Driver for the WM8960 codec

This driver supports controlling a WM8960 codec. It is a literal Python translation of the C-Code provided by NXP/Freescale for their i.MX RT series of MCUs. Almost nothing was added, and just a few API related names were changed or added to cope with the naming style of MicroPython.

Features

The primary purpose of the driver is initialisation and setting operation modes of the codec. I does not do the audio data processing for the codec. That is the task of a separate driver built into the Operating system or firmware of a device.

Connection

The WM8960 supports next to the audio interface the I2C interface. The connection depends on the interface used and the number of devices in the system. For the I2C interface, SCL and SDA have to be connected, and of course GND and Vcc. The I2C default address is 0x1a.

Class

The driver contains the WM8960 class and quite a few name definitions.

```
wm8960=wm8960.WM8960(i2c, *,
    sample_rate=16000,
    bits=16,
    swap=wm8960_swap_none,
    route=wm8960_route_playback_record,
    enable_speaker=False,
    left_input=wm8960_input_differential_mic_input3,
    right_input=wm8960_input_differential_mic_input2,
    play_source=wm8960_play_source_DAC,
    master_clock_source=wm8960_sysclk_source_mclk,
    master_clock_freq=None,
    master_slave=False,
    adc_sync=wm8960_sync_dac,
    protocol=wm8960_bus_I2S,
    i2c_address=WM8960_I2C_ADDR
)
```

Only the first argument, i2c, is mandatory. All others are optional. Arguments:

- *i2c* The I2C bus object. It has be be created beforehand.
- *sample_rate* The audio sample rate. Acceptable values are 8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100, 48000, 96000, 192000 and 384000. Note, that not every I2S hardware will support all values.
- *bits* The number of bits per audio word. Acceptable value are 16, 20, 24, and 32.
- *swap* Swap the left & right channel, if set. For a list of options, see the table below.
- *route* Setting the audio path in the codec. For a list of options, see the table below.
- *enable_speaker* Enable or disable the speaker port.
- *left_input* Set the audio source for the left input channel. For a list of options, see the table below.
- *right_input* Set the audio source for the right input channel.For a list of options, see the table below.
- *play_source* Set the audio target for the output audio. For a list of options, see the table below.
- *master_clock_source* Control, whether the internal master clock called "sysclk" is directly taken from the MCLK input or derived from it using an internal PLL. It is usually not required to change that.
- *master_clock_freq* Argument for telling the frequency to by used. If not set, default values are used.
- *master_slave* Let the WM8960 act as Master of Slave device. The default setting is Slave. In slave mode, sample_rate and bits are controlled by the MCU.
- *adc_sync* Tell which input is used for the ADC sync signal. The default is using the DACLRC pin.
- *protocol* Setting the communication protocol. The default is I2S. For a list of options, see the table below.
- *i2c_address* The I2C address of the WM8960. The default is 0x1a or 26.

If master_clock_freq is not set, the following default values are used:

- master_clock_source == wm8960_sysclk_source_PLL: 11.2896 MHz for sample rates of 44100, 22050 and 11015 Hz, and 12.288 Mhz for sample rates < 48000, otherwise sample_rate * 256.
- master_clock_source == wm8960_sysclk_source_mclk: sample_rate * 256.

Tables of parameter constants

Swap Parameter

Value	Name
1	wm8960_swap_none
2	wm8960_swap_input
4	wm8960_swap_output

Route parameter:

Value	Name
0	wm8960_route_bypass
1	wm8960_route_playback
2	wm8960_route_playback_record
5	wm8960_route_record

Protocol Parameter

Value	Name
2	wm8960_bus_I2S
1	wm8960_bus_left_justified
0	wm8960_bus_right_justified
3	wm8960_bus_PCMA
19	wm8960_bus_PCMB

Input Source Parameter

Value	Name
0	wm8960_input_closed
1	wm8960_input_single_ended_mic
2	wm8960_input_differential_mic_input2
3	wm8960_input_differential_mic_input3
4	wm8960_input_line_input2_
5	wm8960_input_line_input3

Play Source Parameter

Value	Name
1	wm8960_play_source_PGA
2	wm8960_play_source_input
4	wm8960_play_source_DAC

Route Parameter

Value	Name
0	wm8960_route_bypass
1	wm8960_route_playback
2	wm8960_route_playback_record
5	wm8960_route_record

Master Clock Source Parameter

Value	Name
0	wm8960_sysclk_source_mclk
1	wm8960_sysclk_source_PLL

Module Names

Value	Name
0	wm8960_module_ADC
1	wm8960_module_DAC
2	wm8960_module_VREF
3	wm8960_module_headphone
4	wm8960_module_mic_bias
5	wm8960_module_mic
6	wm8960_module_line_in
7	wm8960_module_line_out
8	wm8960_module_speaker
9	wm8960_module_omix

Play Channel Names

Value	Name
1	wm8960_headphone_left
2	wm8960_headphone_right
4	wm8960_speaker_left
8	wm8960_speaker_right

adc_sync Parameters

Value	Name
0	wm8960_sync_adc
1	wm8960_sync_dac

Methods

Next to the initialisation, the driver provides some useful methods for controlling the operation:

set_left_input(input source)

Specify the source for the left input. The input source names are listed above.

set_right_input(input source)

Specify the source for the left input. For a list of suitable parameter values, see the table above.

set_volume(module, value [, value_r])

Sets the volume of a certain module. If two values are given, the first one is used for the left channel, the second for the right channel. For a list of suitable modules and highest values, see the table below.

Module Names and value ranges

Value Range	Name
0-255	wm8960_module_ADC
0-255	wm8960_module_DAC
0-127	wm8960_module_headphone
0-63	wm8960_module_line_in
0-127	wm8960_module_speaker

value = get_volume(module)

Get the actual volumes set for a module as a two element tuple. The module names are the same as for set_volume().

value = volume(module [, value [, value_r]])

Sets or get the volume of a certain module. If not value is supplied, the actual volume is returned. If two values are given, the first one is used for the left channel, the second for the right channel. For a list of suitable modules and highest values, see the table below.

mute(module, True|False)

Mute or unmute a certain module. The Module names are the same as for set_volume().

set_data_route(route)

Set the audio data route. For the parameter value/names, look at the table above.

set_module(module, True|False)

Enable or disable a module. For the list of module names, , look at the table above.

deinit()

Disable all modules.

Example Code

```
# Micro_python WM8960 Codec driver
#
# Setting the driver to Slave mode using the default settings
#
from machine import Pin, I2C
import wm8960

i2c = I2C(0)
wm=wm8960.WM8960(i2c)
```

```
# Micro_python WM8960 Codec driver
#
# Setting the driver to Master mode using specific audio format se
#
from machine import Pin, I2C
import wm8960

i2c = I2C(0)
wm=wm8960.WM8960(i2c, master_slave=True, sample_rate=16000, bits=3
```