

## ESP32FORTH COSINE WAVE GENERATOR

To quickly experiment with new ESP32forth Cosine wave generator words follow the instructions given below.

1. Replace the existing section of the ESP32forth INO file with the following code and recompile. I used Version v7.0.6.8 but other previous versions should also work.

```
#ifndef ENABLE_DAC_SUPPORT
# define OPTIONAL_DAC_SUPPORT
# else
# include <driver/dac.h>
# include <driver/dac_common.h>
# include <soc/rtc_io_reg.h>
# include <soc/rtc_cntl_reg.h>
# include <soc/sens_reg.h>
# include <soc/rtc.h>
# define OPTIONAL_DAC_SUPPORT \
Y(dac_output_enable, n0 = dac_output_enable( (dac_channel_t) n0 ) ) \
Y(dac_output_disable, n0 = dac_output_disable( (dac_channel_t) n0 ) ) \
Y(dac_output_voltage, n0 = dac_output_voltage((dac_channel_t) n1, (gpio_num_t) n0); NIP ) \
Y(dac_cw_generator_enable, PUSH dac_cw_generator_enable() ) \
Y(dac_cw_generator_disable, PUSH dac_cw_generator_disable() ) \
Y(dac_i2s_enable, PUSH dac_i2s_enable() ) \
Y(dac_i2s_disable, PUSH dac_i2s_disable() ) \
Y(rtc_freq_div_set, REG_SET_FIELD(RTC_CNTL_CLK_CONF_REG, RTC_CNTL_CLK8M_DIV_SEL, n0 ); DROP )
\ Y(dac_freq_step_set, SET_PERI_REG_BITS(SENS_SAR_DAC_CTRL1_REG, SENS_SW_FSTEP, n0,
SENS_SW_FSTEP_S); DROP ) \ Y(dac1_scale_set, SET_PERI_REG_BITS(SENS_SAR_DAC_CTRL2_REG,
SENS_DAC_SCALE1, 0, SENS_DAC_SCALE1_S); ) \ Y(dac2_scale_set,
SET_PERI_REG_BITS(SENS_SAR_DAC_CTRL2_REG, SENS_DAC_SCALE2, 0, SENS_DAC_SCALE2_S); ) \
Y(dac1_offset_set, SET_PERI_REG_BITS(SENS_SAR_DAC_CTRL2_REG, SENS_DAC_DC1, n0,
SENS_DAC_DC1_S); DROP ) \ Y(dac2_offset_set, SET_PERI_REG_BITS(SENS_SAR_DAC_CTRL2_REG,
SENS_DAC_DC2, n0, SENS_DAC_DC2_S); DROP ) \ Y(dac1_invert_set,
SET_PERI_REG_BITS(SENS_SAR_DAC_CTRL2_REG, SENS_DAC_INV1, n0, SENS_DAC_INV1_S); DROP ) \
Y(dac2_invert_set, SET_PERI_REG_BITS(SENS_SAR_DAC_CTRL2_REG, SENS_DAC_INV2, n0,
SENS_DAC_INV2_S); DROP ) \ Y(dac1_cosine_enable,
SET_PERI_REG_MASK(SENS_SAR_DAC_CTRL2_REG, SENS_DAC_CW_EN1_M); ) \ Y(dac2_cosine_enable,
SET_PERI_REG_MASK(SENS_SAR_DAC_CTRL2_REG, SENS_DAC_CW_EN2_M); ) \ Y(dacWrite,
```

```
dacWrite(n1, n0); DROPn(2))  
#endif
```

2. Connect pin 25 to a PC speaker in series with a 200-1Kohm resistor. The sound will be weak but otherwise the pin is overloaded and the signal is distorted.
3. Execute the following ESP32forth words to obtain a 476.5Hz sine wave tone totally generated by the ESP32 hardware.  
You will need an oscilloscope to confirm the sine wave shape. I included a picture for the curious ones!

```
1 rtc_freq_div_set \ Set the RTC frequency for no division.  
dac_cw_generator_enable \ Enable the CW generator.  
dac1_cosine_enable \ Connect CW to DAC channel 1 on pin 25.  
0 dac1_scale_set \ Set DAC channel 1 for full scale signal.  
2 dac1_invert_set \ Set the DAC channel 1 for no inversion.  
0 dac1_offset_set \ Set the DAC channel 1 for no DC offset.  
7 dac_freq_step_set \ Set DAC frequency step for a 476.5Hz signal.  
0 dac_output_enable \ Enable DAC channel 1 signal output to pin 25.
```



476.5Hz signal with a 1Kohm resistor instead of the PC speaker

If you want the whole beginners adventure with the ESP32 Cosine Wave generator go to this link  
<https://esp32.forth2020.org/projects-6>  
and download the PDF file [ ESP32FORTH COSINE WAVE GENERATOR 69 ] .

Have fun with ESP32forth!