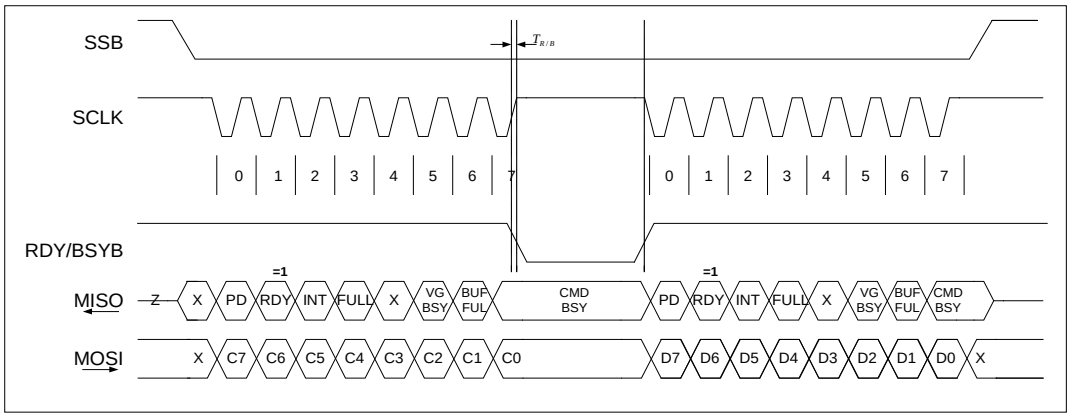


Figure 8-2 RDY/BYSB Timing for SPI Write Transactions

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 ISD2361 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 ISD2361; or (2) invalid data from ISD2361. 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.

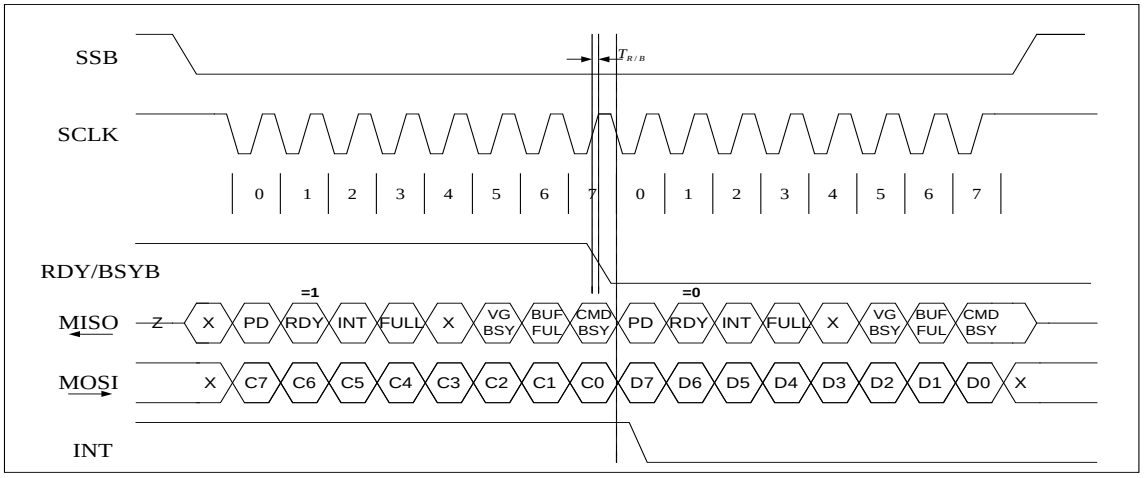


Figure 8-3 RDY/BSYB Ignored for SPI Transactions

8.2 SPI Commands

The ISD2361 provides SPI commands to play audio, query device status, perform digital memory operations and configure the device. Figure 8-4 provides a list of all SPI commands and their function descriptions.

Figure 8-4 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 ISD2361 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 ISD2361 via SPI interface for decoding.
READ_STATUS	0x40	XX	XX	XX	…	Query status of ISD2361.
READ_INT	0x46	XX	XX	XX	…	Query status and clear interrupt flags of ISD2361.
READ_ID	0x48	XX	XX	XX	XX	Read device ID of ISD2361.
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.
OP_EXT	0x38	FLASH BYTE1	FLASH BYTE2	FLASH BYTE3	…	Pass Down command to external flash
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 the device.

Instructions	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 … Byte n	Description
PWR_DN	0x12					Power down ISD2361
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 Figure 8.3-1, 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.

Figure 8-5 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
EXE_VM	0xB0	0	1	x	-	√	√	√	0
EXE_VM@Rn	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
ERASE_MEM	0x24	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
CHIP_ERASE	0x26	0	1	x	-	0	0	0	0
OP_EXT	0x38	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
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]
	ISD2361	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]	
	ISD2361	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]
	ISD2361	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]
	ISD2361	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]

	ISD2361	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]	
	ISD2361	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]	
	ISD2361	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		
	ISD2361	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 ISD2361. 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		
	ISD2361	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					
	ISD2361	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.

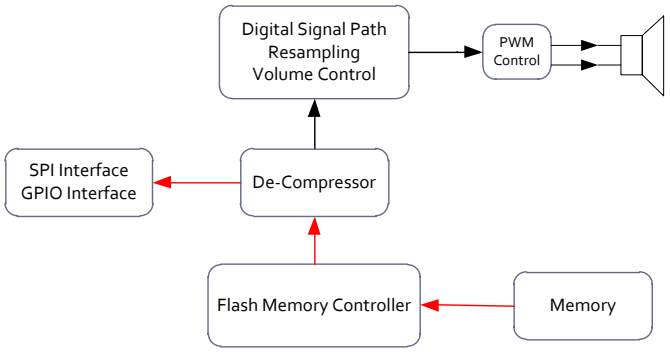


Figure 8-6 SPI Read De-Compressed Data (Playback)

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]
	ISD2361	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.

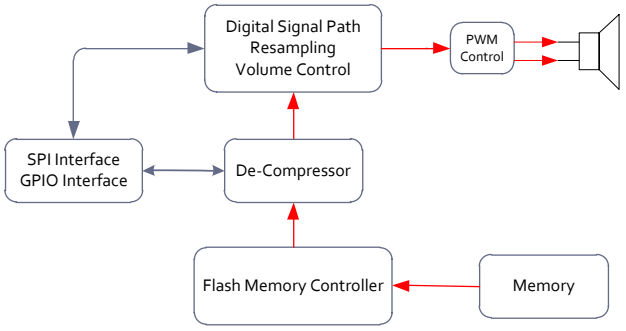


Figure 8-7 SPI Send Compressed Data to Decode

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
	ISD2361	Status Byte	Interrupt Status Byte
Description:	Query device status.		

This command queries the ISD2361 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	
	ISD2361	Status Byte	Interrupt Status Byte	
Description:	Query device status and clear interrupt flags.			

This command queries the ISD2361 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
	ISD2361	Status Byte	PART_ID	MAN_ID	MEM_TYPE	DEV_ID
Description:	Return ID of ISD2361					

This command queries the ISD2361 and returns four bytes to identify the ISD2361 part, the manufacturer, and the size and type of internal memory of the device. The bytes returned are:

- One byte ISD2361 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 ISD2361. 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
	ISD2361	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
	ISD2361	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 ISD2361 will finish the Flash write operation. When this operation is complete, the ISD2361 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]
	ISD2361	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
	ISD2361	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 OP_EXT – Pass Down to external Flash

OP_EXT					
Byte Sequence:	Host Controller	0x38	Flash_Code1	Flash_Code2	….
	ISD2361	Status	Flash Status	Flash Status	Flash Status
Description:	Pass Down to external flash				
Interrupt Generation:	CMD_ERR if the device is busy. From second byte clock MISO returns external flash status.				

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.

ISD2361 supports external flash SPI line feed through, which means if need to do digital read from or digital write into external flash, the SPI line interfacing host controller will connect to the external flash SPI interface directly. Thus allows user access to full flash SPI command set.

The command format is 0x38 followed by a valid SPI flash command for that flash. For example when using Winbond flash as the external flash:

- To read flash Status Register-1, send command sequence 0x38, 0x05. 0x05 is the SPI command for Winbond flash Read Status Register-1.
- To do sector erase for a Winbond flash at 3-Byte address mode, send command 0x38 0x20, A23-A16, A15-A8, A7-A0. Command 0x20, A23-A16, A15-A8, A7-A0 is the SPI command for Winbond flash Sector Erase (4KB).

Return status: MISO data corresponding to opcode 0x38 is the ISD2361 status; MISO data corresponding to SPI command is the external flash status.

8.4.3.6 CHECKSUM – Calculate Hardware Checksum

CHECKSUM					
Byte Sequence:	Host Controller	0xF2	EA[23:16]	EA[15:8]	EA[7:0]
	ISD2361	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 ISD2361. 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		
	ISD2361	Status		…
Description:	Powers up the device and initiates the power up sequence.			

This command powers up the ISD2361. 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		
	ISD2361	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 WR_CFG_REG – Write Configuration Register

WR_CFG_REG						
Byte Sequence:	Host Controller	0xB8	REG[7:0]	D0	…	Dn
	ISD2361	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.

8.4.4.4 RD_CFG_REG – Read Configuration Register

RD_CFG_REG						
Byte Sequence:	Host Controller	0xBA	REG[7:0]	X	…	X
	ISD2361	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.

8.4.4.5 RESET – Reset Device

RESET						
Byte Sequence:	Host Controller	0x14	X	…	X	
	ISD2361	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

Figure 9-1 Register Operations

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
-	-	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	TALARM_INT									Set when temperature exceeds temperature threshold set in CFG5[3:0].
			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.
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-compres	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.

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
		sion Control Register										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.
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

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Write only.
5	05	Thermal Control Register	-									Reserved
			TALARM_INTEN									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	-	EXF_RD_CMD									EXF_RD_CMD[7:0] sets the custom external page read command. When playing Voice Prompt or Voice Macro, and the content is stored in the external memory, the chip will send SPI flash Read command to access the external memory. The default value is 8'h03, which is the Read command for Winbond NOR flash. If the customers use different external memory which have different read command. This register need to be overwritten with the read command opcode for that flash.
			Default >>	0	0	0	0	0	0	1	1	0x03 reset value. Read/Write.
7	07	-	FOUR_BYTE_ADDR									To correctly address external flash, host MCU or a Voice Macro must configure this address mode bit and external flash at the same time. To access flash memory beyond 128Mbyte, FOUR_BYTE_ADDR bit need to be set to 1, meantime the external flash 4-byte mode need to be enabled. By default, this bit is set to 0, so the I2361 uses 3-byte mode and can address within 128Mbyte range.
												Reserved
			Default >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
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	1	0	0	0	0x08 reset value. Read/Write.

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
10	0A		Reserved									Reserved
			GPIO_PE[9]									0 = no pull, 1 = pull high
			GPIO_PE[8]									0 = no pull, 1 = pull high
			GPIO_OE[9]									0 = SCLKx pin; 1 = tristate pin
			GPIO_OE[8]									0 = MOSIx pin; 1 = tristate pin
			Default >>		0	0	0	1	1	0	0	0x0C reset value. Read/Write.
11	0B											
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. 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

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
			GPIO1_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
			GPIO0_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.
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	GPIO_DOUT[7]									0 = GPIO7 output 0; 1 = GPIO7 output 1
			Reserved									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

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
			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.
			GPIO_OE[7]									0 = tristate (input mode); 1 = enable GPIO7 output
			GPIO_OE[6]									0 = SSBx pin; 1 = tristate pin (input mode)
			GPIO_OE[5]									0 = disable GPIO5 output; 1 = enable GPIO5 output
			GPIO_OE[4]									0 = disable GPIO4 output; 1 = enable GPIO4 output
26	1A	Output Enable Control Register	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.
			GPIO_PE[7]									0 = no pull; 1 = GPIO7 pull high
			GPIO_PE[6]									0 = no pull; 1 = GPIO6 pull high
			GPIO_PE[5]									0 = disable GPIO5 pull ; 1 = enable GPIO5 pull
			GPIO_PE[4]									0 = disable GPIO4 pull; 1 = enable GPIO4 pull
27	1B	Pull Enable Control Register	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.
			GPIO_DIN[7]									0 = GPIO7 input data 0 ; 1 = GPIO7 input data 1
			GPIO_DIN[6]									0 = GPIO6 input data 0 ; 1 = GPIO6 input data 1
			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
28	1C	Input Data Control Register	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 >>	1	1	1	1	1	1	1	0	0xFE reset value. Read/Write.
			-									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
29	1D	Pull Select Control Register	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 >>	0	0	0	0	0	0	0	0	0x00 reset value. Read/Write.
			Reserved									Reserved
			AF1[5]									Combined with register 0x1F to define GPIO[5:0] pin functions. See Table 5.2-1 for GPIO function definition.
			AF1[4]									
30	1E	Alternate Function Control 1 Register	AF1[3]									
			AF1[2]									
			AF1[1]									
			AF1[0]									

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	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[5:0] pin functions. See Table 5.2-1 for GPIO function definition.
			AF0[4]									
			AF0[3]									
			AF0[2]									
			AF0[1]									
			AF0[0]									
			Default >>	0	0	0	0	1	1	1	1	0x0F 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	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.
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

Register		Function	Name	Bit								Description
Dec	Hex			7	6	5	4	3	2	1	0	
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 ISD2361 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 Mode Application under MCU Control

The ISD2361 has two SPI interfaces. One SPI interface is for the communication with host microcontroller. The other SPI interface is for external flash control.

For the SPI interface with host microcontroller, a RDY/BSYB signal (pin) is available for digital read/write data flow control.

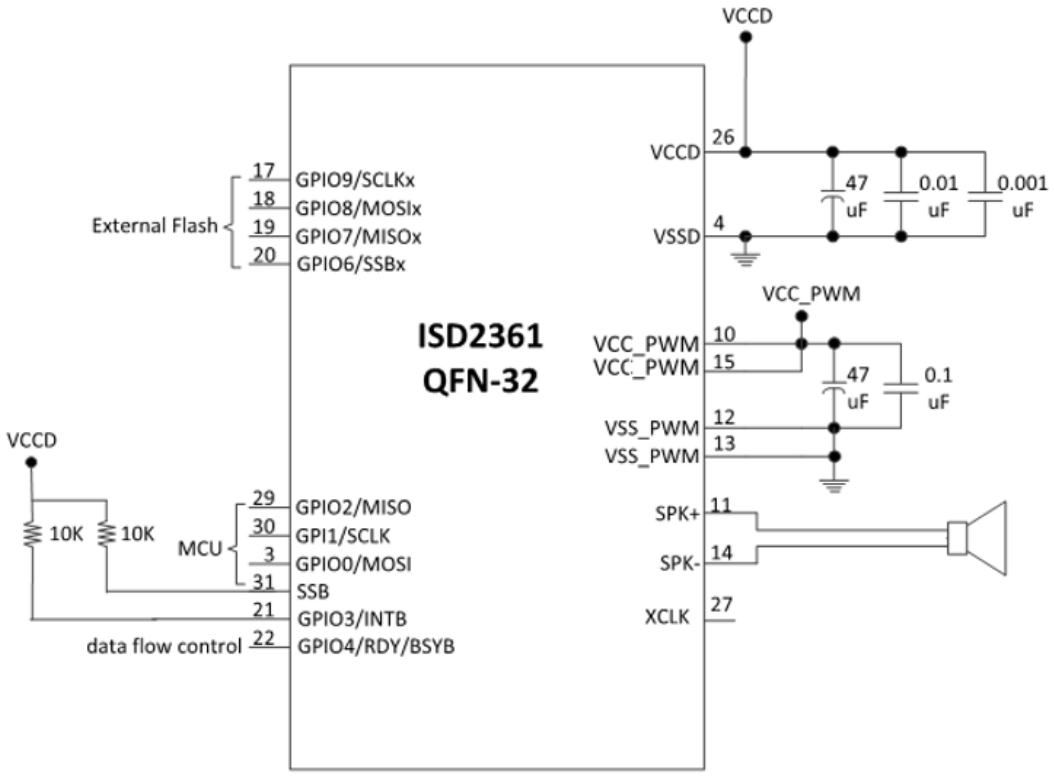


Figure 10-1 SPI Mode Application under MCU Control

10.2 GPIO Trigger Standalone Application

The ISD2361 can operate in standalone mode by triggers applied to the device's six General Purpose Input/output (GPIO) pins. Once Trigger mode is activated, a button press event will trigger the associated VM to run.

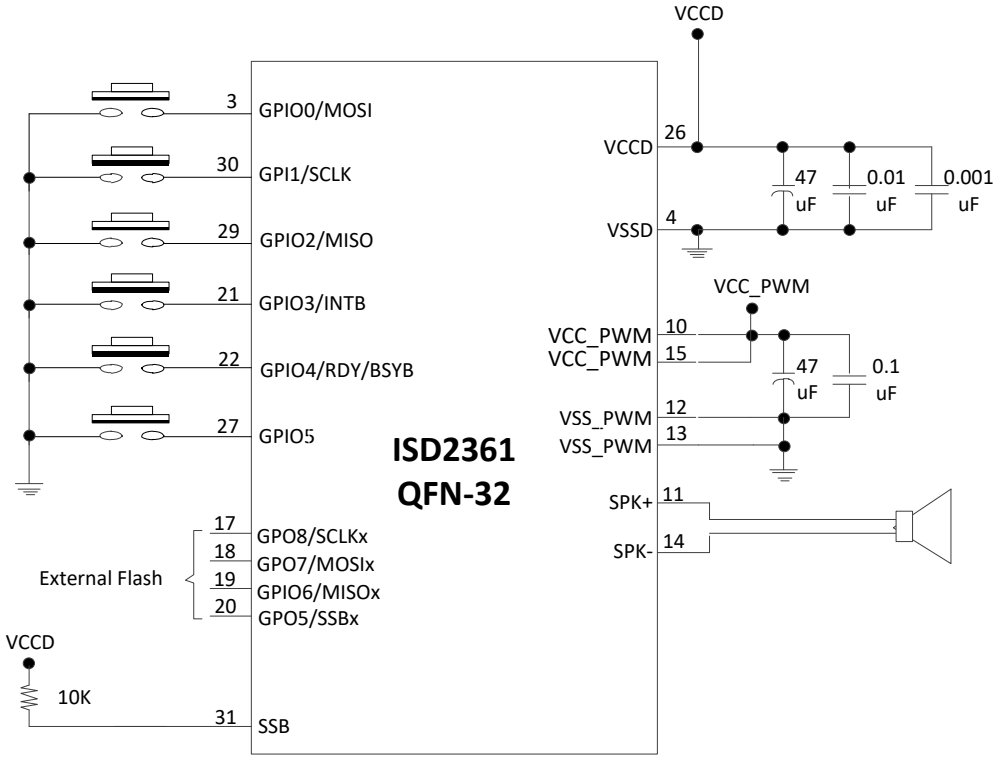


Figure 10-2 GPIO Trigger Standalone Application

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 11.2-1 Operating Conditions (Industrial Packaging)

Condition	Value
Operating temperature range (Case temperature)	-40°C to +105°C[1]
Supply voltage (V_{DD}) [2]	+2.4 V to +5.5 V
Ground voltage (V_{SS}) [3]	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] TBD: MAXIMUM OPERATING TEMPERATURE TO BE FURTHER CHARACTERIZED.

[2] $V_{DD} = V_{CCD} = V_{CCPWM}$

[3] $V_{SS} = V_{SSD} = V_{SSPWM}$

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	Max	Units	CONDITION
Output Power	P_{OUT_SPK}			440	mW	@ 3.3V, Load 4Ω [1][2]
				1.2	W	@ 5.0V, Load 4Ω [1][2]
				1.5	W	@ 5.5V, Load 4Ω [1][2]
				330	mW	@ 3.3V, Load 8Ω [1]
				800	mW	@ 5.0V, Load 8Ω [1]
				1.0	W	@ 5.5V, Load 8Ω [1]
THD, Memory to SPK+/SPK-	THD %					w/o load [1] [3]
Minimum Load Impedance	$R_{L(SPK)}$	4	8		Ω	

Notes:
[1] T_A=25°C , 0dB FS, 12-bit PCM, 8K SR.
[2] Reg0x0B is configured as 0x5C.
[3] All measurements are C-message weighted.

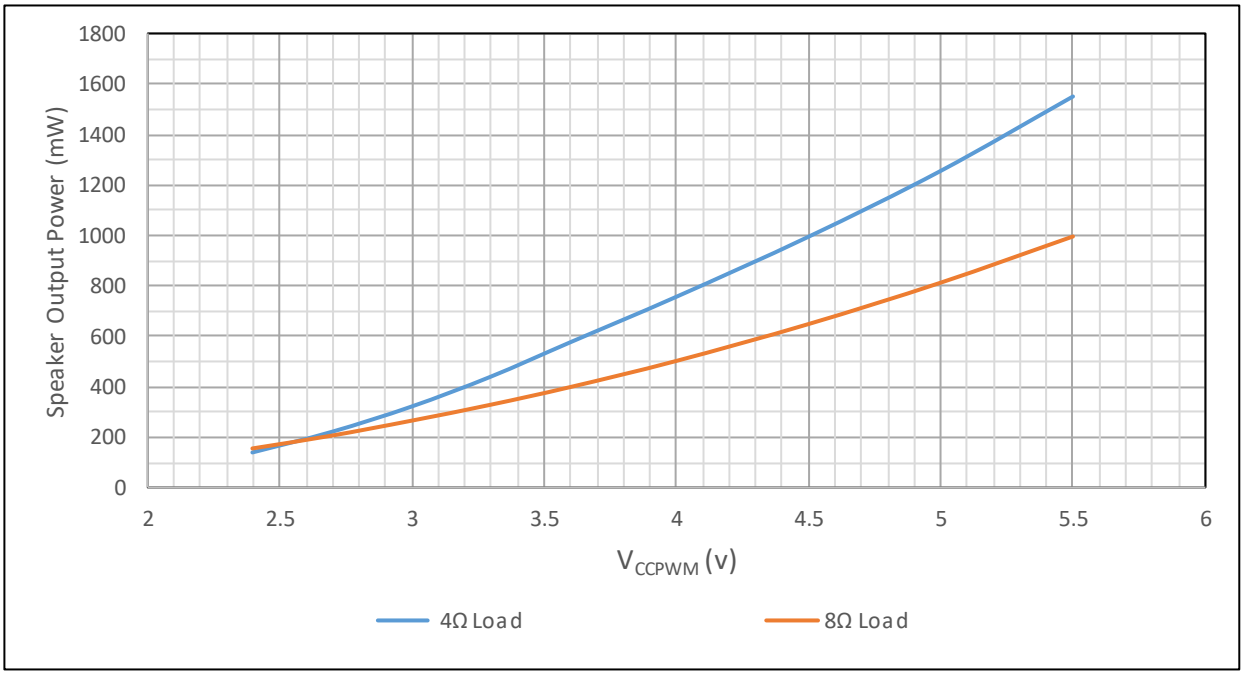


Figure 11-1 Speaker output power

11.3.3 DC Parameters

PARAMETER	SYMBOL	MIN	TYP [1]	MAX	UNITS	CONDITIONS
Supply Voltage	V_{DD}	2.4		5.5	V	
Input Low Voltage	V_{IL}	V_{SS}-0.3		0.3xV_{DD}	V	
Input High Voltage	V_{IH}	0.7xV_{DD}		V_{DD}	V	
Output Low Voltage	V_{OL}	V_{SS}-0.3		0.4	V	I_{OL} = 1mA
Output High Voltage	V_{OH}	2.4			V	I_{OH} = -1mA
Pull-up Resistance	R_{PU}		50		kΩ	
Pull-down Resistance	R_{PD}		10		kΩ	
INTB Output Low Voltage	V_{OH1}			0.4	V	
Maximum Operating Current	I_{DD_MAX}		8			V_{DD}= 5.5V, No load, Sampling freq. 8 kHz, all blocks enabled.
Playback Current	$I_{DD\_Playback}$		4		mA	No Load, V_{DD}=3V
Standby Current	I_{SB}		2.5	10	µA	V_{DD}= 5.5V
Input Leakage Current	I_{IL}			±10	µA	Force V_{DD}

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

11.3.4 SPI Timing

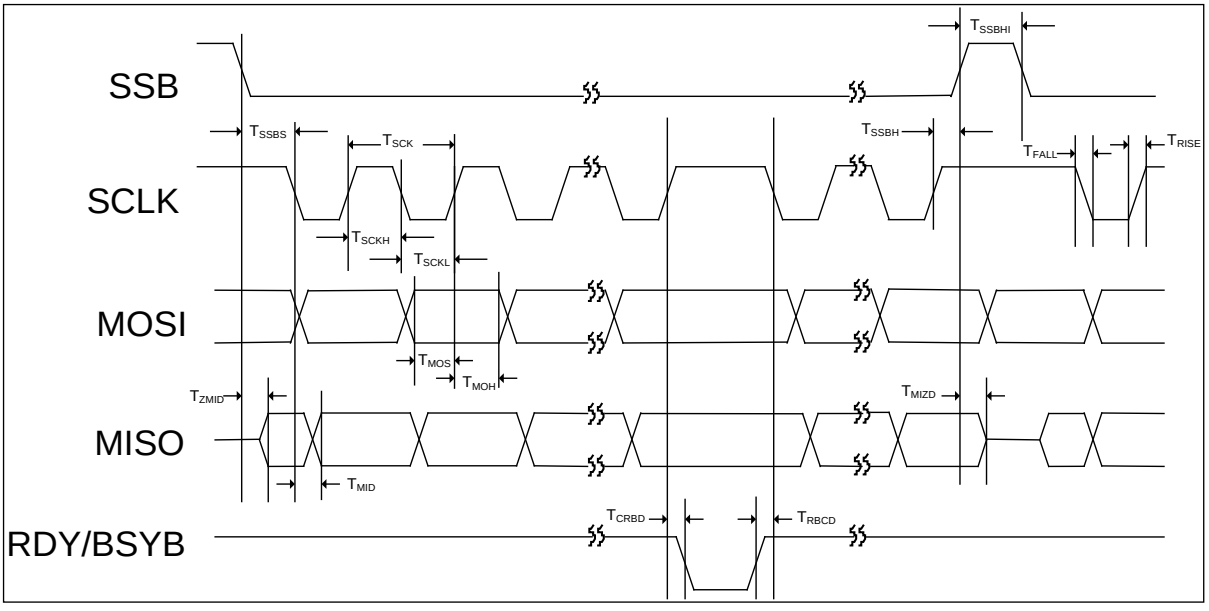
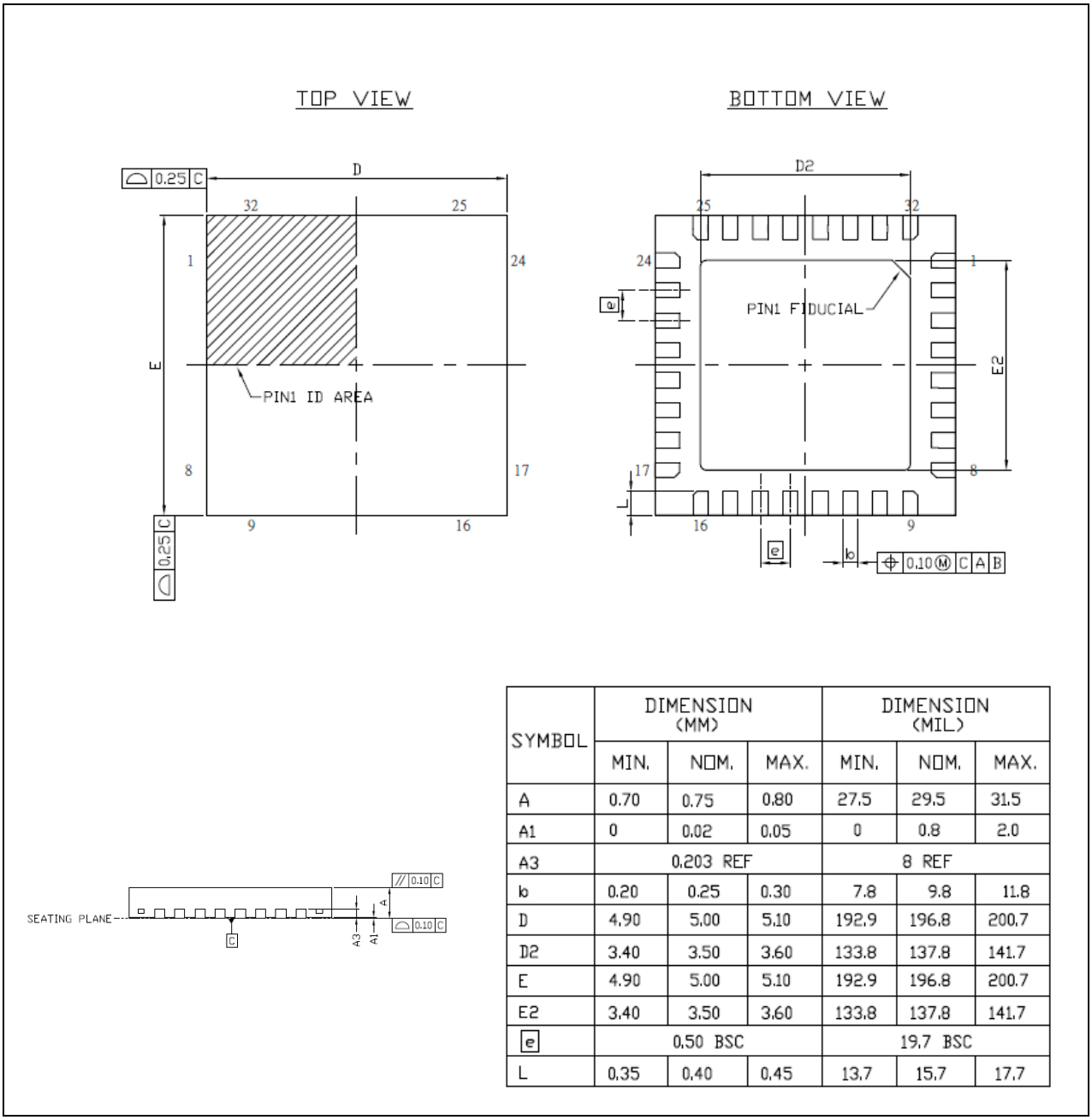


Figure 11-2 SPI Timing

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

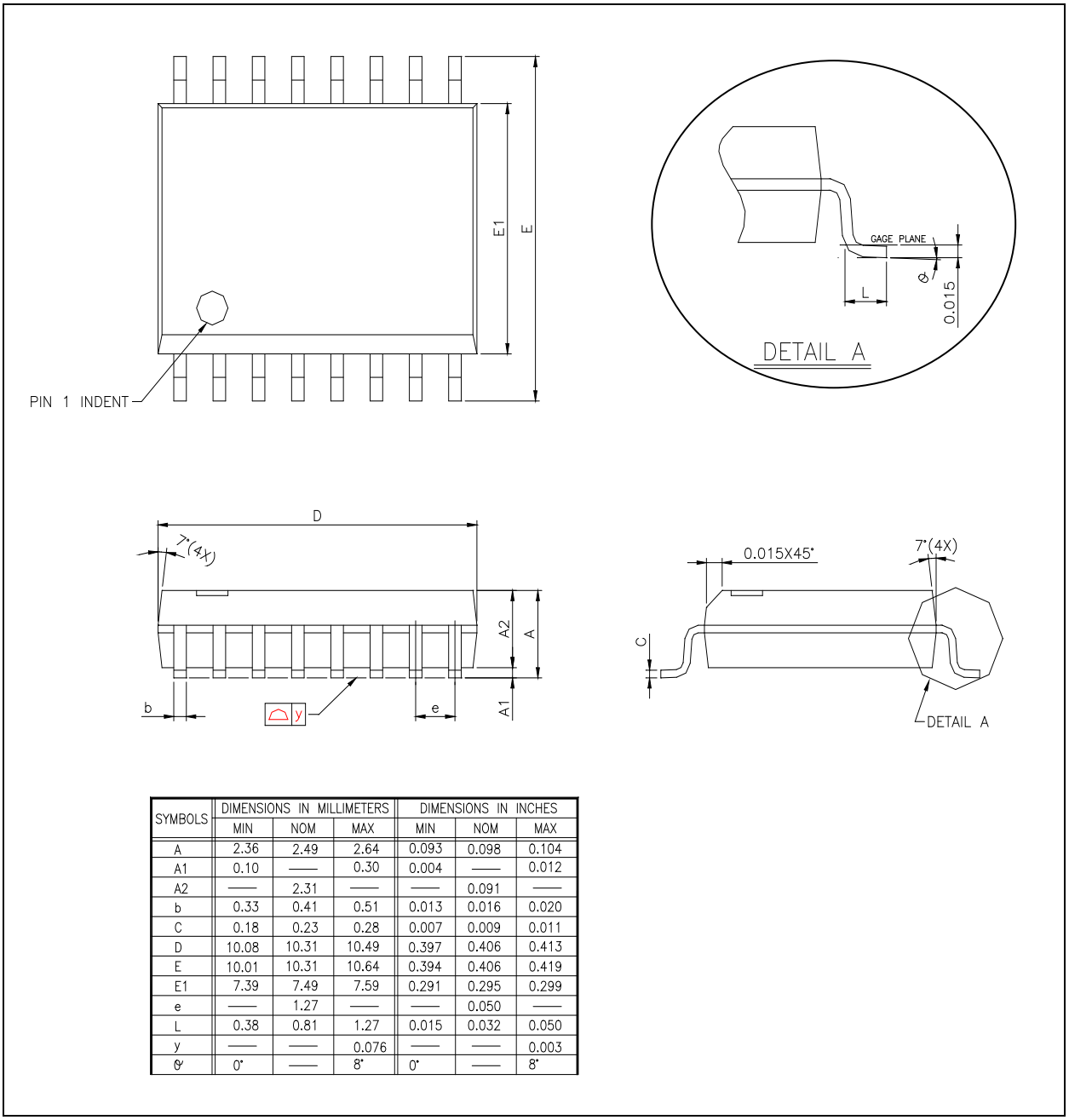
12. Package Dimensions

The ISD2361 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

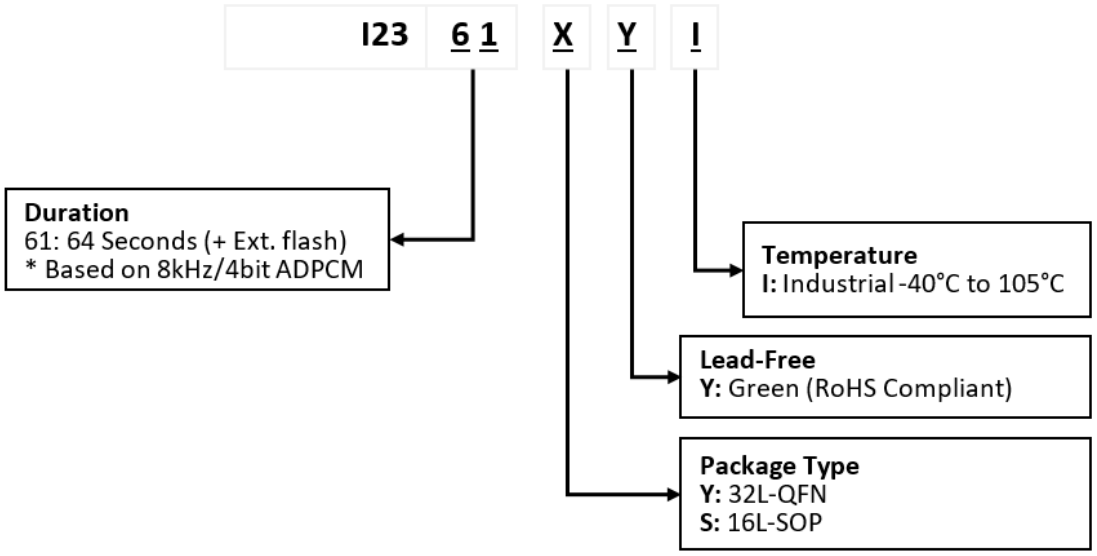
Figure 12-1 QFN 32-Lead Package



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



Package Number	Part Number	Ordering Number	Duration	Package	Temperature	Notes
ISD2361YYI	ISD2361YYI	I2361YYI	64s + ext. flash	32L-QFN	-40°C to 105°C	
ISD2361YYI TR	ISD2361YYI	I2361YYI	64s + ext. flash	32L-QFN, Tape & Reel	-40°C to 105°C	
ISD2361SYI	ISD2361SYI	I2361SYI	64s	16L-SOP	-40°C to 105°C	
ISD2361SYI TR	ISD2361SYI	I2361SYI	64s	16L-SOP, Tape & Reel	-40°C to 105°C	

14. Revision History

Version	Date	Description
1.0	Nov 11, 2021	First release
1.1	Apr 6, 2021	Simplified description
1.2	May 6, 2021	Update output power data
1.3	Feb 28, 2022	Update ISD2361YYI and ISD2361SYI difference
1.4	Mar 28, 2022	Update ordering package information
1.5	May 18, 2022	Update feature description Update GPIO[9:6] description

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.