mipea Documentation

Release 2.0.0

jasLogic

Jan 02, 2020

Contents

CHAPTER 1

Installation

The source code is hosted on GitHub. mipea uses autotools (autoconf, automake, libtool) to build and install the library. The git repository does *not* include the `configure` script and `Makefile.in` which means that you have two options for installing the library.

1.1 Download the configure script

On GitHub, when you look under the tab releases you will find some release with a name like for example "v2.0.0". Then you can download the corresponding file named "mipea_x.x.x" which *includes the configure script and Makefile.in*. After downloading just run these commands from inside the downloaded directory:

```
$ ./configure
$ make
$ sudo make install
```

1.2 Building the configure script

When you have the GNU autotools installed you can simply clone this repository and build the `configure` script and `Makefile.in` yourself:

```
$ git clone https://github.com/jasLogic/mipea.git
$ cd mipea
$ autoreconf --install
$ ./configure
$ make
$ sudo make install
```

1.3 Troubleshooting

1.3.1 Configure script can not find /proc/cpuinfo

If the `configure` script prints this warning:

```
configure: WARNING: cannot find file /proc/cpuinfo
```

than the script was unable to find the `cpuinfo` file which is needed to determine the SoC (BCM2835 or BCM2836/7) and the revision. Pis with a revision number where the last four digits are less than `0004` use I2C bus 0 instead of 1, like the new ones.

This error can be fixed by editing the `config.h` file ensuring that it contains these lines (depending on your Pi):

```
#define BCM2835         1
#define BCM2836_7       1
#define USE_I2C_BUS_0   1
```

1.3.2 When running a program, the shared library file is not found

I noticed that sometimes the library can be linked, but when running a program an error message appears saying: `File or directory not found`. If you havethis problem just run `ldconfig` or follow the output from `sudo make install`:

```
----------------------------------------------------------------------
Libraries have been installed in:
   /usr/local/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the `-LLIBDIR'
flag during linking and do at least one of the following:
   - add LIBDIR to the `LD_LIBRARY_PATH' environment variable
     during execution
   - add LIBDIR to the `LD_RUN_PATH' environment variable
     during linking
   - use the `-Wl,-rpath -Wl,LIBDIR' linker flag
   - have your system administrator add LIBDIR to `/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
----------------------------------------------------------------------
```

1.3.3 Wifi stops working when using the library

In versions 2.1.0 and below the GPIO map and unmap functions cleared *all* pullup / -downs on all pins. This could lead to the wifi not working until a reboot. This issue should be fixed with version 2.1.1.

CHAPTER 2

Peripherals

The peripheral functions are something like the *core* of the library. They map and unmap the memory used by all other parts.

2.1 Macros

PERIPHERAL_BASE_BCM2835

```
0x20000000
```

This macro holds the value of the peripheral base, when a BCM2835 is used.

PERIPHERAL_BASE_BCM2836_7

```
0x3F000000
```

This macro holds the value of the peripheral base, when a BCM2836 or BCM2837 is used.

2.2 Functions

int **peripheral_map** (volatile uint32_t **_map_, uint32_t _offset_, uint32_t _size_)
This function maps a code memory block of size `size` at offset `offset` from the peripheral base.

> **Note:** The `offset` must be a multiple of the page size which is `4096` on the Raspberry Pi.

The function returns 0 on success and `-1` on error.

void **peripheral_unmap** (void* _map_, uint32_t _size_)
This function unmaps the memory mapped to pointer `map` with size `size`.

int **peripheral_ismapped**(void **map*, uint32_t *size*)

This function checks if a pointer `map` is already mapped to a memory region with the size `size`. It returns `true` if the pointer is already mapped and `false` if not.

CHAPTER 3

Clock Manager

3.1 Registers

struct **clock_manager_register_map**

This struct maps the registers of the clock manager. The names of the struct members correspond to the registers. Unfortunately, the official datasheet does not feature this chapter. But there is an upload of this chapter here: BCM2835 clocks:

```
struct clock_manager_register_map {
    uint32_t GP0CTL;
    uint32_t GP0DIV;
    uint32_t GP1CTL;
    uint32_t GP1DIV;
    uint32_t GP2CTL;
    uint32_t GP2DIV;
    uint32_t: 32;
    uint32_t: 32;
    uint32_t: 32;
    uint32_t: 32;
    uint32_t PCMCTL;
    uint32_t PCMDIV;
    uint32_t PWMCTL;
    uint32_t PWMDIV;
}
```

extern volatile struct *clock_manager_register_map* ***CM**

```
CM = (volatile struct clock_manager_register_map *)(clock_manager_base_ptr + 28);
```

By using this variable, the registers of the clock manager can be accessed like this `CM->PWMCTL`.

3.2 Enums

3.2.1 Clock sources

This enum holds the values for the different clock sources:

```
enum {
    CLOCK_GND,
    CLOCK_OSC,
    CLOCK_TST0,
    CLOCK_TST1,
    CLOCK_PLLA,
    CLOCK_PLLC,
    CLOCK_PLLD,
    CLOCK_HDMI
};
```

3.3 Global Variables

extern const uint32_t CM_PASSWD;

```
const uint32_t CM_PASSWD = 0x5A000000;
```

This variable holds the clock manager password. This value must always be present when writing to a clock manager register (e.g. by OR with the value).

3.4 Functions

int **clock_map**(void)
: This function maps the clock manager registers. It calls *peripheral_map()* with the values `CLOCK_MANAGER_OFFSET` and `CLOCK_MANAGER_SIZE`. On error `-1` is returned.

void **clock_unmap**(void)
: This function unmaps the clock manager.

The following functions all take a pointer to a clock manager register as an argument because all the registers for the *different clocks* have the *same structure*. This means that you just need to tell the clock manager which clock to use (by pointing to the right register). For example: `clock_enable(&CM->PWMCTL);`

void **clock_enable**(volatile uint32_t **reg*)
: This function enables the clock with the register pointed to by `reg`.

void **clock_disable**(volatile uint32_t **reg*)
: This function disables the clock with the register pointed to by `reg`.

void **clock_configure**(volatile uint32_t **reg*, clock_source_t *src*, unsigned int *divisor*, unsigned int *mash*)
: This function configures the clock with the register pointed to by `reg` and sets up the `clock_source_t src`, the divisor `divisor` with the mash factor `mash`.

Todo: Add a decimal places to the divisor.

CHAPTER 4

GPIOs

4.1 Registers

struct **gpio_register_map**

This struct maps the registers of the GPIOs. The names of the struct members correspond to the registers from the Datasheet:

```
struct gpio_register_map {
        uint32_t FSEL[6];
        uint32_t: 32;
        uint32_t SET[2];
        uint32_t: 32;
        uint32_t CLR[2];
        uint32_t: 32;
        uint32_t LEV[2];
        uint32_t: 32;
        uint32_t EDS[2];
        uint32_t: 32;
        uint32_t REN[2];
        uint32_t: 32;
        uint32_t FEN[2];
        uint32_t: 32;
        uint32_t HEN[2];
        uint32_t: 32;
        uint32_t LEN[2];
        uint32_t: 32;
        uint32_t AREN[2];
        uint32_t: 32;
        uint32_t AFEN[2];
        uint32_t: 32;
        uint32_t PUD;
        uint32_t PUDCLK[2];
// BCM2711 only
        uint32_t: 32;
```

(continues on next page)

(continued from previous page)

```
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t: 32;
        uint32_t PUPPDN[4];
};
```

extern volatile struct *gpio_register_map* ***GP**

```
GP = (volatile struct gpio_register_map *)gpio_base_ptr;
```

By using this struct, the registers of the GPIOs can be accessed like this `GP->SET[0]`.

4.2 Enums

4.2.1 Pin functions

This enum holds the values for the various pin functions:

```
enum {
        INPUT, OUTPUT, ALT0, ALT1, ALT2, ALT3, ALT4, ALT5
};
```

4.2.2 Pullup / -downs

This enum holds the values for the states of the pullups / -downs:

```
enum {
        PUD_DISABLE, PUD_DOWN, PUD_UP
};
```

4.3 Functions

int **gpio_map**(void)

This function maps the GPIO registers. It calls *peripheral_map()* with the values `GPIO_OFFSET` and `GPIO_SIZE`. On error `-1` is returned.

void **gpio_unmap**(void)
This function unmaps the GPIOs.

void **gpio_func**(uint32_t *pin*, int *function*)
This function sets the pin `pin` to the pin function `function`.

void **gpio_set**(uint32_t *pin*)
Set the pin `pin`.

void **gpio_clr**(uint32_t *pin*)
Clear the pin `pin`.

uint32_t **gpio_tst**(uint32_t *pin*)
Test the pin `pin`. This function returns `0` or `false` when the pin is low and non-zero if the pin is high.

void **gpio_pud**(uint32_t *pin*, int *pud*)
Use the pullup / -down functionality `pud` on the pin `pin`.

void **gpio_inp**(uint32_t *pin*)
Make pin `pin` an input.

void **gpio_out**(uint32_t *pin*)
Make pin `pin` an output.

CHAPTER 5

I2C

5.1 Registers

struct **i2c_register_map**
This struct maps the registers of the BSC controller. The names of the struct members correspond to the registers from the Datasheet:

```
struct i2c_register_map {
    uint32_t C;
    uint32_t S;
    uint32_t DLEN;
    uint32_t A;
    uint32_t FIFO;
    uint32_t DIV;
    uint32_t DEL;
    uint32_t CLKT;
};
```

extern volatile struct *i2c_register_map* ***I2C**

```
I2C = (volatile struct i2c_register_map *)i2c_base_ptr;
```

By using this variable, the registers of the I2C can be accessed like this `I2C->FIFO`.

5.2 Functions

int **i2c_map**(void)
This function maps the I2C registers. It calls *peripheral_map()* with the values `I2C_OFFSET` and `I2C_SIZE`. `I2C_OFFSET` is defined in `i2c.c`. On error `-1` is returned.

void **i2c_unmap**(void)
This function unmaps the I2C registers.

void **i2c_set_address**(uint8_t *addr*)
This function sets the address of the I2C device to communicate with. The address is a seven bit value.

void **i2c_set_clkdiv**(uint16_t *divisor*)
This function sets the clock divisor of the BSC controller.

Note: The clock source is the core clock with a frequency, according to the Datasheet, of `150 MHz` and according to this file and other sources of `250 MHz`. When I tested the clock speed of I2C and SPI with a logic analyzer, it seems that `250 MHz` **is correct** (at least for the Raspberry Pi Zero I use).

---

void **i2c_set_clkstr**(uint16_t *clkstr*)
This function sets the clock stretch timeout (or delay). This means that the master will wait `clkstr` cycles after the rising clock edge for the slave to respond. After this the timeout flag is set. This can often be left at reset value `0x40`.

void **i2c_start**(void)
Starts the BSC controller and clears the flag register.

void **i2c_stop**(void)
Disables the BSC controller.

void **i2c_write_byte**(uint8_t *byte*)
Write a `byte` of data.

uint8_t **i2c_read_byte**(void)
This function receives a `byte` of data and returns it.

void **i2c_write_data**(const uint8_t **data*, uint16_t *length*)
This function writes `length` bytes of data pointed to by `data`.

void **i2c_read_data**(uint8_t **data*, uint16_t *length*)
This function receives `length` bytes of data and writes them to the array `data`.

void **i2c_write_register**(uint8_t *reg*, uint8_t *data*)
This function writes to bytes of data. First `reg` and then `data`.

Note: You *cannot* use two calls to *i2c_write_byte()* instead of this function because this is only *one* transmission, while two times *i2c_write_byte()* would be *two* different transmissons.

---

uint8_t **i2c_read_register**(uint8_t *reg*)
In contrast to *i2c_write_register()* you *can* use a call to *i2c_write_byte()* and to *i2c_read_byte()*. This is because I2C needs to make two transmissions anyway to change the read / write bit.

5.2.1 Useful Values

`I2C_FIFO_SIZE`	The size of the I2C FIFO
`I2C_C_I2CEN`	Enable I2C
`I2C_C_ST`	Start transfer
`I2C_C_CLEAR`	Clear the FIFO
`I2C_C_READ`	This transfer read from the slave
`I2C_S_RXS`	FIFO can be read
`I2C_S_TXD`	FIFO is full
`I2C_S_DONE`	Transfer done

CHAPTER 6

PWM

Note: The Datasheet specifies PWM channels 0 and 1. The Raspberry Pi has pins for PWM channels 1 and 2, you just need to add one.

6.1 Macros

RNG_CHANNEL0

DAT_CHANNEL0

RNG_CHANNEL1

DAT_CHANNEL1

```
#define RNG_CHANNEL0    PWM->RNG1
#define DAT_CHANNEL0    PWM->DAT1
#define RNG_CHANNEL1    PWM->RNG2
#define DAT_CHANNEL1    PWM->DAT2
```

To prevent confusion (because the Datasheet calls the PWM channels 1 and 2 and the Raspberry Pi 0 and 1) the values of the registers which need to be used “on the fly” are :code;‘defined‘ from 2 to 1 and from 1 to 0.

6.2 Registers

struct **pwm_register_map**

This struct maps the registers of the PWM. The names of the struct members correspond to the registers from the Datasheet:

```
struct pwm_register_map {
    uint32_t CTL;
    uint32_t STA;
    uint32_t DMAC;
    uint32_t: 32;
    uint32_t RNG1;
    uint32_t DAT1;
    uint32_t FIF1;
    uint32_t: 32;
    uint32_t RNG2;
    uint32_t DAT2;
};
```

extern volatile struct *pwm_register_map* ***PWM**

```
PWM = (volatile struct pwm_register_map *)pwm_base_ptr;
```

By using this variable, the registers of the PWM can be accessed like this `PWM->RNG1`.

6.3 Enums

6.3.1 PWM channel number

This enum holds the values distinguishing PWM channel 0 and 1:

```
enum {
    PWM_CHANNEL0, PWM_CHANNEL1
};
```

6.4 Structs

pwm_channel_config

This struct is used to configure a PWM channel:

```
typedef struct {
    union {
        struct {
            uint32_t: 1;
            uint32_t mode: 1;
            uint32_t rptl: 1;
            uint32_t sbit: 1;
            uint32_t pola: 1;
            uint32_t usef: 1;
            uint32_t: 1;
            uint32_t msen: 1;
        };
        uint32_t ctl_register;
    };
    unsigned int divisor;
    uint32_t range;
} pwm_channel_config;
```

uint32_t **ctl_register**
: This member can be directly edited by the anonymous struct inside this union. This register maps directly to the `CTL` register, with some offset for PWM 1. The settings of this register are described in the *Macros*.

unsigned int **divisor**
: The divisor which is passed to the *Clock Manager*.

uint32_t **range**
: The range to which the PWM counter counts before it starts over.

6.5 Functions

int **pwm_map** (void)
: This function maps the PWM registers. It calls *peripheral_map()* with the values `PWM_OFFSET` and `PWM_SIZE`. On error `-1` is returned.

void **pwm_unmap** (void)
: This function unmaps the PWM registers.

void **pwm_configure** (int *channel*, *pwm_channel_config* *config)
: This function configures `channel` with a *pwm_channel_config* pointed to by `config`.

void **pwm_enable** (int *channel*)
: This function enables `channel`.

void **pwm_disable** (int *channel*)
: This function disables `channnel`.

6.5.1 Configuration Values

`PWM_CTL_MODE_PWM`	Use PWM mode
`PWM_CTL_MODE_SERIALISER`	Use serialiser mode
`PWM_RPTL_STOP`	If serialiser mode: Transmission stops when fifo empty
`PWM_RPTL_REPEAT`	If serialiser mode: Repeat last data when fifo empty
`PWM_SBIT_LOW`	Output low when no transmission active
`PWM_SBIT_HIGH`	Output high when no transmission active
`PWM_POLA_DEFAULT`	Polarity is default
`PWM_POLA_INVERTED`	Polarity is innverted
`PWM_USEF_DATA`	Data register is transmitted
`PWM_USEF_FIFO`	Data from fifo is transmitted
`PWM_MSEN_PWMALGORITHM`	Use PWM algorithm
`PWM_MSEN_MSRATIO`	Use MS ratio

CHAPTER 7

SPI

7.1 Registers

struct **spi_register_map**

This struct maps the registers of the SPI. The names of the struct members correspond to the registers from the Datasheet:

```
struct spi_register_map {
    uint32_t CS;
    uint32_t FIFO;
    uint32_t CLK;
    uint32_t DLEN;
    uint32_t LTOH;
    uint32_t DC;
};
```

extern volatile struct *spi_register_map* ***SPI**

```
SPI = (volatile struct spi_register_map *)spi_base_ptr;
```

By using this variable, the registers of the SPI can be accessed like this `SPI->CS`.

7.2 Structs

spi_channel_config

This struct is used to configure SPI:

```
typedef struct {
    union {
        struct {
```

(continues on next page)

(continued from previous page)

```
            uint32_t: 2;
            uint32_t cpha: 1;
            uint32_t cpol: 1;
            uint32_t: 2;
            uint32_t cspol: 1;
            uint32_t: 14;
            uint32_t cspol0: 1;
            uint32_t cspol1: 1;
            uint32_t cspol2: 1;
        };
        uint32_t cs_register;
    };

    uint16_t divisor;
} spi_channel_config;
```

uint32_t **cs_register**
: This member can be directly edited by the anonymous struct inside this union. This register maps directly to the `CS` register. The settings of this register are described in the **'Macros'_**.

uint16_t **divisor**
: The master clock divisor.

Note: The clock source is the core clock with a frequency, according to the Datasheet, of `150 MHz` and according to this file and other sources of `250 MHz`. When I tested the clock speed of I2C and SPI with a logic analyzer, it seems that `250 MHz` **is correct** (at least for the Raspberry Pi Zero I use).

7.3 Functions

int **spi_map**(void)
: This function maps the SPI registers. It calls *`peripheral_map()`* with the values `SPI_OFFSET` and `SPI_SIZE`. On error `-1` is returned.

void **spi_unmap**(void)
: This function unmaps the SPI registers.

void **spi_configure**(*spi_channel_config *config*)
: This function configures SPI with a *`spi_channel_config`* pointed to by `config`.

void **spi_set_ce**(uint8_t *ce*)
: This function sets which chip enable line the SPI controller should use. This can be a 3 bit value.

void **spi_transfer_start**(void)
: This function starts a SPI transfer.

void **spi_transfer_stop**(void)
: This function stops the current SPI transfer.

uint8_t **spi_transfer_byte**(uint8_t *data*)
: While there is a SPI transfer active you can call this function as often as needed by the slave, to send and receive. This function needs to be called between *`spi_transfer_start()`* and *`spi_transfer_stop()`*, it sends `data` over SPI and asynchronously receives data and *returns* it.

uint8_t **spi_send2_recv1** (uint8_t *data0*, uint8_t *data1*)

This function writes to bytes of data and than keeps the clock running to receive and return the third byte. *spi_transfer_start()* and *spi_transfer_stop()* *may not* be called when using this function.

7.3.1 CS Register Bit Values

SPI_CS_CE0	Use chip enable 0
SPI_CS_CE1	Use chip enable 1
SPI_CS_CE2	Use chip enable 2
SPI_CPHA_CLK_BEGINNING	Data on clock leading edge
SPI_CPHA_CLK_MIDDLE	Data on clock trailing edge
SPI_CPOL_RESET_LOW	Clock polarity: active low
SPI_CPOL_RESET_HIGH	Clock polarity: active high
SPI_CSPOL_ACTIVE_LOW	Chip enable: active low
SPI_CSPOL_ACTIVE_HIGH	Chip enable: active high

CHAPTER 8

Timer

8.1 Registers

struct **timer_register_map**

This struct maps the registers of the timer. The names of the struct members correspond to the registers from the Datasheet:

```
struct i2c_register_map {
    uint32_t CS;
    uint32_t CLO;
    uint32_t CHI;
    uint32_t C0;
    uint32_t C1;
    uint32_t C2;
    uint32_t C3;
};
```

extern volatile struct *timer_register_map* ***TMR**

```
TMR = (volatile struct timer_register_map *)timer_base_ptr;
```

By using this variable, the registers of the timer can be accessed like this `TMR->CLO`.

8.2 Functions

int **timer_map**(void)

This function maps the timer registers. It calls *peripheral_map()* with the values `TIMER_OFFSET` and `TIMER_SIZE`. On error `-1` is returned.

void **timer_unmap**(void)

This function unmaps the timer registers.

void timer_read(uint64_t *counter);

This function reads the value of the timer into the 64-bit varbiale pointed to by `counter`.

CHAPTER 9

Mipea Wrapper

The mipea.c / h files are just a wrapper for all the other parts of the library. If you are lazy (or need all peripherals mapped) than this wrapper is usefull.

9.1 Functions

int **mipea_map** (void)
: This function maps all the peripherals and returns `-1` on error.

void **mipea_unmap** (void)
: This function unmaps all the peripherals.

CHAPTER 10

License

```
Copyright (C) 2018 Jaslo Ziska
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
   list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

CHAPTER 11

Documentation License

```
Attribution-ShareAlike 4.0 International

=======================================================================

Creative Commons Corporation ("Creative Commons") is not a law firm and
does not provide legal services or legal advice. Distribution of
Creative Commons public licenses does not create a lawyer-client or
other relationship. Creative Commons makes its licenses and related
information available on an "as-is" basis. Creative Commons gives no
warranties regarding its licenses, any material licensed under their
terms and conditions, or any related information. Creative Commons
disclaims all liability for damages resulting from their use to the
fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and
conditions that creators and other rights holders may use to share
original works of authorship and other material subject to copyright
and certain other rights specified in the public license below. The
following considerations are for informational purposes only, are not
exhaustive, and do not form part of our licenses.

     Considerations for licensors: Our public licenses are
     intended for use by those authorized to give the public
     permission to use material in ways otherwise restricted by
     copyright and certain other rights. Our licenses are
     irrevocable. Licensors should read and understand the terms
     and conditions of the license they choose before applying it.
     Licensors should also secure all rights necessary before
     applying our licenses so that the public can reuse the
     material as expected. Licensors should clearly mark any
     material not subject to the license. This includes other CC-
     licensed material, or material used under an exception or
```

(continues on next page)

(continued from previous page)

```
    limitation to copyright. More considerations for licensors:
       wiki.creativecommons.org/Considerations_for_licensors

    Considerations for the public: By using one of our public
    licenses, a licensor grants the public permission to use the
    licensed material under specified terms and conditions. If
    the licensor's permission is not necessary for any reason--for
    example, because of any applicable exception or limitation to
    copyright--then that use is not regulated by the license. Our
    licenses grant only permissions under copyright and certain
    other rights that a licensor has authority to grant. Use of
    the licensed material may still be restricted for other
    reasons, including because others have copyright or other
    rights in the material. A licensor may make special requests,
    such as asking that all changes be marked or described.
    Although not required by our licenses, you are encouraged to
    respect those requests where reasonable. More_considerations
    for the public:
       wiki.creativecommons.org/Considerations_for_licensees

=======================================================================

Creative Commons Attribution-ShareAlike 4.0 International Public
License

By exercising the Licensed Rights (defined below), You accept and agree
to be bound by the terms and conditions of this Creative Commons
Attribution-ShareAlike 4.0 International Public License ("Public
License"). To the extent this Public License may be interpreted as a
contract, You are granted the Licensed Rights in consideration of Your
acceptance of these terms and conditions, and the Licensor grants You
such rights in consideration of benefits the Licensor receives from
making the Licensed Material available under these terms and
conditions.


Section 1 -- Definitions.

  a. Adapted Material means material subject to Copyright and Similar
     Rights that is derived from or based upon the Licensed Material
     and in which the Licensed Material is translated, altered,
     arranged, transformed, or otherwise modified in a manner requiring
     permission under the Copyright and Similar Rights held by the
     Licensor. For purposes of this Public License, where the Licensed
     Material is a musical work, performance, or sound recording,
     Adapted Material is always produced where the Licensed Material is
     synched in timed relation with a moving image.

  b. Adapter's License means the license You apply to Your Copyright
     and Similar Rights in Your contributions to Adapted Material in
     accordance with the terms and conditions of this Public License.

  c. BY-SA Compatible License means a license listed at
     creativecommons.org/compatiblelicenses, approved by Creative
     Commons as essentially the equivalent of this Public License.

  d. Copyright and Similar Rights means copyright and/or similar rights
```

(continues on next page)

(continued from previous page)

```
     closely related to copyright including, without limitation,
     performance, broadcast, sound recording, and Sui Generis Database
     Rights, without regard to how the rights are labeled or
     categorized. For purposes of this Public License, the rights
     specified in Section 2(b)(1)-(2) are not Copyright and Similar
     Rights.

  e. Effective Technological Measures means those measures that, in the
     absence of proper authority, may not be circumvented under laws
     fulfilling obligations under Article 11 of the WIPO Copyright
     Treaty adopted on December 20, 1996, and/or similar international
     agreements.

  f. Exceptions and Limitations means fair use, fair dealing, and/or
     any other exception or limitation to Copyright and Similar Rights
     that applies to Your use of the Licensed Material.

  g. License Elements means the license attributes listed in the name
     of a Creative Commons Public License. The License Elements of this
     Public License are Attribution and ShareAlike.

  h. Licensed Material means the artistic or literary work, database,
     or other material to which the Licensor applied this Public
     License.

  i. Licensed Rights means the rights granted to You subject to the
     terms and conditions of this Public License, which are limited to
     all Copyright and Similar Rights that apply to Your use of the
     Licensed Material and that the Licensor has authority to license.

  j. Licensor means the individual(s) or entity(ies) granting rights
     under this Public License.

  k. Share means to provide material to the public by any means or
     process that requires permission under the Licensed Rights, such
     as reproduction, public display, public performance, distribution,
     dissemination, communication, or importation, and to make material
     available to the public including in ways that members of the
     public may access the material from a place and at a time
     individually chosen by them.

  l. Sui Generis Database Rights means rights other than copyright
     resulting from Directive 96/9/EC of the European Parliament and of
     the Council of 11 March 1996 on the legal protection of databases,
     as amended and/or succeeded, as well as other essentially
     equivalent rights anywhere in the world.

  m. You means the individual or entity exercising the Licensed Rights
     under this Public License. Your has a corresponding meaning.


Section 2 -- Scope.

  a. License grant.

       1. Subject to the terms and conditions of this Public License,
          the Licensor hereby grants You a worldwide, royalty-free,
```

(continues on next page)

(continued from previous page)

```
          non-sublicensable, non-exclusive, irrevocable license to
          exercise the Licensed Rights in the Licensed Material to:

            a. reproduce and Share the Licensed Material, in whole or
               in part; and

            b. produce, reproduce, and Share Adapted Material.

       2. Exceptions and Limitations. For the avoidance of doubt, where
          Exceptions and Limitations apply to Your use, this Public
          License does not apply, and You do not need to comply with
          its terms and conditions.

       3. Term. The term of this Public License is specified in Section
          6(a).

       4. Media and formats; technical modifications allowed. The
          Licensor authorizes You to exercise the Licensed Rights in
          all media and formats whether now known or hereafter created,
          and to make technical modifications necessary to do so. The
          Licensor waives and/or agrees not to assert any right or
          authority to forbid You from making technical modifications
          necessary to exercise the Licensed Rights, including
          technical modifications necessary to circumvent Effective
          Technological Measures. For purposes of this Public License,
          simply making modifications authorized by this Section 2(a)
          (4) never produces Adapted Material.

       5. Downstream recipients.

            a. Offer from the Licensor -- Licensed Material. Every
               recipient of the Licensed Material automatically
               receives an offer from the Licensor to exercise the
               Licensed Rights under the terms and conditions of this
               Public License.

            b. Additional offer from the Licensor -- Adapted Material.
               Every recipient of Adapted Material from You
               automatically receives an offer from the Licensor to
               exercise the Licensed Rights in the Adapted Material
               under the conditions of the Adapter's License You apply.

            c. No downstream restrictions. You may not offer or impose
               any additional or different terms or conditions on, or
               apply any Effective Technological Measures to, the
               Licensed Material if doing so restricts exercise of the
               Licensed Rights by any recipient of the Licensed
               Material.

       6. No endorsement. Nothing in this Public License constitutes or
          may be construed as permission to assert or imply that You
          are, or that Your use of the Licensed Material is, connected
          with, or sponsored, endorsed, or granted official status by,
          the Licensor or others designated to receive attribution as
          provided in Section 3(a)(1)(A)(i).

  b. Other rights.
```

(continues on next page)

(continued from previous page)

```
       1. Moral rights, such as the right of integrity, are not
          licensed under this Public License, nor are publicity,
          privacy, and/or other similar personality rights; however, to
          the extent possible, the Licensor waives and/or agrees not to
          assert any such rights held by the Licensor to the limited
          extent necessary to allow You to exercise the Licensed
          Rights, but not otherwise.

       2. Patent and trademark rights are not licensed under this
          Public License.

       3. To the extent possible, the Licensor waives any right to
          collect royalties from You for the exercise of the Licensed
          Rights, whether directly or through a collecting society
          under any voluntary or waivable statutory or compulsory
          licensing scheme. In all other cases the Licensor expressly
          reserves any right to collect such royalties.


Section 3 -- License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the
following conditions.

  a. Attribution.

       1. If You Share the Licensed Material (including in modified
          form), You must:

            a. retain the following if it is supplied by the Licensor
               with the Licensed Material:

                 i. identification of the creator(s) of the Licensed
                    Material and any others designated to receive
                    attribution, in any reasonable manner requested by
                    the Licensor (including by pseudonym if
                    designated);

                ii. a copyright notice;

               iii. a notice that refers to this Public License;

                iv. a notice that refers to the disclaimer of
                    warranties;

                 v. a URI or hyperlink to the Licensed Material to the
                    extent reasonably practicable;

            b. indicate if You modified the Licensed Material and
               retain an indication of any previous modifications; and

            c. indicate the Licensed Material is licensed under this
               Public License, and include the text of, or the URI or
               hyperlink to, this Public License.

       2. You may satisfy the conditions in Section 3(a)(1) in any
```

(continues on next page)

(continued from previous page)

```
           reasonable manner based on the medium, means, and context in
           which You Share the Licensed Material. For example, it may be
           reasonable to satisfy the conditions by providing a URI or
           hyperlink to a resource that includes the required
           information.

        3. If requested by the Licensor, You must remove any of the
           information required by Section 3(a)(1)(A) to the extent
           reasonably practicable.

  b. ShareAlike.

     In addition to the conditions in Section 3(a), if You Share
     Adapted Material You produce, the following conditions also apply.

        1. The Adapter's License You apply must be a Creative Commons
           license with the same License Elements, this version or
           later, or a BY-SA Compatible License.

        2. You must include the text of, or the URI or hyperlink to, the
           Adapter's License You apply. You may satisfy this condition
           in any reasonable manner based on the medium, means, and
           context in which You Share Adapted Material.

        3. You may not offer or impose any additional or different terms
           or conditions on, or apply any Effective Technological
           Measures to, Adapted Material that restrict exercise of the
           rights granted under the Adapter's License You apply.


Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that
apply to Your use of the Licensed Material:

  a. for the avoidance of doubt, Section 2(a)(1) grants You the right
     to extract, reuse, reproduce, and Share all or a substantial
     portion of the contents of the database;

  b. if You include all or a substantial portion of the database
     contents in a database in which You have Sui Generis Database
     Rights, then the database in which You have Sui Generis Database
     Rights (but not its individual contents) is Adapted Material,

     including for purposes of Section 3(b); and
  c. You must comply with the conditions in Section 3(a) if You Share
     all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not
replace Your obligations under this Public License where the Licensed
Rights include other Copyright and Similar Rights.


Section 5 -- Disclaimer of Warranties and Limitation of Liability.

  a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE
     EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS
```

(continues on next page)

(continued from previous page)

```
     AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF
     ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS,
     IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION,
     WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR
     PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS,
     ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT
     KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT
     ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.

  b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE
     TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION,
     NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT,
     INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES,
     COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR
     USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN
     ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR
     DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR
     IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.

  c. The disclaimer of warranties and limitation of liability provided
     above shall be interpreted in a manner that, to the extent
     possible, most closely approximates an absolute disclaimer and
     waiver of all liability.


Section 6 -- Term and Termination.

  a. This Public License applies for the term of the Copyright and
     Similar Rights licensed here. However, if You fail to comply with
     this Public License, then Your rights under this Public License
     terminate automatically.

  b. Where Your right to use the Licensed Material has terminated under
     Section 6(a), it reinstates:

       1. automatically as of the date the violation is cured, provided
          it is cured within 30 days of Your discovery of the
          violation; or

       2. upon express reinstatement by the Licensor.

     For the avoidance of doubt, this Section 6(b) does not affect any
     right the Licensor may have to seek remedies for Your violations
     of this Public License.

  c. For the avoidance of doubt, the Licensor may also offer the
     Licensed Material under separate terms or conditions or stop
     distributing the Licensed Material at any time; however, doing so
     will not terminate this Public License.

  d. Sections 1, 5, 6, 7, and 8 survive termination of this Public
     License.


Section 7 -- Other Terms and Conditions.

  a. The Licensor shall not be bound by any additional or different
```

(continues on next page)

(continued from previous page)

```
     terms or conditions communicated by You unless expressly agreed.

  b. Any arrangements, understandings, or agreements regarding the
     Licensed Material not stated herein are separate from and
     independent of the terms and conditions of this Public License.


Section 8 -- Interpretation.

  a. For the avoidance of doubt, this Public License does not, and
     shall not be interpreted to, reduce, limit, restrict, or impose
     conditions on any use of the Licensed Material that could lawfully
     be made without permission under this Public License.

  b. To the extent possible, if any provision of this Public License is
     deemed unenforceable, it shall be automatically reformed to the
     minimum extent necessary to make it enforceable. If the provision
     cannot be reformed, it shall be severed from this Public License
     without affecting the enforceability of the remaining terms and
     conditions.

  c. No term or condition of this Public License will be waived and no
     failure to comply consented to unless expressly agreed to by the
     Licensor.

  d. Nothing in this Public License constitutes or may be interpreted
     as a limitation upon, or waiver of, any privileges and immunities
     that apply to the Licensor or You, including from the legal
     processes of any jurisdiction or authority.


=======================================================================

Creative Commons is not a party to its public
licenses. Notwithstanding, Creative Commons may elect to apply one of
its public licenses to material it publishes and in those instances
will be considered the “Licensor.” The text of the Creative Commons
public licenses is dedicated to the public domain under the CC0 Public
Domain Dedication. Except for the limited purpose of indicating that
material is shared under a Creative Commons public license or as
otherwise permitted by the Creative Commons policies published at
creativecommons.org/policies, Creative Commons does not authorize the
use of the trademark "Creative Commons" or any other trademark or logo
of Creative Commons without its prior written consent including,
without limitation, in connection with any unauthorized modifications
to any of its public licenses or any other arrangements,
understandings, or agreements concerning use of licensed material. For
the avoidance of doubt, this paragraph does not form part of the
public licenses.

Creative Commons may be contacted at creativecommons.org.
```

CHAPTER 12

Indices and tables

- genindex
- modindex
- search

Index

T