aPR33A3K

CPU Serial Mode (C1.1)

Datasheet

Recording voice IC

APLUS INTEGRATED CIRCUITS INC.

Address:

3 F-10, No. 32, Sec. 1, Chenggung Rd., Taipei, Taiwan 115, R.O.C.

TEL:

886-2-2782-9266

FAX:

886-2-2782-9255

WEBSITE :

http://www.aplusinc.com.tw

Technology E-mail:

edit@aplusinc.com.tw

Sales E-mail:

sales@aplusinc.com.tw

■ FEATURES

- Operating Voltage Range: 3V ~ 6.5V
- Single Chip, High Quality Audio/Voice Recording & Playback Solution
  - No External ICs Required
  - Minimum External Components
- User Friendly, Easy to Use Operation
  - Programming & Development Systems Not Required
- 680 sec. Voice Recording Length in aPR33A3K
- Powerful 16-Bits Digital Audio Processor.
- Nonvolatile Flash Memory Technology
  - No Battery Backup Required
- External Reset pin.
- Powerful Power Management Unit
  - Very Low Standby Current: 1uA
  - Low Power-Down Current: 15uA
  - Supports Power-Down Mode for Power Saving
- Built-in Audio-Recording Microphone Amplifier
  - No External OPAMP or BJT Required
  - Easy to PCB layout
- Configurable analog interface
  - Differential-ended MIC pre-amp for Low Noise
  - High Quality Line Receiver
- High Quality Analog to Digital and PWM module
  - Resolution up to 16-bits
- Up To Maximum 1024 Voice Sections controlled through 5 pins only
- Built-in Memory-Management System

■ DESCRIPTION

Today's consumers demand the best in audio/voice. They want crystal-clear sound wherever they are in whatever format they want to use. APLUS delivers the technology to enhance a listener's audio/voice experience.

The aPR33A3K is a powerful audio processor along with high performance audio analog-to-digital converters (ADCs) and digital-to-analog converters (DACs). The aPR33A3K is a fully integrated solution offering high performance and unparalleled integration with analog input, digital processing and analog output functionality. The aPR33A3K is incorporates all the functionality required to perform demanding audio/voice applications. High quality audio/voice systems with lower bill-of-material costs can be implemented with the aPR33A3K because of their integrated analog data converters and full suite of quality-enhancing features such as sample-rate convertor.

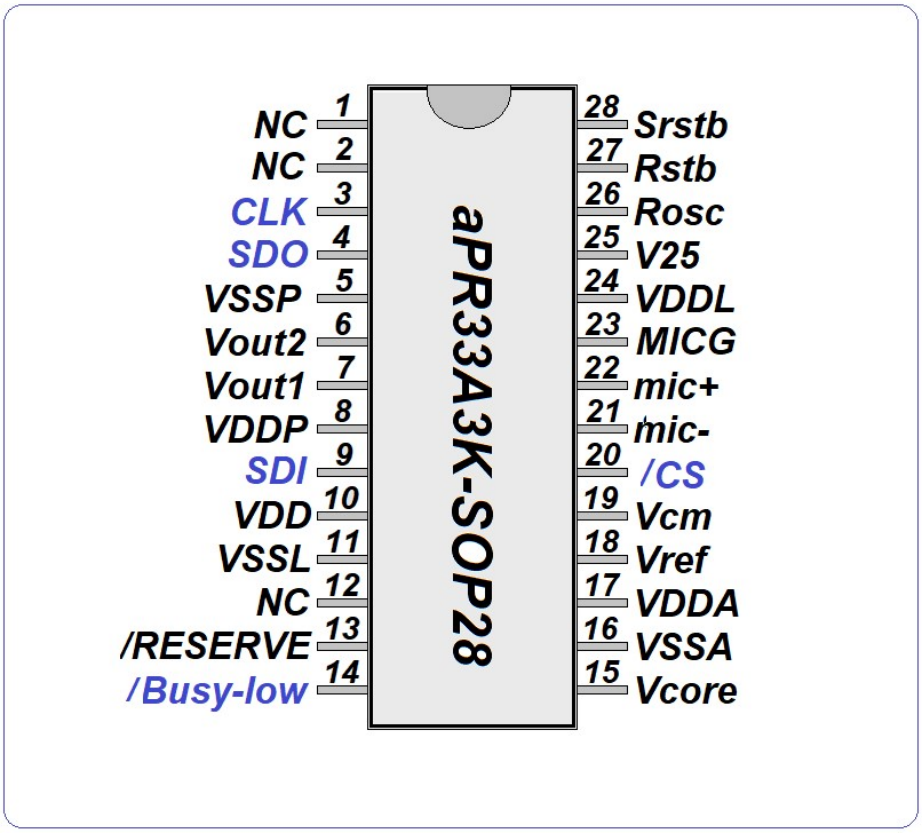
The aPR33A3K-C1.1. is specially designed for simple CPU interface, user can record or playback up to 1024 voices with only 5 I/Os. This mode has a built in one complete memory-management system. User does not need to be burdened with complicated memory distribution problems. It only needs

simple instruction to process the audio/voice recording & playback, which will largely shorten the

developing time. The IC also provides the power-management system. Users can set the IC in

power-down mode when unused. It can effectively reduce electric current consuming to 15uA and increase the usage time in any projects powered by batteries.

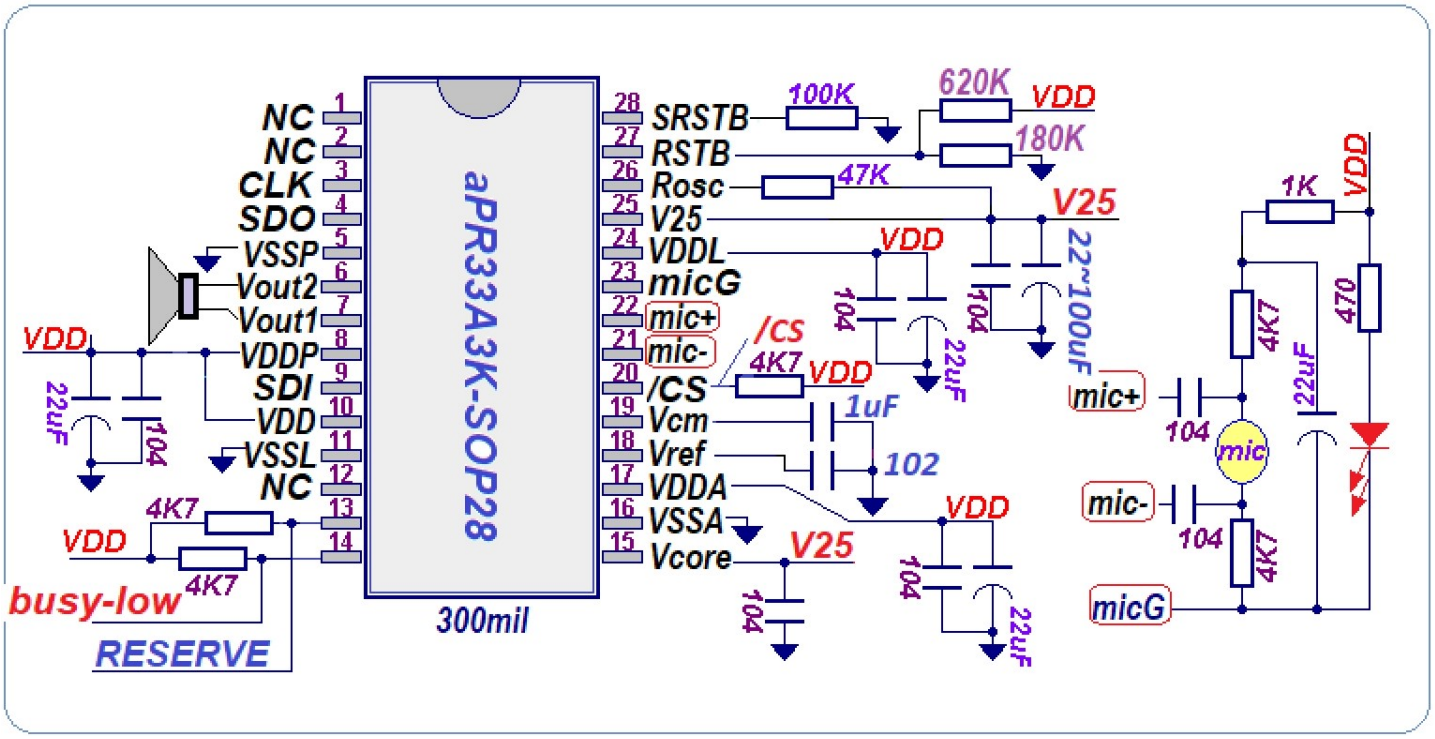
■ PIN CONFIGURATION :

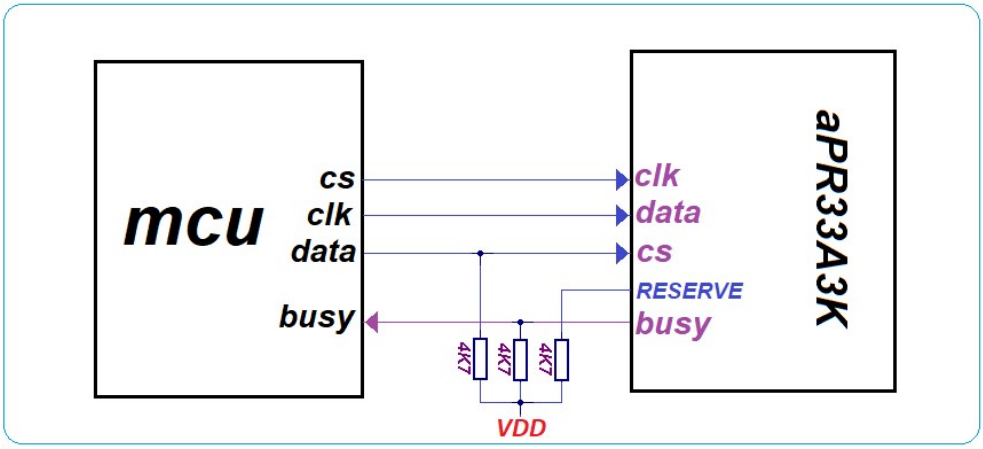


■ PIN DESCRIPTION

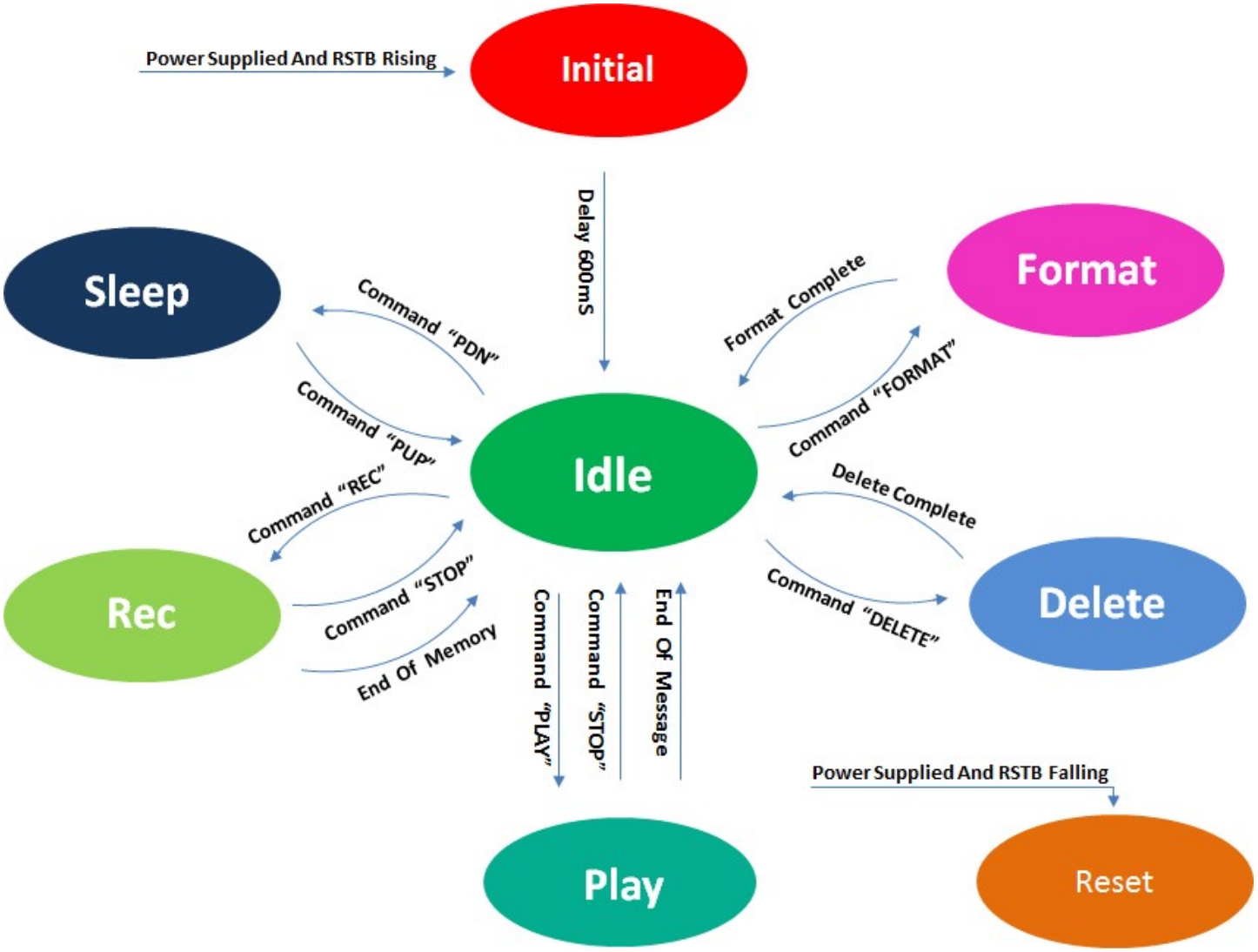
Pin Names	Pin No	TYPE	Description
VDDP VDD VDDA VDDL	8 10 17 24		Positive power supply.
VSSP VSSL VSSA	5 11 16		Power ground.
V_{25}	25		Internal LDO output.
V_{CORE}	15		Positive power supply for core.
V_{REF}	18		Reference voltage.
V_{CM}	19		Common mode voltage.
Rosc	26	INPUT	Oscillator resistor input.
RSTB	27	INPUT	Reset. (Low active)
SRSTB	28	INPUT	System reset, pull-down a resistor to the VSSL.
MIC+ MIC-	22 21	INPUT	Microphone differential input.
MICG	23	OUTPUT	Microphone ground.
VOUT2 VOUT1	6 7	OUTPUT	PWM output to drive speaker directly.
/CS	20	INPUT	Chip select. (Low active)
SCK	3	INPUT	Serial clock.
SDI	9	INPUT	Serial data input.
SDO	4	OUTPUT	Serial data output.
/BUSY	14	OUTPUT	System busy output.
/RESERVE	13	OUTPUT	Output reserve.

■ CONNECTION DIAGRA





■ SYSTEM STATE

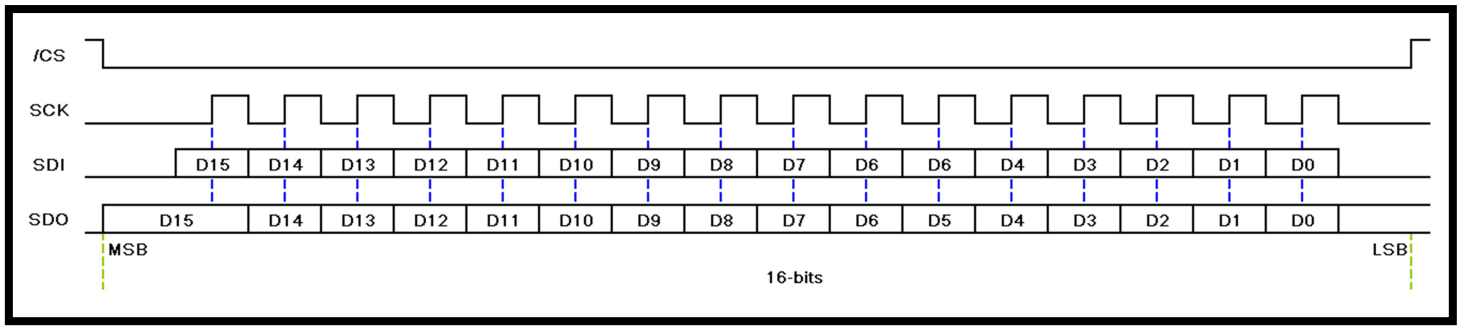


NOTICE: User must execute the Format command when using the IC for the first time.

■ SERIAL COMMAND

The aPR33A3K C1.1 is specially designed for simple CPU interface. The IC is controlled by command sent from a host CPU. The /CS pin is used to select the IC. The SCK and SDI pins are used to input command of word datatype into the chip while SDO and BUSY are output pins from the chip to the host CPU for feedback response.

Each input command contains 16-bit data. The following list shows the command format and the summary of the available commands :



Command	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
STOP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DELETE	0	0	0	1	0	0	Voice No In Binary									
REC	0	0	1	0	0	0	Voice No In Binary									
PLAY	0	0	1	1	0	0	Voice No In Binary									
PUP	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0
PDN	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
FORMAT	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1

NOTICE: User must execute the Format command when using the IC for the first time.

■ FORMAT command :

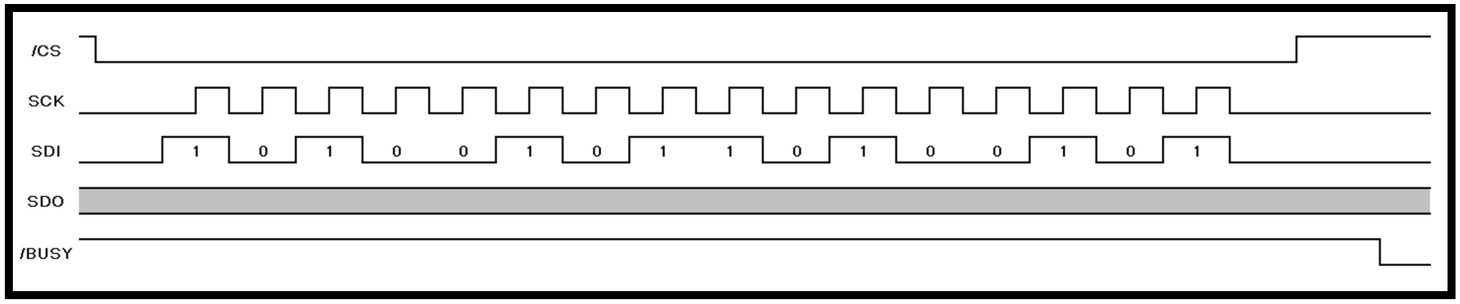
The FORMAT command is used to clear and re-configure data memory.

The FORMAT command is 1010010110100101 in binary, from bit-15 to bit-0.

After the FORMAT command is sent, the /BUSY pin will be pulled to low to indicate the format operation has started. When the format operation is finished, the /BUSY pin will be released back to high.

All of the voice in the memory will be deleted after executing format operation.

NOTICE: User must execute the Format command when using the IC for the first time.



■ REC command :

The REC command is used to record the voice and store to the IC with a specified voice number.

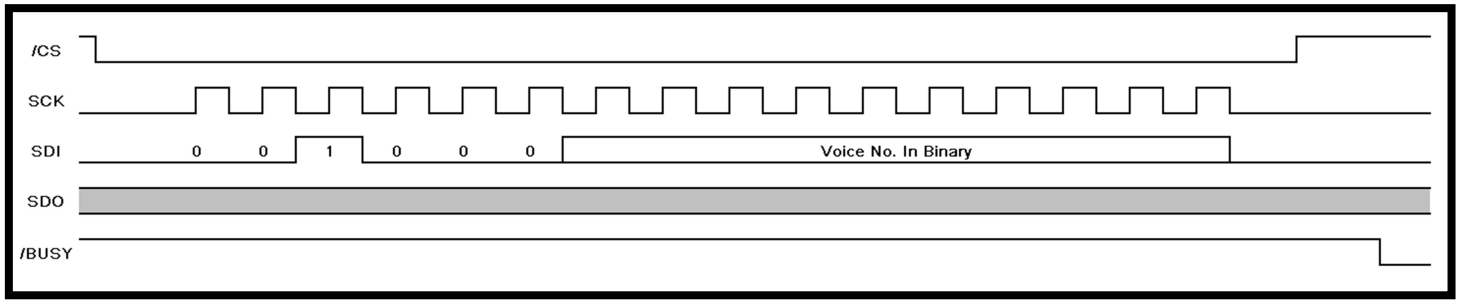
The REC command contains two parts: the command 001000 in binary, from bit-15 to bit-10 and the voice number in binary from bit-9 to bit-0. User can specify the voice number from 0 to 1023.

After the REC command is sent, the /BUSY pin will be pulled to low and playback a “beep” tone to indicate the record operation has started.

During the record operation, the /BUSY pin will keep low, and any command except STOP will be ignored.

The record operation will continue until user sends the STOP command or when the memory is full. In this case, the /BUSY pin will be released back to high and playback two “beeps” to indicate the record operation is finished.

If data already exists in the specified voice number or if the memory is full, the /BUSY pin will not drive to low and the REC command will not be executed. User must use the DELETE command to clear specified voice number and run the REC command again.



■ PLAY command :

The PLAY command is used to playback the voice in the specified voice number.

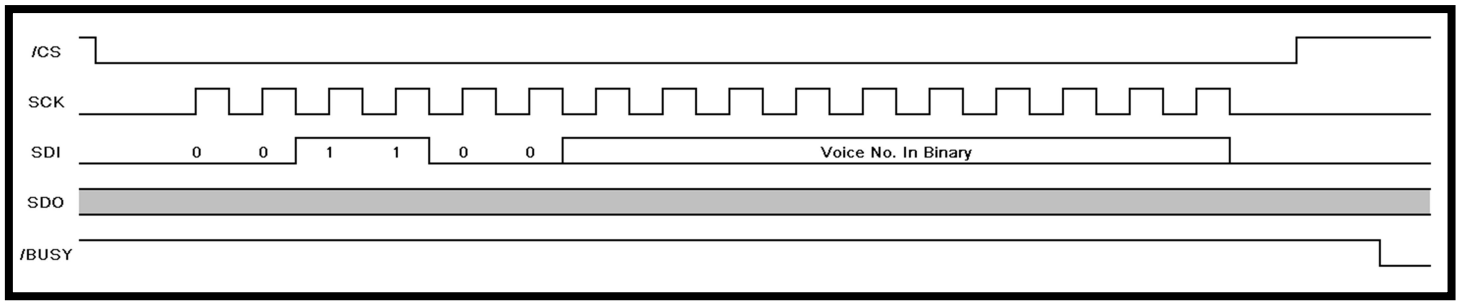
The PLAY command contains two parts : the command 001100 in binary, from bit-15 to bit-10, and the voice number in binary from bit-9 to bit-0. User can specify the voice number from 0 to 1023.

After the PLAY command is sent, the /BUSY pin will be pulled low to indicate the playback operation has started.

During the playback operation, the /BUSY pin will keep low, and any command except STOP will be ignored.

The playback operation will continue until users sends the STOP command or when the playback is finished The /BUSY pin will be released back to high to indicate the playback operation is finished.

If the specified voice number is empty, the /BUSY pin will keep high and will not playback.



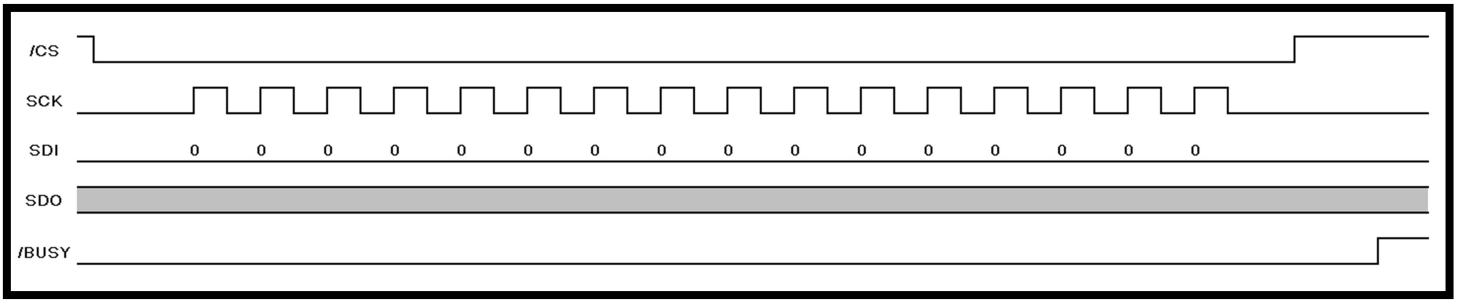
■ STOP command :

The STOP command is used to stop the current operation.

The STOP command is 0 in binary, from bit-15 to bit-0.

After the STOP command is sent, the /BUSY pin will be released back to high to indicate the current operation is finished.

The STOP command is applicable only in play or record operations.



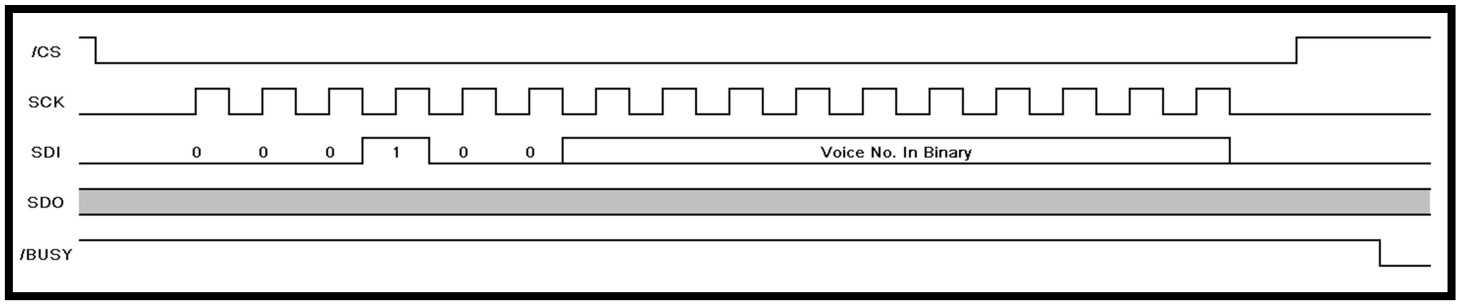
■ DELETE command :

The DELETE command is used to delete the data in the specified voice number.

The DELETE command contains two parts: the command 000100 in binary, from bit-15 to bit-10, and the voice number in binary from bit-9 to bit-0. User can specify the voice number from 0 to 1023.

After the DELETE command is sent, the /BUSY pin will be pulled to low to indicate the delete operation has started. When the delete operation is finished, the /BUSY pin will be released back to high.

The data in the memory space of the specified voice number will be erased after the delete operation. User can space by deleting unwanted data or recording.



■ PDN command :

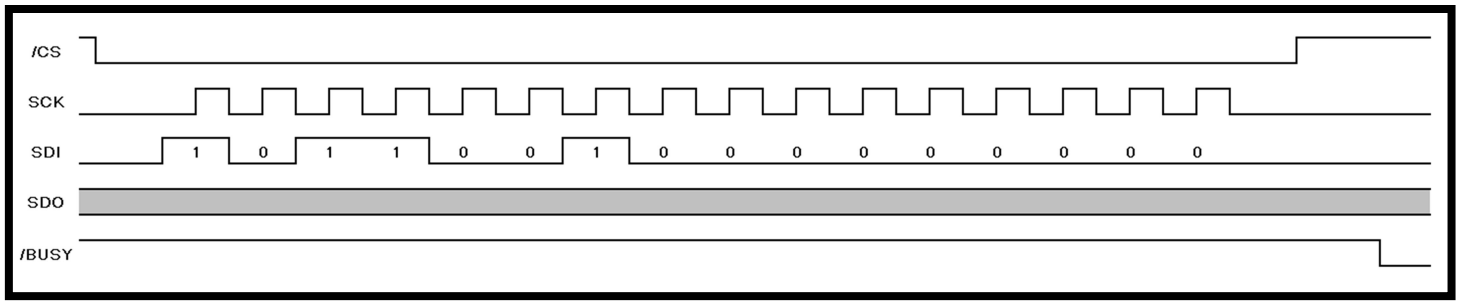
The PDN command is used to enter the power-down mode.

The PDN command is 10110010 in binary, from bit-15 to bit-8.

After the PDN command is sent, the /BUSY pin will be pulled to low to indicate the power-down operation has started.

When the IC is in the sleep mode, the current consumption is reduced to I_{PDN} and any command except PUP will be ignored.

When the power-down operation is finished and IC is out of the sleep mode, the /BUSY pin will be back to high.



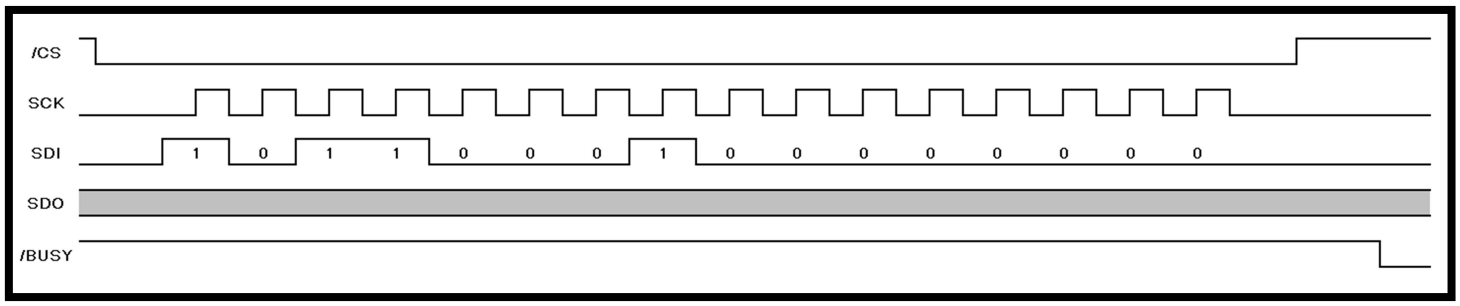
■ PUP command :

The PUP command is used to power up from sleep mode.

The PUP command is 10110001 in binary, from bit-15 to bit-8.

After the PUP command is sent, the /BUSY pin will be pulled to low to indicate the power up operation has started. When the power-up operation is finished and the IC is in idle mode, the /BUSY pin will be back to high and the current is $I_{OP(IDLE)}$.

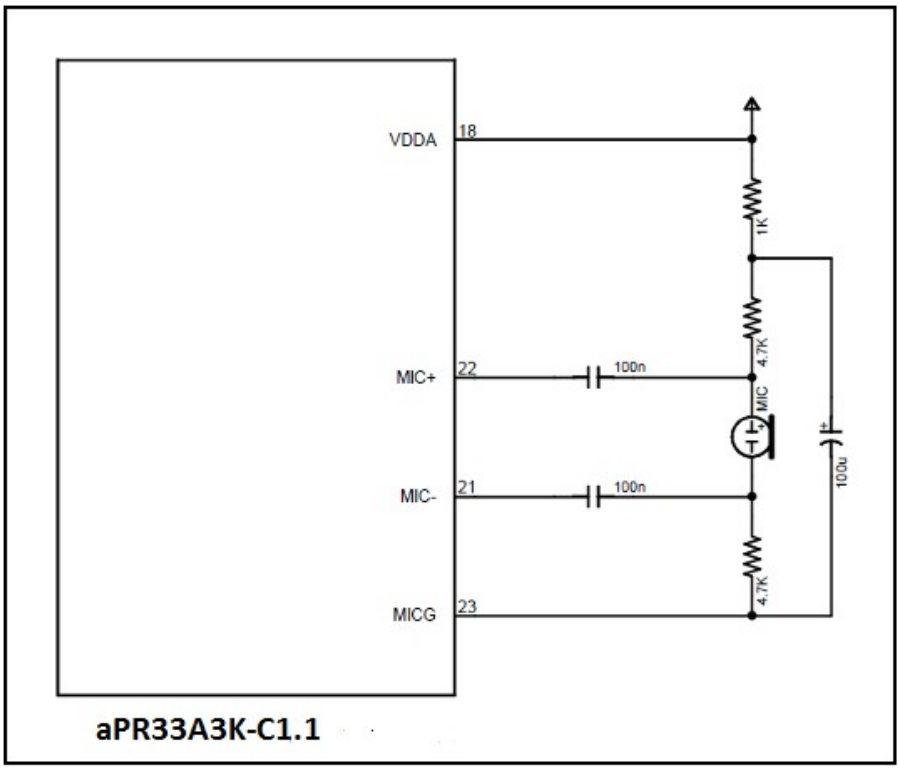
User can execute REC, PLAY DELETE or other command in idle mode.



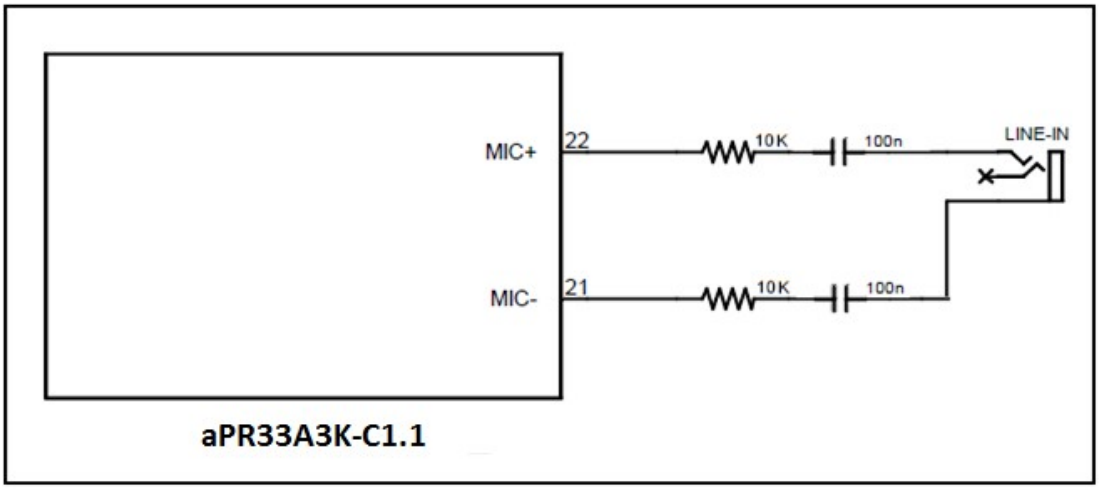
■ VOICE INPUT

The aPR33A3K-C1.1 supports single channel voice input by microphone or line-in. The following figure shows the circuit for different input methods: microphone, line-in and both.

Note: The 10K resistor used for input signal adjust, and the value are just for reference.



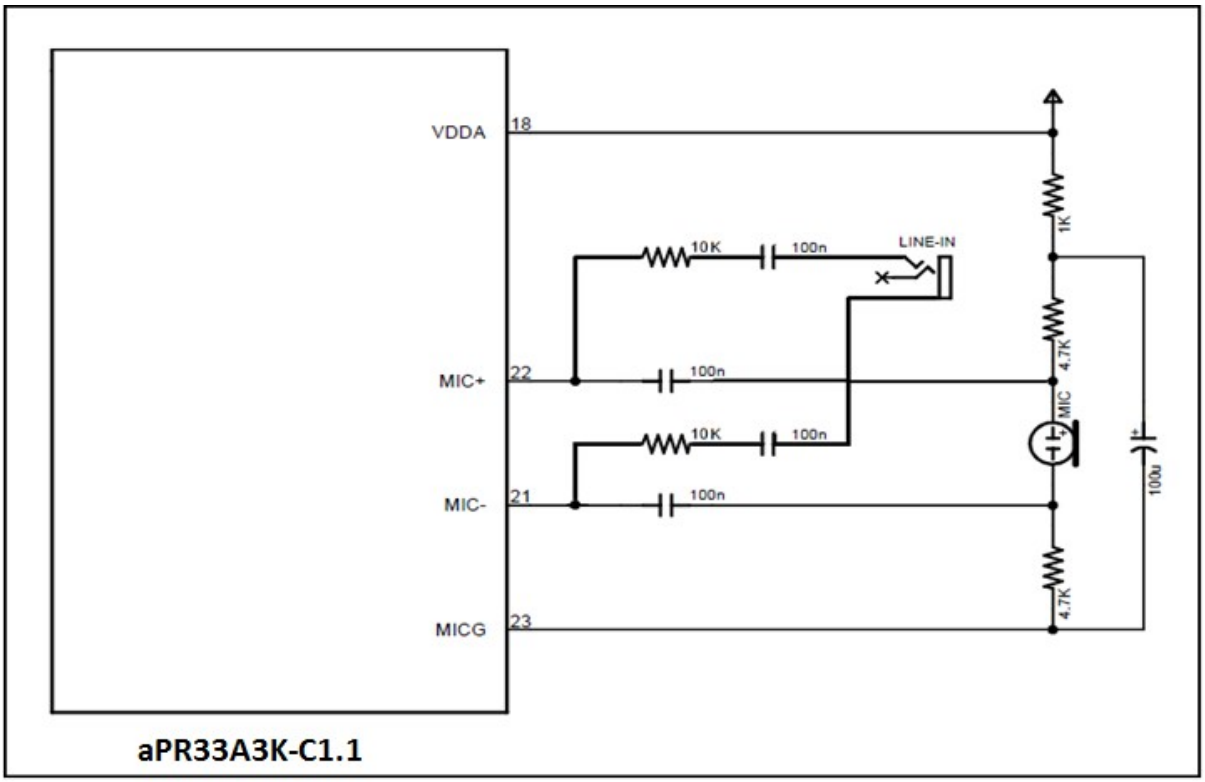
(A) Microphone



Note: The 10K resistor used for input signal adjust, and the value are just for reference.

(B) Line-In

.



Note: The 10K resistor used for input signal adjust, and the value are just for reference.

(C) Microphone + Line-In

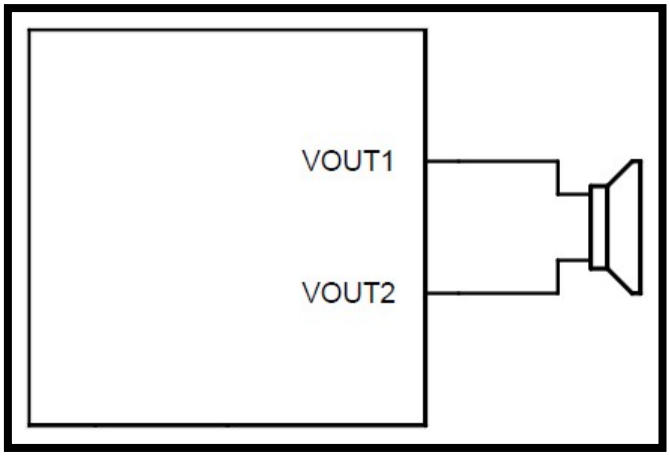
■ VOICE OUTPUT

The aPR33A3K –C1.1 support 2 voice output modes, PWM and DAC.

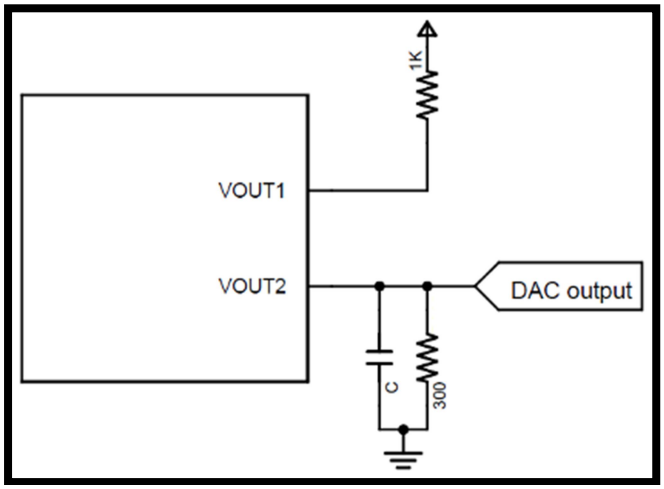
The PWM mode uses VOUT1 and VOUT2 pins to drive the speaker directly without external components to save cost.

The DAC mode uses VOUT2 pin to output current signal. Users can use the signal to drive audio amplifier or mix with other components in their applications to provide higher voice volume.

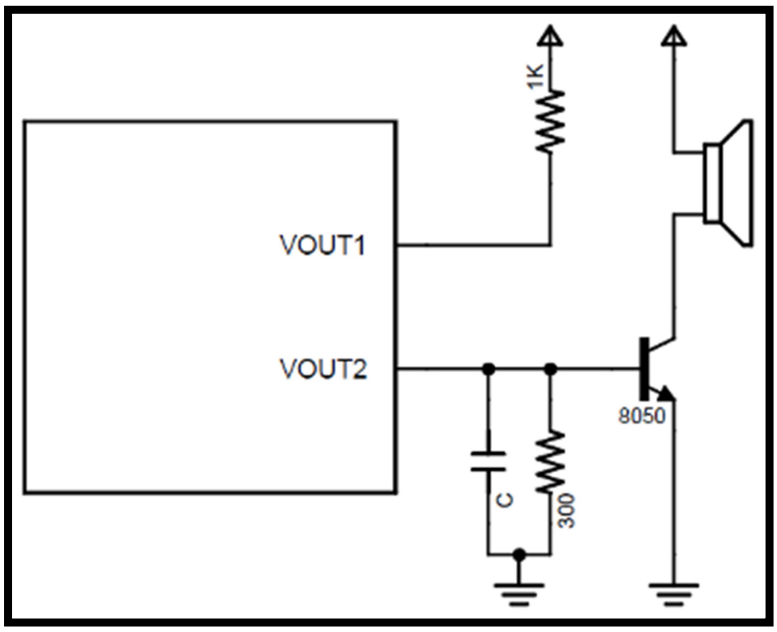
The following figure shows the circuit for different output methods: PWM, DAC, DAC with transistor and DAC with audio amplifier AP4890B.



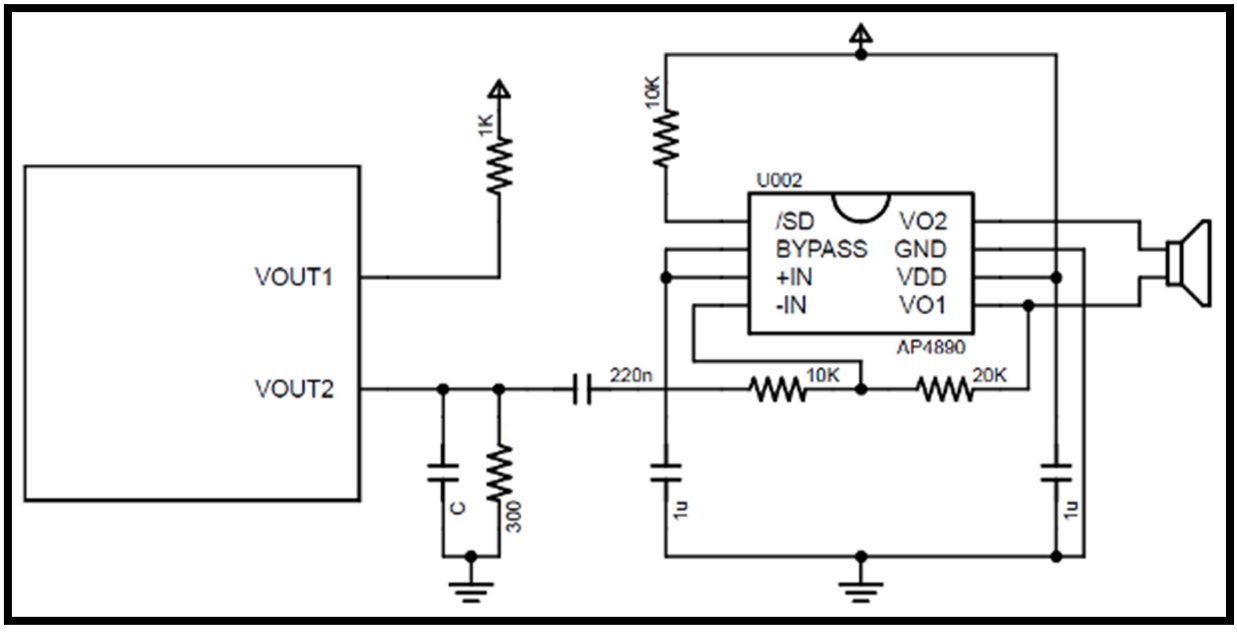
(A) PWM



(B) DAC



(C) DAC with transistor



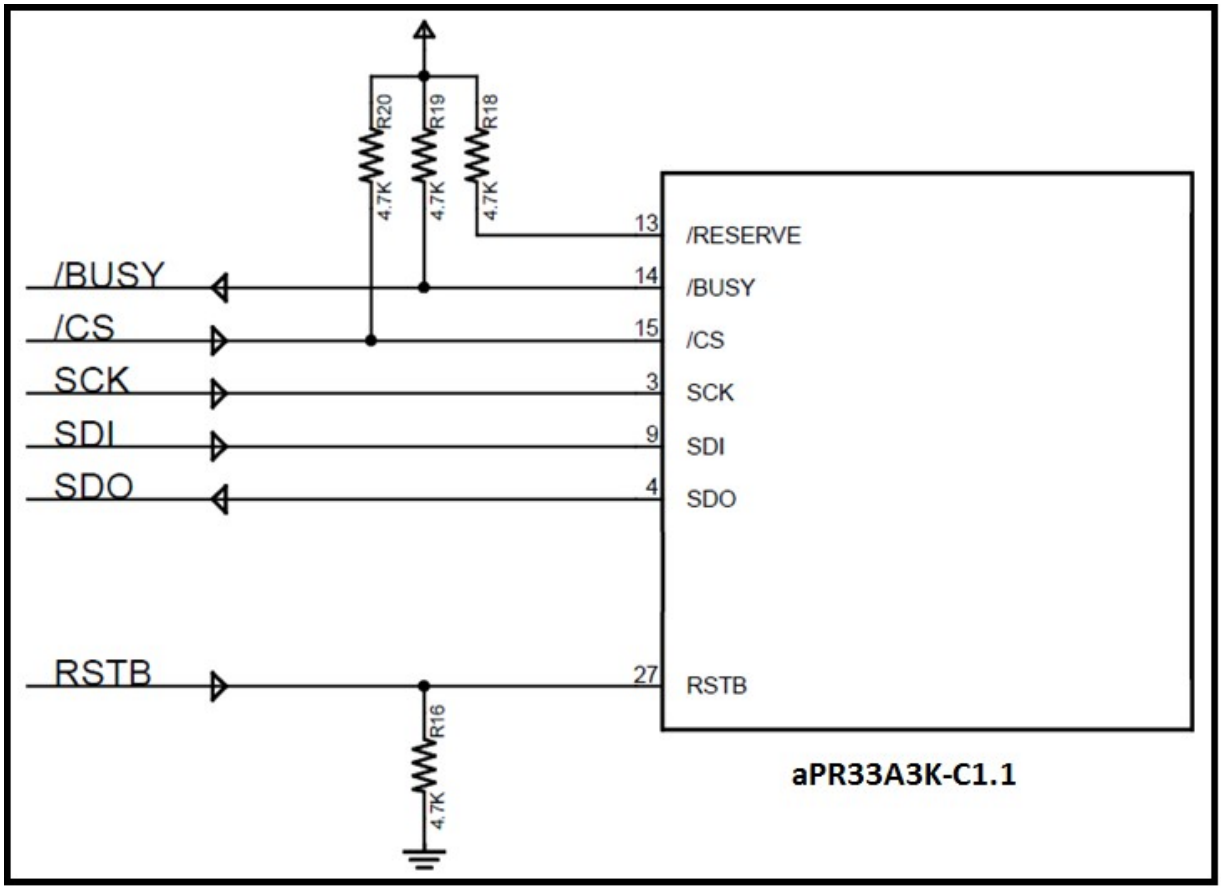
(D) DAC with audio amplifier AP4890K

■ RESET :

aPR33A3K-C1.1 can enter in standby mode by setting the RSTB pin to low. When in the standby mode, the current consumption is reduced to I_{SB}. Any operation will be stopped and user cannot execute any new command in this mode.

The standby mode will continue be active until the RSTB pin goes to high. The IC will start to reinitialize and playback a “beep” tone to indicate it is entering in idle mode.

User can get less current consumption by controlling the RSTB pin specially in some applications which standby current is a concern.


R20
R19
R18
4.7K
4.7K
4.7K
13
/RESERVE
/BUSY
14
/BUSY
/CS
15
/CS
SCK
3
SCK
SDI
9
SDI
SDO
4
SDO
RSTB
27
RSTB
R16
4.7K
aPR33A3K-C1.1

The below C code example shows all the operating instructions.

```
//============================================================================
// I/O Define
//=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
// P_CS       : aPR33A3Kx /CS pin.
// P_SCK      : aPR33A3Kx SCK pin.
// P_MOSI     : aPR33A3Kx SDI pin.
// P_MISO     : aPR33A3Kx SDO pin.
// P_BUSY: : aPR33A3Kx /BUSY pin.
// KEY_REC : Key for record, high active.
// KEY_PLAY: Key for playback, high active
//==================================================================================
// Type Define
//=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
// unsigned char: 1-byte.
// unsigned int : 2-bytes.
typedef union                  UTYPE
{
       unsigned int            TWORD;                             // Dual-byte
       struct
       {
              unsigned            TBIT0:1;
              unsigned            TBIT1:1;
              unsigned            TBIT2:1;
              unsigned            TBIT3:1;
              unsigned            TBIT4:1;
              unsigned            TBIT5:1;
              unsigned            TBIT6:1;
              unsigned            TBIT7:1;
              unsigned            TBIT8:1;
              unsigned            TBIT9:1;
              unsigned            TBIT10:1;
              unsigned            TBIT11:1;
              unsigned            TBIT12:1;
              unsigned            TBIT13:1;
              unsigned            TBIT14:1;
```

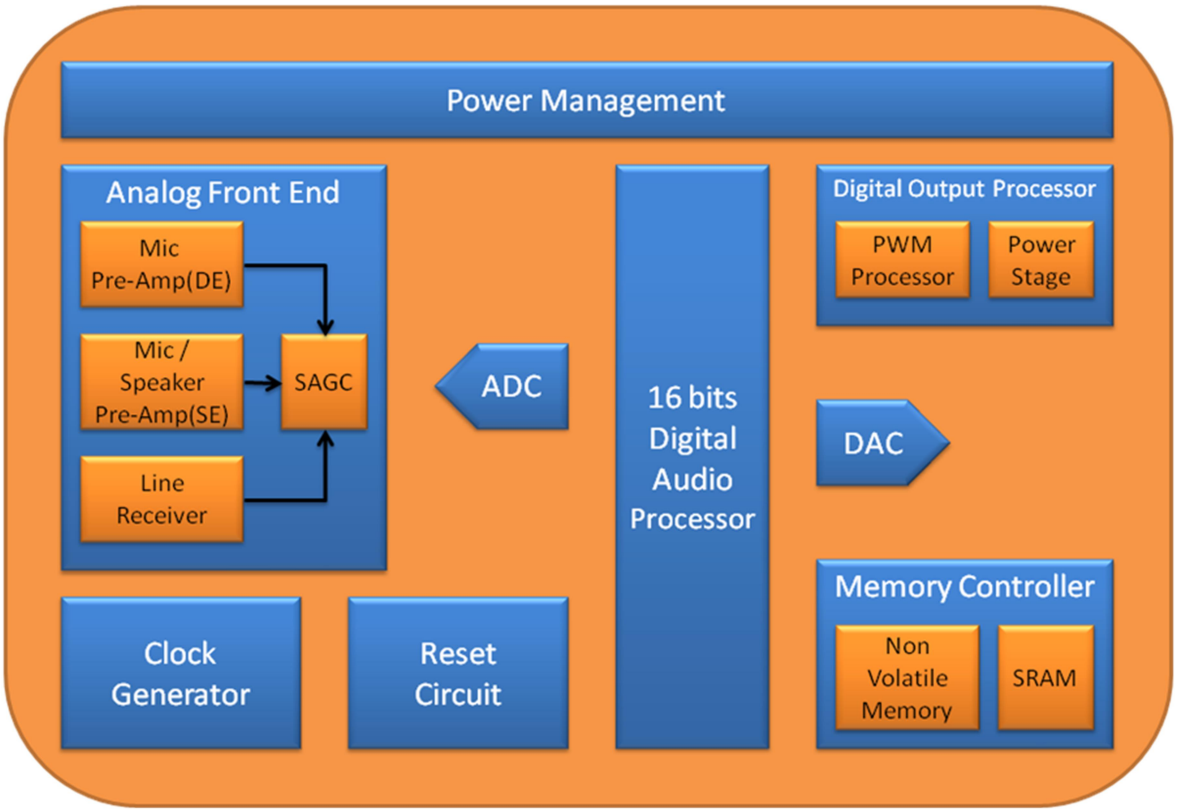
```
        unsigned            TBIT15:1;
    };
}UTYPE;
//=====================================================================
// Prototype
//=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
void                    CS(BOOL Value)          {   P_CS =Value;                        }
void                    SCK(BOOL Value)         {   P_SCK=Value;    Delay_500nS();  }
void                    SDO(BOOL Value)         {   P_MOSI=Value;   Delay_500nS();  }
BOOL                      SDI()                 {   return(P_MISO);                     }
//=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
unsigned int            SendCmd(unsigned int Value)
{
    UTYPE                   TxData, RxData;
    //-----------------------------------------------------------------------------------------
    TxData.TWORD = Value;
    //-----------------------------------------------------------------------------------------
                    SDO(TxData.TBIT15);     RxData.TBIT15=SDI();    SCK(1);
    SCK(0);         SDO(TxData.TBIT14);     RxData.TBIT14=SDI();    SCK(1);
    SCK(0);         SDO(TxData.TBIT13);     RxData.TBIT13=SDI();    SCK(1);
    SCK(0);         SDO(TxData.TBIT12);     RxData.TBIT12=SDI();    SCK(1);
    SCK(0);         SDO(TxData.TBIT11);     RxData.TBIT11=SDI();    SCK(1);
    SCK(0);         SDO(TxData.TBIT10);     RxData.TBIT10=SDI();    SCK(1);
    SCK(0);         SDO(TxData.TBIT9);      RxData.TBIT9=SDI();     SCK(1);
    SCK(0);         SDO(TxData.TBIT8);      RxData.TBIT8=SDI();     SCK(1);
    SCK(0);         SDO(TxData.TBIT7);      RxData.TBIT7=SDI();     SCK(1);
    SCK(0);         SDO(TxData.TBIT6);      RxData.TBIT6=SDI();     SCK(1);
    SCK(0);         SDO(TxData.TBIT5);      RxData.TBIT5=SDI();     SCK(1);
    SCK(0);         SDO(TxData.TBIT4);      RxData.TBIT4=SDI();     SCK(1);
    SCK(0);         SDO(TxData.TBIT3);      RxData.TBIT3=SDI();     SCK(1);
    SCK(0);         SDO(TxData.TBIT2);      RxData.TBIT2=SDI();     SCK(1);
    SCK(0);         SDO(TxData.TBIT1);      RxData.TBIT1=SDI();     SCK(1);
    SCK(0);         SDO(TxData.TBIT0);      RxData.TBIT0=SDI();     SCK(1);
    SCK(0);
    //-----------------------------------------------------------------------------------------
    return (RxData.TWORD);
}
```

```
//=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
void    PUP(void)                          {   CS(0);   Delay_10mS();   SendCmd(0xB100);   CS(1);   }
void    PDN(void)                          {   CS(0);   Delay_10mS();   SendCmd(0xB200);   CS(1);   }
void    FORMAT(void)                       {   CS(0);   Delay_10mS();   SendCmd(0xA5A5);   CS(1);   }
void    STOP(void)                         {   CS(0);   Delay_10mS();   SendCmd(0x0000);   CS(1);   }
void    DELETE(unsigned int VoiceNo)       {  CS(0); Delay_10mS(); SendCmd(0x1000|(VoiceNo&0x03FF));  CS(1);  }
void    REC(unsigned int VoiceNo)          {  CS(0); Delay_10mS(); SendCmd(0x2000|(VoiceNo&0x03FF));  CS(1);  }
void    PLAY(unsigned int VoiceNo)         {  CS(0);  Delay_10mS(); SendCmd(0x3000|(VoiceNo&0x03FF));  CS(1);  }
//=====================================================================================
// Main
//=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
void    main(void)
{
    //--------------------------------------------------------------------------------------------
    Init_IO();                              // Initial I/O: /CS=0, SCK=0, SDI=0, SDO=input, /BUSY=input.
    Delay_600mS();
    while(!P_BUSY);                         // Check for /BUSY pull-up.
    //--------------------------------------------------------------------------------------------
    // Format chip
    FORMAT();
    while(P_BUSY);    while(!P_BUSY);             // Wait for format operation start & finished.
    //--------------------------------------------------------------------------------------------
    while(1)
    {
        if(KEY_REC)
        {
            DELETE(0x0000);
            while(P_BUSY);   while(!P_BUSY);          // Wait for delete operation start & finished.

            REC(0x0000);     Delay_10mS();            // Record the No.0 voice.
            while((KEY_REC)&(!P_BUSY));               // Wait for release record key or full of memory.

            STOP();
            while(!P_BUSY);                           // Wait for record finished.
        }
        if(KEY_PLAY)
        {
```

```
            PLAY(0x0000);     Delay_10mS();                    // Play the No.0 voice.
            while((KEY_PLAY)&(!P_BUSY));                 // Wait for release play key or end of voice.
            STOP();
            while(!P_BUSY);                                  // Wait for play finished.
        }
    }
    //-------------------------------------------------------------------------------------
    return ;
}
//=============================================================================================
```

■ BLOCK DIAGRAM

Figure 1. Block Diagram



■ ABSOLUTE MAXIMUM RATINGS

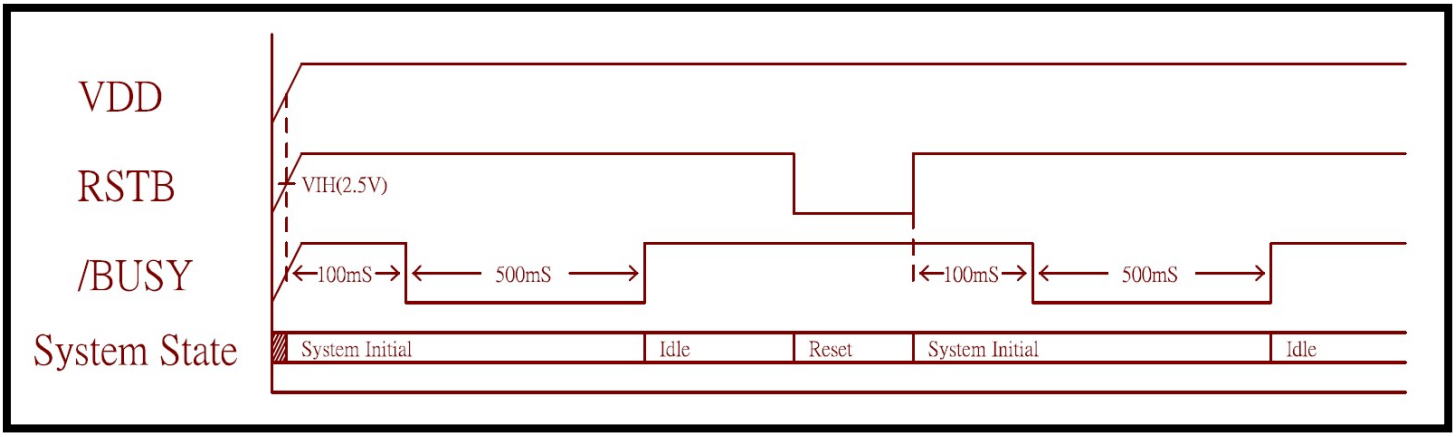
Symbol	Rating	Unit
VDD – VSS	-0.3 ~ +10.0	V
V_{IN}	VSS-0.3 < V_{IN} < VDD+0.3	V
V_{OUT}	VSS < V_{OUT} < VDD	V
T(Operating)	-40 ~ +85	℃
T(Junction)	-40 ~ +125	℃
T(Storage)	-40 ~ +125	℃

■ DC CHARACTERISTICS:

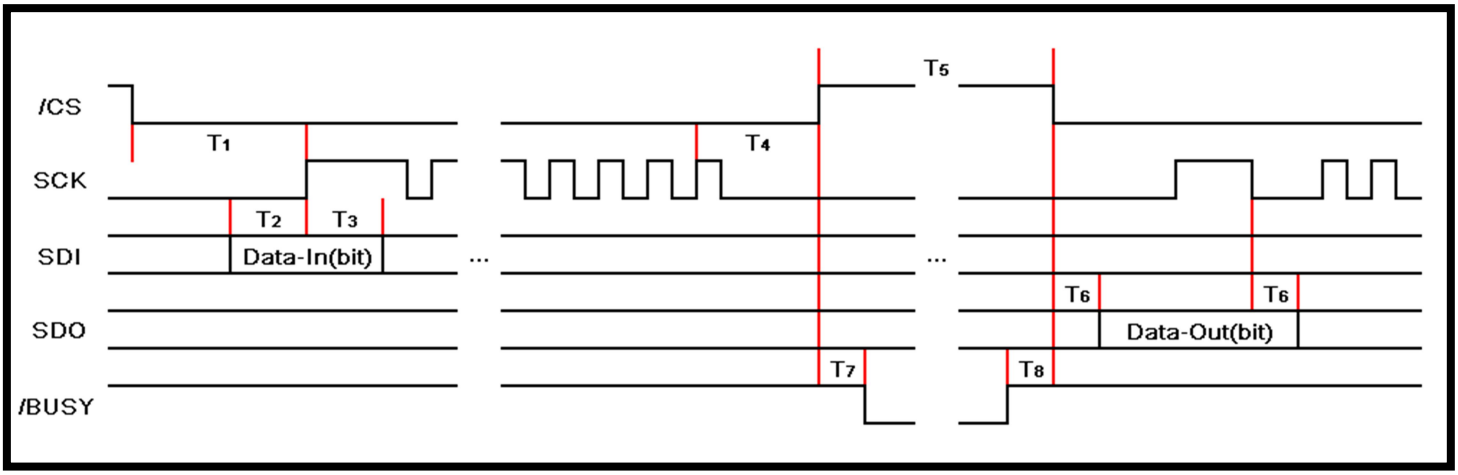
Symbol	Parameter	Min.	Typ.	Max.	Unit	Conditions
VDD	Operating Voltage	3.0		6.5	V	
I_{SB}	Standby Current			1	μA	
I_{PDN}	Power-Down Current		15	20	μA	
$I_{OP(IDLE)}$	Operating Current (Idle)		20		mA	VDD = 5V
$I_{OP(REC)}$	Operating Current (Record)		35		mA	VDD = 5V
$I_{OP(PLAY)}$	Operating Current (Playback)		25		mA	VDD = 5V
V_{IH}	"H" Input Voltage	2.5			V	
V_{IL}	"L" Input Voltage			0.6	V	
I_{VOUT}	VOUT Current		185		mA	
I_{OH}	O/P High Current		8		mA	VDD = 5V / VOH=4.5V
I_{OL}	O/P Low Current		14		mA	VDD = 5V / VOH=0.5V
R_{NPIO}	Input pin pull-down resistance		300		KΩ	External floating or drive low.
			1		MΩ	External drive high.
R_{UPIO}	Input pin pull-up resistance		4.7		KΩ	
△ Fs/Fs	Frequency stability			5	%	VDD = 5V ± 1.0V
△ Fc/Fc	Chip to chip Frequency Variation			5	%	Also apply to lot to lot variation.

■ AC CHARACTERISTICS

Power Up & Reset



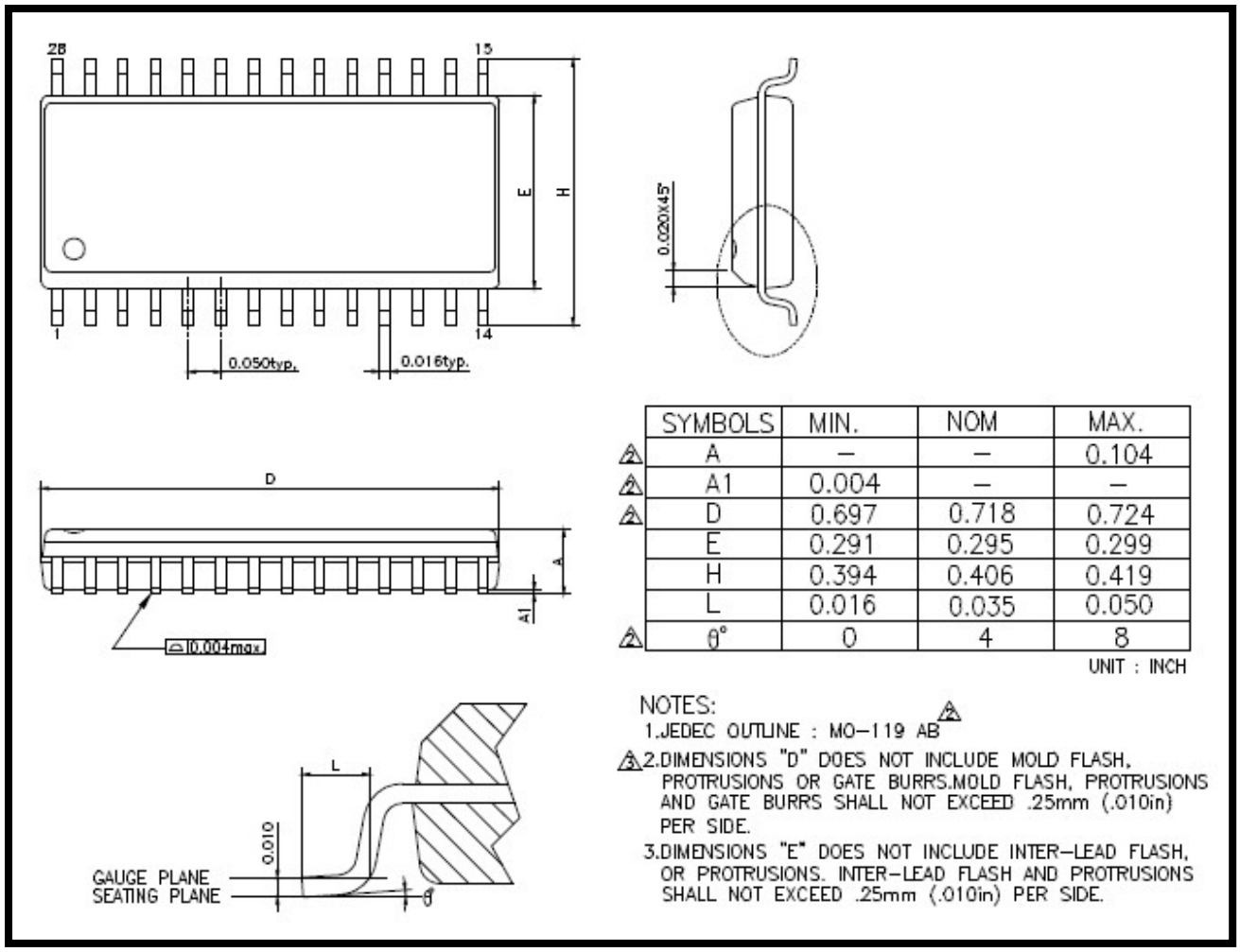
Command



Symbol	Parameter	Min.	Typ.	Max.	Unit	Conditions
T1	CS Setup Time	10	--	--	mS	VDD=5.0V
T2	Data-In Setup Time	500	--	--	nS	VDD=5.0V
T3	Data-In Hold Time	500	--	--	nS	VDD=5.0V
T4	/CS Hold Time	500	--	--	nS	VDD=5.0V
T5	/CS High Time	10	--	--	mS	VDD=5.0V
T6	Data-Out Setup Time	--	--	500	nS	VDD=5.0V
T7	BUSY Setup Time	--	--	10	mS	VDD=5.0V
T8	BUSY Hold Time	--	--	10	mS	VDD=5.0V

■ PACKAGE INFORMATION :

28Pin 300mil SOP Package



	SYMBOLS	MIN.	NOM	MAX.
2	A	–	–	0.104
2	A1	0.004	–	–
2	D	0.697	0.718	0.724
	E	0.291	0.295	0.299
	H	0.394	0.406	0.419
	L	0.016	0.035	0.050
2	θ°	0	4	8

UNIT : INCH

NOTES:
1.JEDEC OUTLINE : MO-119 AB 2
3 2.DIMENSIONS "D" DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED .25mm (.010in) PER SIDE.
3.DIMENSIONS "E" DOES NOT INCLUDE INTER-LEAD FLASH, OR PROTRUSIONS. INTER-LEAD FLASH AND PROTRUSIONS SHALL NOT EXCEED .25mm (.010in) PER SIDE.

■ HISTORY

Ver. A (2022/3/15)

- Original version data sheet for aPR33A3K- C1.1.