Ledc-h Installation Guide

Clone the repository, and install the folders in the following way:

Place the folder FreeRTOS-Plus-CLI-vESP32 downloaded from this repository into your Arduino/libraries folder.

your directory structure should look like:

```
Arduino/
        libraries/
                FreeRTOS-Plus-CLI-vESP32/
                                FreeRTOS_CLI.c to FreeRTOS_CLI_vESP32.c
                                FreeRTOS_CLI.h to FreeRTOS_CLI_vESP32.h
```

In you Arduino/Sketches folder copy the ledc-h folder downloaded from the repository

```
Arduino/
        sketches/
                (your sketches)
                ledchv00/
                        ledchv00.ino
                        cli.ino
                        etc
```

Open and compile the Sketch to verify there are no errors.

Connect your ESP32 board and upload the sketch.

Open your Serial Monitor console (baud rate=115200), reset your ESP32 board again, to see the messages displayed

Startup message:

```
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:9720
ho 0 tail 12 room 4
load:0x40080400,len:6352
entry 0x400806b8

Queues created: xQSerialIn, xQSpooler



Tasks created:
$Spool, $Serial,
Tasks created:
$Spool, $Serial,
Commands registered:
help
gpiocfg, gpioset, gpiointdis, gpiointen
halledc, getpulsein
ledcchannelcfg, ledctimercfg, ledcfadeinstall, ledcfadeuninstall, ledcfadetime, ledcfadestart,
ledcupdateduty, ledcsetduty, ledcsetduty, ledcpause, ledcresume
hwmark, taskdelay

 ledc-h (ESP32 ledc 'protoboard companion')
 based on FreeRTOS+CLI(c) and esp-idf(c) API's
```

Type “help”

help (command echoed in the Serial Monitor)

```
help:
 Lists all the registered commands



gpiocfg (pin#)<..> (Mode)< D|I|O|U|X|B > (Pullup)< D|E > (Pulldown)< D|E >  (Interrupts) < D|P|
N|A|L|H >
configure a GPIO PIN
 Mode:
  D, DISABLE: disable input and output
  I, INPUT: input only
  O, OUTPUT: output only
  U, OUTPUT_OD: output only with open-drain
  X, INPUT_OUTPUT_OD: input/output open-drain
  B, INPUT_OUTPUT: input/output
 Interrupt type:
  D, DISABLE
  P, POSEDGE: rising edge
  N, NEGEDGE: falling edge
  A, ANYEDGE: rising and falling edge
  L, LOW_LEVEL: input low level
  H, HIGH_LEVEL: input hig
gpioset (pin#)<..> (level) < 0|1 >
set PIN level ( 0 | 1 )
ex: gpioset 14 1

gpiointdis (pin#)<..>
GPIO Interrupt Disable
example: gpiointdis 14

gpiointen (pin#)<..>
GPIO Interrupt Enable
example: gpiointen 14

halledc < S(etup)| W(rite)| T(one)| R(eadFreq)| A(ttachPin) | D(etachPin) > <additional
parameter(s)for each command>
hal-ledc functions to generate PWM signals
 S(etup): (channel)<'0-7'> (freq)<double> (resolution_bits)<uint8_t>
 W(rite): (channel)<'0-7'> (duty)<uint32_t >
 T(one): (channel)<'0-7'> (freq)<double>
 R(ead): (channel)<'0-7'>
 A(ttachPin): (channel)<'0-7'> (PIN)<uint8_t>
 D(etachPin): (PIN)<uint8_t>
 example:
 halledc S 1 4186 8

getpulsein (PIN)<uint8_t>
Arduino PulseIn function, returns HIGH and LOW timings, estimated frequency and duty cycle
a continuous pulse stream is expected
 ex: getpulsein 14

ledcchcfg (pin)< > (speed mode)<H|L> (channel)<0 - 7> (intr_type)< D|E > (timer)<0 - 3>
(duty)< ... > (hpoint)< ... >
Configure LEDC channel
speed_mode:
-H, HIGH_SPEED_MODE
-L, LOW_SPEED_MODE
intr_type:
-D, INTR_DISABLE
-E, INTR_FADE_END
duty: [0, (2**duty_resolution)]
hpoint: max value is 0xfffff
ex: ledcchcfg 2 H 0 D 0 0 0

ledctimercfg (speed mode)<H|L> (bits duty resolution)< ... > (timer)<0 - 3> (frequency Hz)< ... >
Configure LEDC timer
duty resolution: the range of duty setting is [0, (2**duty_resolution)]
```

ex: ledctimercfg H 13 0 5000
return
- ESP_OK Success
- ESP_ERR_INVALID_ARG Parameter error
- ESP_FAIL Can not find a proper pre-divider number base on the given frequency and the current duty_resolution.

ledcfadeinstall < (int)intr_alloc_flags >
Install LEDC fade function.
parameter intr_alloc_flags: ESP_INTR_FLAG_* values. See esp_intr_alloc.h for more info.
ex: ledcfadeinstall 0
return
- ESP_OK Success
- ESP_ERR_INVALID_STATE Fade function already installed.

ledcfadeuninstall
Uninstall LEDC fade function

ledcfadetime (speed mode)<H|L> (channel)<0 - 7> (target_duty)<..> (fade_time_ms)<..>
Set LEDC fade function, with a limited time
target_duty of fading.<0 - (2**duty_resolution-1)>
max_fade_time_ms: maximum time of the fading (ms)
ex: ledcfadetime H 0 4000 3000
return
- ESP_ERR_INVALID_ARG Parameter error
- ESP_OK Success
- ESP_ERR_INVALID_STATE Fade function not installed.
- ESP_FAIL Fade function init error

ledcfadestart (speed mode)<H|L> (channel)<0 - 7> (fade_mode) < N | W >
Start LEDC fading.
speed_mode:
-H, HIGH_SPEED_MODE
-L, LOW_SPEED_MODE
fade_mode
-N, FADE_NO_WAIT
-W, FADE_WAIT_DONE will block until fading to the target duty
ex: ledcfadestart H 0 N
return
- ESP_OK Success
- ESP_ERR_INVALID_STATE Fade function not installed.
- ESP_ERR_INVALID_ARG Parameter error

ledcupdateduty (speed mode)<H|L> (channel)<0 - 7>
LEDC update channel parameters.
Call this function to activate the LEDC updated parameters.
speed_mode:
-H, HIGH_SPEED_MODE
-L, LOW_SPEED_MODE
ex: ledcupdateduty L 0
return
- ESP_OK Success

- ESP_ERR_INVALID_ARG Parameter error

ledcsetduty (speed mode)<H|L> (channel)<0 - 7> (duty)<....>
LEDC set duty.
only after calling ledc_update_duty will the duty update.
speed_mode:
-H, HIGH_SPEED_MODE
-L, LOW_SPEED_MODE
ex: ledcupdateduty L 0 128
return
- ESP_OK Success
- ESP_ERR_INVALID_ARG Parameter error

ledcstop (speed mode)<H|L> (channel)<0 - 7> (idle_level)<....>
LEDC stop.
Disable LEDC output, and set idle level.
speed_mode:
-H, HIGH_SPEED_MODE
-L, LOW_SPEED_MODE
ex: ledcstop H 0 0
return
- ESP_OK Success
- ESP_ERR_INVALID_ARG Parameter error

ledcpause (speed mode)<H|L> (timer)<0 - 3>
Pause LEDC timer counter.
speed_mode:
-H, HIGH_SPEED_MODE
-L, LOW_SPEED_MODE
ex: ledcpause L 0
return
- ESP_OK Success
- ESP_ERR_INVALID_ARG Parameter error

ledcresume (speed mode)<H|L> (timer)<0 - 3>
Resume LEDC timer counter.
speed_mode:
-H, HIGH_SPEED_MODE
-L, LOW_SPEED_MODE
ex: ledcresume L 0
return
- ESP_OK Success
- ESP_ERR_INVALID_ARG Parameter error

hwmark
returns the minimum amount of remaining stack space that was available to the tasks since the tasks
started executing – that is the amount of stack that remained unused when the task stack was at its greatest (deepest) value.
This is what is referred to as the stack ‘high water mark’.
example: hwmark

taskdelay (delay_ms)<..>
executes RTOS vTaskDelay function for the specified time interval in microsec.
example: taskdelay 1000

Session example:

```
hwmark

 unused stack when the task stack was at its greatest (deepest) value
$Spool: 900
$Serial: 1624
```

“fade” example.

The following commands show how to reproduce the fade example of the peripherals/ledc_example.c program (esp-idf examples), with 5 commands

commands could be entered individually

```
ledctimercfg H 13 0 5000
ledcchcfg 2 H 0 D 0 0 0
ledcfadeinstall 0
ledcfadetime H 0 4000 6000
ledcfadestart H 0 N
```

or “chained” in one line using “;”

```
ledctimercfg H 13 0 5000 ; ledcchcfg 2 H 0 D 0 0 0 ; ledcfadeinstall 0 ; ledcfadetime H 0 4000 6000 ; ledcfadestart H 0 N ;
```

return

```
 ESP_OK
 ledc_timer_config(speed_mode 0,duty_resolution 13,timer_num 0,freq_hz 5000)

 ESP_OK
 ledc_channel_config(gpio 2, speed_mode 0, channel 0, int_type 0,timer 0, duty 0, hpoint 0 )

 ESP_OK ledc_fade_func_install( 0 )

 ESP_OK
 ledc_set_fade_with_time(speed_mode 0, channel 0, target_duty 4000, fade_time 6000)

 ESP_OK
 ledc_fade_start(speed_mode  0, channel  0, fade_mode  0)
```

even 2 different exercises can be chained together:

```
ledctimercfg H 13 0 5000 ; ledcchcfg 2 H 0 D 0 0 0 ; ledcfadeinstall 0 ; ledcfadetime H 0 4000 6000 ; ledcfadestart H 0 W ; ledctimercfg H 13 0 5000 ; ledcchcfg 2 H 0 D 0 0 0 ; ledcfadeinstall 0 ; ledcfadetime H 0 4000 6000 ; ledcfadestart H 0 N
```

the first part ends with the ledcfadestart H 0 W command. It stops there (the ‘W’ parameter) to wait for the fade operation to finish. If ‘No Wait’ is specified, the second fade operation does not take place.

```
ESP_OK
 ledc_timer_config(speed_mode 0,duty_resolution 13,timer_num 0,freq_hz 5000)

 ESP_OK
 ledc_channel_config(gpio 2, speed_mode 0, channel 0, int_type 0,timer 0, duty 0, hpoint 0 )

 ESP_OK ledc_fade_func_install( 0 )

 ESP_OK
 ledc_set_fade_with_time(speed_mode 0, channel 0, target_duty 4000, fade_time 6000)

 ESP_OK
 ledc_fade_start(speed_mode  0, channel  0, fade_mode  0)
ledctimercfg H 13 0 5000
ledcchcfg 2 H 0 D 0 0 0
ledcfadeinstall 0
ledcfadetime H 0 4000 6000

 ESP_OK
 ledc_timer_config(speed_mode 0,duty_resolution 13,timer_num 0,freq_hz 5000)

ledcfadestart H 0 W ←-- first fade operation ends here

ledctimercfg H 13 0 5000
ledcchcfg 2 H 0 D 0 0 0
ledcfadeinstall 0
ledcfadetime H 0 4000 6000
ledcfadestart H 0 N

 ESP_OK
 ledc_channel_config(gpio 2, speed_mode 0, channel 0, int_type 0,timer 0, duty 0, hpoint 0 )

 ESP_ERR_NOT_FOUND ledc_fade_func_install( 0 )

 ESP_OK
 ledc_set_fade_with_time(speed_mode 0, channel 0, target_duty 4000, fade_time 6000)

 ESP_OK
 ledc_fade_start(speed_mode  0, channel  0, fade_mode  1)

 ESP_OK
 ledc_timer_config(speed_mode 0,duty_resolution 13,timer_num 0,freq_hz 5000)

 ESP_OK
 ledc_channel_config(gpio 2, speed_mode 0, channel 0, int_type 0,timer 0, duty 0, hpoint 0 )

 ESP_ERR_NOT_FOUND ledc_fade_func_install( 0 )

 ESP_OK
 ledc_set_fade_with_time(speed_mode 0, channel 0, target_duty 4000, fade_time 6000)

 ESP_OK
 ledc_fade_start(speed_mode  0, channel  0, fade_mode  0)
```

hal-ledc and pulseIn session example:

```
halledc S 0 1000 8 ; halledc W 0 128 ; halledc A 0 2

gpiocfg 2 B D D D ; halledc S 0 1000 8 ; halledc W 0 128 ; halledc A 0 2 ; getpulsein 2
 ESP_OK gpio_config(pin 2,Mode 3,Pup 0,Pdn 0,Int 0 )

 Channel 0 frequency set to 1000.000000 Hz


 Channel 0 duty cycle changed to 128


 PIN 2 attached to Channel 0


Measured PulseIn (microsec): THigh 500  TLow 499 T 999.000000
 frequency: 1001.001001(Hz)  duty 0.500501

 halledc W 0 64; getpulsein 2
halledc W 0 64
getpulsein 2

 Channel 0 duty cycle changed to 64


Measured PulseIn (microsec): THigh 250  TLow 749 T 999.000000
 frequency: 1001.001001(Hz)  duty 0.250250

 halledc S 0 10000 8 ; halledc W 0 128 ; halledc A 0 2 ; getpulsein 2
halledc S 0 10000 8
halledc W 0 128
halledc A 0 2
getpulsein 2

 Channel 0 frequency set to 10000.000000 Hz


 Channel 0 duty cycle changed to 128


 PIN 2 attached to Channel 0


Measured PulseIn (microsec): THigh 50  TLow 50 T 100.000000
 frequency: 10000.000000(Hz)  duty 0.500000

 halledc S 0 20000 8 ; halledc W 0 128 ; halledc A 0 2 ; getpulsein 2
halledc S 0 20000 8
halledc W 0 128
halledc A 0 2
getpulsein 2

 Channel 0 frequency set to 20000.000000 Hz


 Channel 0 duty cycle changed to 128


 PIN 2 attached to Channel 0


Measured PulseIn (microsec): THigh 25  TLow 24 T 49.000000
 frequency: 20408.163265(Hz)  duty 0.510204

 halledc S 0 2000 8 ; halledc W 0 128 ; halledc A 0 2 ; getpulsein 2
halledc S 0 2000 8
halledc W 0 128
halledc A 0 2
getpulsein 2

 Channel 0 frequency set to 2000.000000 Hz


 Channel 0 duty cycle changed to 128


 PIN 2 attached to Channel 0


Measured PulseIn (microsec): THigh 250  TLow 250 T 500.000000
 frequency: 2000.000000(Hz)  duty 0.500000

halledc W 0 64; getpulsein 2
halledc W 0 64
getpulsein 2

 Channel 0 duty cycle changed to 64


Measured PulseIn (microsec): THigh 124  TLow 375 T 499.000000
 frequency: 2004.008016(Hz)  duty 0.248497
```

Note1:
the precision of the pulseIn measurement decreases with the frequency. It is intended to be used as a “quick” checkout, not as a precise measurment