

SN76489

1.0

Generated by Doxygen 1.9.1

1 LICENSE	1
2 SN76489 Sound IC library	3
2.1 Release Versions	3
2.1.1 Current	3
2.1.2 Past	3
2.2 Requirements	3
2.3 Building	4
2.4 Documentation	4
2.4.1 Example Code	4
3 File Documentation	5
3.1 LICENSE.md File Reference	5
3.2 README.md File Reference	5
3.3 sn76489.h File Reference	5
3.3.1 Function Documentation	6
3.3.1.1 getSN76489_FreqDiv()	6
3.3.1.2 initSN76489()	6
3.3.1.3 setSN76489noise_attn()	6
3.3.1.4 setSN76489noiseCtrl()	6
3.3.1.5 setSN76489voice_attn()	7
3.3.1.6 setSN76489voice_freq()	7
Index	9

Chapter 1

LICENSE

license: MIT

Copyright 2022 Johnathan Convertino

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Chapter 2

SN76489 Sound IC library

Software Library for TI SN76489 sound chips.

author: Jay Convertino

data: 2022.01.31

license: MIT

2.1 Release Versions

2.1.1 Current

- release_v0.0.1

2.1.2 Past

- none

2.2 Requirements

- xc8-cc V2.32
- PIC18F45K50 (MCU can be changed in makefile)
- PICerino development board

2.3 Building

- make : builds all
- make dox_gen : doxygen only
- make test : test only
- make libSN76489.a : static library only
- make clean : remove all build outputs.

2.4 Documentation

- See doxygen generated document
- Method for ready check is universal, NOT efficient. Optimize send data for your application!

2.4.1 Example Code

```
void main(void)
{
    struct s_sn76489 sn76489;
    /* OSCCON SETUP */
    OSCCONbits.IRCF = 0x7;
    OSCCONbits.OSTS = 0;
    OSCCONbits.SCS = 0x3;
    OSCCON2bits.PLEN = 1;
    /* PORT E SETUP */
    INTCON2bits.nRBPU = 1;
    /* disable analog inputs */
    ANSELA = 0;
    ANSELC = 0;
    ANSELD = 0;
    ANSELE = 0;
    /* wait for chip to be ready */
    __delay_ms(10);
    initSN76489port(&sn76489, &TRISA, &TRISD, &TRISC, 6, 7, 0);
    /* mutes all channels as default */
    initSN76489(&sn76489, &LATA, &LATD, &PORTC);
    /* voice one freq */

    /* set attenuation */
    setSN76489voice_attn(&sn76489, 1, 2);

    /* set frequency to 440 hz */
    setSN76489voice_freq(&sn76489, 1, 254);

    /* play this lovely tune forever */
    for(;;);
}
```


Chapter 3

File Documentation

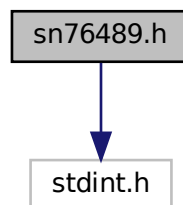
3.1 LICENSE.md File Reference

3.2 README.md File Reference

3.3 sn76489.h File Reference

```
#include <stdint.h>
```

Include dependency graph for sn76489.h:



Functions

- void [initSN76489](#) ()
Initialize sn76489 to mute all channels.
- uint16_t [getSN76489_FreqDiv](#) (uint32_t refClk, uint32_t voiceFreq)
Calculate frequency from hertz to binary value.
- void [setSN76489voice_freq](#) (uint8_t voice, uint16_t freqDiv)
Set sn76489 voice frequency.
- void [setSN76489voice_attn](#) (uint8_t voice, uint8_t attenuate)
Set sn76489 voice attenuation.
- void [setSN76489noise_attn](#) (uint8_t attenuate)
Set sn76489 noise attenuation.
- void [setSN76489noiseCtrl](#) (uint8_t type, uint8_t rate)
Set sn76489 noise type.

3.3.1 Function Documentation

3.3.1.1 getSN76489_FreqDiv()

```
uint16_t getSN76489_FreqDiv (
    uint32_t refClk,
    uint32_t voiceFreq )
```

Calculate frequency from hertz to binary value.

Parameters

<i>refClk</i>	is the reference clock in hertz for the sound chip.
<i>voiceFreq</i>	is the target frequency in hertz.

Returns

A unsigned 16 bit number that will result in the frequency wanted.

3.3.1.2 initSN76489()

```
void initSN76489 ( )
```

Initialize sn76489 to mute all channels.

3.3.1.3 setSN76489noise_attn()

```
void setSN76489noise_attn (
    uint8_t attenuate )
```

Set sn76489 noise attenuation.

Parameters

<i>attenuate</i>	: 8 = 16db, 4 = 8db, 2 = 4db, 1 = 2 db, 15 = Mute
------------------	---------------------------------------------------

3.3.1.4 setSN76489noiseCtrl()

```
void setSN76489noiseCtrl (
```

```
uint8_t type,  
uint8_t rate )
```

Set sn76489 noise type.

Parameters

<i>type</i>	: 0 = periodic, 1 = white.
<i>rate</i>	: 0 = N/512, 1 = N/2048, 2 = N/1024, 3 = voice 3

3.3.1.5 setSN76489voice_attn()

```
void setSN76489voice_attn (  
    uint8_t voice,  
    uint8_t attenuate )
```

Set sn76489 voice attenuation.

Parameters

<i>voice</i>	the voice (1,2, or 3) to set the attenuation to.
<i>attenuate</i>	: 8 = 16db, 4 = 8db, 2 = 4db, 1 = 2 db, 0 = 0 db?, 15 = Mute

3.3.1.6 setSN76489voice_freq()

```
void setSN76489voice_freq (  
    uint8_t voice,  
    uint16_t freqDiv )
```

Set sn76489 voice frequency.

Parameters

<i>voice</i>	the voice (1,2, or 3) to set the frequency to.
<i>freqDiv</i>	is binary number to set the frequency ($f = \text{refClk}/(32 * \text{freqDiv})$)

Index

getSN76489_FreqDiv
sn76489.h, [6](#)

initSN76489
sn76489.h, [6](#)

LICENSE.md, [5](#)

README.md, [5](#)

setSN76489noise_attn
sn76489.h, [6](#)

setSN76489noiseCtrl
sn76489.h, [6](#)

setSN76489voice_attn
sn76489.h, [7](#)

setSN76489voice_freq
sn76489.h, [7](#)

sn76489.h, [5](#)
getSN76489_FreqDiv, [6](#)
initSN76489, [6](#)
setSN76489noise_attn, [6](#)
setSN76489noiseCtrl, [6](#)
setSN76489voice_attn, [7](#)
setSN76489voice_freq, [7](#)