

TI81XX PSP PCI Express Endpoint Driver User Guide



TI81XX PCI Express Endpoint Driver User Guide

Hemant Pedanekar

Introduction

This document is applicable to DM816x/AM389x, DM814x/AM387x and DM813x family of devices referred hereafter as TI816X, TI814X and TI813X respectively.

Descriptions common across these device families use the term TI81XX/ti81xx.

TI81XX devices have PCI Express hardware module which can either be configured to act as a Root Complex or a PCIe Endpoint. This document caters to the Endpoint mode of operation and describes the Driver needed to configure and operate on TI81XX PCI Express device as Endpoint.

This driver is referred as EP Driver and has to be loaded on TI81XX EP running Linux kernel from PSP package. It is expected that a userspace application would use this driver to configure the PCIe module and do transfers with peers.

NOTE: This document is applicable to PSP GIT kernel available from Arago git post PSP 04.00.01.13 release for TI816X and 04.01.00.06 release for TI814X or kernel from release 04.00.02.14/04.01.00.07/04.04.00.01 or later for respective platforms. Jump to Package Contents and Download section for details about getting all associated components.

Scope of the Document

At present, this document covers following areas:

- Details about the EP Driver
- Summary of applications and kernel modules provided as sample
- Build configuration and instructions
- Execution

Following items will be incorporated in future versions of this document:

- Internals of driver and sample code details - for now, please refer the respective source code.
- More details about setups
- Design and expected/recommended use cases of the EP driver
- EDMA usage details
- Throughput data

Assumptions and Conventions

Following points/prerequisites are assumed throughout this document:

1. If using PSP release 04.00.01.13, the PSP kernel tarball package from Arago git is extracted and build environment is set up as described in PSP User Guide associated with 04.00.01.13 release package.
2. Else, if using PSP release 04.00.02.14 (TI816X) or 04.01.00.07 (TI814X) or 04.04.00.01, the kernel tarball from release package is extracted and build environment is set up as described in User Guides packaged with respective releases.
3. For TI813X support, the latest kernel source from ti81xx-master branch on Arago is used.
4. All kernel modules/driver source and header file paths are relative to the kernel source directory where the package is extracted, except otherwise stated.
5. References to kernel modules provided as sample assume that the patches for the same are applied to kernel extracted in case #3 above. For other kernel versions, some manual merging of patches may be
6. References to sample applications assume the application patch is applied in an empty directory created by the user on the build host.
7. All application files are referred with path relative to the above directory.
8. The term "TI81XX EP" refers to local TI816X/TI814X/TI813X EP where the EP driver instance runs.

Terminology

- PCIe - PCI Express
- EP - PCIe Endpoint
- RC - PCIe Root Complex

Package Contents and Download

The EP Driver package consists of main driver along with sample test applications and supplementary drivers (custom samples).

- Out of these, the EP driver is available in kernel source for TI816X release 04.00.02.14 or TI814X release 04.01.00.07 and onwards. Users of earlier release may download the kernel tarball including first version of EP driver directly from TI81XX Arago tree on ti81xx-master branch from [here](#)^[1], though recommended approach is to get kernel source from tip of **ti81xx-master**^[2] branch on Arago to get *TI813X boot and EP driver support* and also avoid merge conflicts when applying the sample driver patches.
- The sample applications for RC side as well as EP side are in attached tarball. **Click here** to download. Further discussions assume that this file is downloaded on build host as `epdrv_sample_apps.tar.gz`.
- EDMA sample driver for TI81XX EP side is provided as patch file **here** to be applied to kernel snapshot from ti81xx-master branch
- Sample driver for TI81XX RC side is provide as a **patch** on the same kernel snapshot (after applying EDMA patch).

Supported Setups

Currently following setups are tested:

- TI814X RC <-> TI816X EP
- TI816X RC <-> TI816X EP
- TI816X RC <-> TI814X EP
- TI816X RC <-> TI813X EP

For details about doing these setups, please refer following documents:

TI81XX PCIe RC Driver User Guide ^[3]

TI81XX PCIe Boot Driver User Guide ^[4]

Note: There may be some porting effort and code changes required when non TI81XX RC is involved.

Not Supported / Known Issues

- Following warning is observed while building kernel and RC module for TI81XX after applying RC driver patch

```
warning: 'virt_to_bus' is deprecated (declared at /db/psp_git/users/a0393588/temp/linux-omap-2.6/arch/arm/include/asm/memory.h:222)
```

- This doesn't have any impact on execution and can be ignored
- Must follow the sequence of loading various components as described in steps in Execution section below, other sequences are not yet validated and may not work
- Interrupt reception and handling on EP side from RC is not supported (this any way is non-standard PCIe feature).
- MSI generation at EP side and handling at RC side is supported for throughput (THPT) mode tests.
- Running RC application on EP and vice versa may lead to issues and should be avoided.
- ioctl values for various drivers currently overlap with existing drivers in kernel. Particularly, debugging tools such as strace will treat the ioctls for the drivers as one of the existing ioctl already supported in kernel: eg., TI81XX_EDMA_SET_BUF_INFO is reported as SNDCTL_DSP_CHANNELS

EP Driver Summary

The EP driver controls the TI816X/TI814X/TI813X Endpoint device it is running on.

It creates a device node named "ti81xx_pcie_ep" under "/dev" to enable application to open and configure the device through character device interface.

Driver Calls

open() Application have to first open the device to get the file descriptor for calling subsequent functions.

ioctl() Various IOCTLS can be used by the application to access the PCIe hardware on local TI81XX EP.

mmap() MMAP interface can be used by the application using EP driver to map the reserved memory for PCIe as well as the h/w window of 256MB starting from 0x20000000.

close() Close the driver and cleanup outbound mappings.

IOCTLs Supported

TI81XX_SET_INBOUND Set inbound translation for a given BAR on TI81XX EP.

TI81XX_SET_OUTBOUND_SIZE Set the size for outbound window. The same size is applicable to all windows configured.

TI81XX_SET_OUTBOUND Create outbound window for specified address. This address has to be specified by the application such that a valid PCIe address is generated over the PCIe link.

TI81XX_ACCESS_REGS Read or write the specified PCIe module register - this could be from PCIe module's application space or local configuration space.

TI81XX_SEND_MSI Send MSI targeted to specified address. **Presently not supported.**

TI81XX_GET_PCIE_MEM_INFO Application can use this to query the size and physical address of the reserved memory on TI81XX EP for PCIe transfers. This memory space can then be divided by the application across different BARs (for inbound mapping) and to perform CPU/EDMA read/write transfers from EP. To access this memory in userspace, the application can use mmap facility provided by the EP driver (for CPU transfers) or by EDMA sample driver for EDMA transfers.

TI81XX_GET_OUTBOUND_STATUS Return a 32-bit mask indicating available outbound windows. A value '1' for specific bit position indicates corresponding window is free and not enabled for outbound.

TI81XX_CLR_OUTBOUND_MAP Clear outbound mapping for specified address. Corresponding window will be marked as free.

TI81XX_GET_INTR_CNTR Get the current count of received MSI interrupts. This count is cumulative from the initialization of opening of driver.

TI81XX_GET_BAR_INFO Returns the information associated with particular BAR such as address assigned to it (by RC) and size of the BAR.

EP Driver Build Configuration

The default TI81XX kernel build doesn't enable EP driver. Follow steps below to enable and build EP driver:

NOTE: Currently, EP driver can be enabled only if PCI Bus Support is disabled in kernel configuration. This is done to avoid any conflict when mistakenly running EP driver on a Root Complex.

Using the correct kernel

- Ensure that the kernel from downloaded from ti81xx-master branch is used and extracted on build host.

Ensuring default kernel configuration

- Configure the kernel with default options. E.g., for TI816X EVM,

```
make ARCH=arm ti8168_evm_defconfig
```

- When building for TI814X EVM as EP,

```
make ARCH=arm ti8148_evm_defconfig
```

- For TI813X EP,

```
make ARCH=arm dm385_evm_defconfig
```

Enter kernel configuration menu

```
make ARCH=arm menuconfig
```

You will be presented with menu such as below:

```
General setup --->
[*] Enable loadable module support --->
-* Enable the block layer --->
    System Type --->
        Bus support --->
...
...
Device Drivers --->
```

Disable PCI Bus Support

- Remove PCI bus support, which will in turn remove PCIe Root Complex Driver.
- Scroll till "Bus Support" listed above using DOWN arrow and press SPACE to enter the option menu:

```
[*] PCI support
[*] Message Signaled Interrupts (MSI and MSI-X)
[ ] PCI Debugging
< > PCI Stub driver
[ ] PCI IOV support
< > PCCard (PCMCIA/CardBus) support --->
```

Disable PCI support by pressing 'n' key on "PCI support" option. This will automatically disable other dependent configurations.

```
[ ] PCI support
< > PCCard (PCMCIA/CardBus) support --->
```

Now press RIGH arrow to highlight "Exit" and press ENTER to return to initial menu.

Enable EP Driver

Scroll with DOWN arrow key till "Device Drivers" option and press SPACE. This will present following menu:

```
Generic Driver Options --->
< > Connector - unified userspace <-> kernelspace linker --->
...
...
    Character devices --->
```

Scroll till "Character devices" and press SPACE, which will show a menu similar to following:

```
-*- Virtual terminal
...
...
< > TI81XX PCIe Endpoint Driver (NEW)
```

Using DOWN arrow key, go till PCIe Endpoint Driver option as shown above.

Press 'y' to build the driver into kernel

```
...
<*> TI81XX PCIe Endpoint Driver
```

or 'm' to build as loadable module.

```
...
<M> TI81XX PCIe Endpoint Driver
```

NOTE: At present, only modular build is supported/validated so it is recommended to build the driver as module.

Use RIGHT arrow key to highlight "Exit" and press ENTER successively till you are prompted to save the kernel configuration. Select "Yes" and press enter to exit the configuration menu.

Building EP Driver

Once the EP driver is enabled in kernel build configuration as described in earlier section, you can build it as follows:

Building statically with kernel

In this case, rebuild of kernel is required and driver will be built into kernel

```
make ARCH=arm uImage
```

Building as loadable module

NOTE: Even when loading the EP driver as module, you need to build the kernel at least once and make sure that it is used for EP before loading the module otherwise the EP driver may not get loaded successfully. In addition, to enable reserving memory at boot time and for successful EP driver build, PCIe EP driver must be selected in build configuration before building the kernel.

Build the kernel with PCI support disabled and EP driver selected to be built as module (refer earlier section for configuration):

```
make ARCH=arm uImage
```

Build kernel modules

```
make ARCH=arm modules
```

This will build EP driver module located inside `drivers/char` as:

```
drivers/char/ti81xx_pcie_epdrv.ko
```

Resource Reservation

The EP driver expects a contiguous block of memory reserved at boot time to facilitate applications to do mmap when doing data transfer using PCIe. The desired size should vary depending upon the application requirement.

By default there is no memory reserved for EP driver unless a kernel commandline argument "pcie_mem" with non-zero value specifying size of the memory to be reserved is passed.

NOTE1: PCIe EP driver must be included in kernel configuration to be able to reserve memory for PCIe.

NOTE2: The EP driver doesn't require any memory to be reserved for its own functionality, but this memory is required to provide mmap regions mapping to PCIe BARs (inbound transfers) and/or EDMA regions (outbound transfers) to allow application to perform PCIe communication and transfers.

An example for reserving 8 MB of RAM for PCIe EP, append following to kernel bootargs:

```
pcie_mem=8M
```

Note that, though the user can specify the size of memory to be reserved for PCIe, the start address of this chunk is not guaranteed to be same across different booting of the kernel and may vary due to parameters such as different boot arguments, kernel configuration and changes in kernel. The EP drivers provides an IOCTL to enable application

(using EP driver) to query the physical base address and size of the reserved PCIe memory block.

WARNING: If PCIe memory is not reserved during boot time, the EP driver would still load successfully but the application on EP might fail if it expects reserved RAM available to transfer data using mmap or DMA. To ensure that the driver had reserved memory available, the application must use `TI81XX_GET_PCIE_MEM_INFO` ioctl to check if reserved memory size fits application's requirement.

Prerequisites on initialization

EP driver expects that the PCIe h/w on (local) EP is already initialized before the driver is initialized. This includes:

- PCIe subsystem is set as PCIe Endpoint (h/w default)
- PCIe link is up. This means SerDes, PLL is setup to provide 2.5 GHz clock from 100MHz reference and PRCM is setup to enable PCIe module and LTSSM on.
- Root Complex has detected this EP and allocated resources such as memory BARs, interrupt required for its operation.

Following the "Execution" steps described in this document automatically ensures all of the above since PCIe boot operation of EP takes care of them.

In case you want to use a non-PCIe boot mode, all of the above needs to be taken care of before starting the EP driver. Refer TRM for PCIe h/w initialization details. Note that to ensure that the RC detected and configures this device, the PCIe initialization should be done at early stages during boot up.

Inbound Mapping

EP driver, on its own, doesn't create inbound mapping for any BAR. It is the application's responsibility to allot a contiguous region from PCIe reserved memory to create inbound mapping using IOCTL `TI81XX_SET_INBOUND`.

Note: It is recommended but not forced on the application to use the region from PCIe reserved memory to create inbound mapping. It may as well use a different mechanism to reserve memory (outside of EP driver) and pass the physical address etc to EP driver for setting up inbound translation.

Outbound Mapping

Unlike inbound, outbound mapping doesn't directly depend on the PCIe memory reserved, since it is internal masters (Cortex A8 CPU or EDMA) who need to be aware of (or provided with) memory to be used as source/destination for outbound and is independent of the PCIe and EP driver.

In this case also, a contiguous space for transfer is needed and it may be convenient to use a part of memory already reserved for PCIe. For example, in case the memory reserved is 8 MB, designate buffers in the lower 4 MB to carry out EDMA reads/writes to/drom.

NOTE: EP driver, on its own, doesn't provide any EDMA transfer functionality and it is the application's responsibility to set up outbound transfer parameters such as outbound mapping (using IOCTL `TI81XX_SET_OUTBOUND`), determining source/destination address for transfer, etc. before proceeding to use EDMA driver.

EP Driver Files

```
drivers/char/ti81xx_pcie_epdrv.h
drivers/char/ti81xx_pcie_epdrv.c
```

Kernel Files/Functions Added/Updated

- arch/arm/plat-omap/devices.c
 - Adds mechanism to reserve RAM for PCIe EP operations at boot time with size specified by "pcie_mem" boot argument.
- arch/arm/plat-omap/include/plat/common.h
- arch/arm/plat-omap/common.c
 - Reserve memory during boot

Sample Components

The various sample components to utilize the EP driver on TI81XX EP are attached to this page.

These include:

- EP side: Sample kernel module for EDMA
- RC side: Sample kernel module for TI81XX RC which provides RC side application with access over PCI
- RC & EP side: Sample Applications

The details are covered in individual sections below.

EDMA Kernel Module

This module is provided as a sample driver to access EDMA functionality from user space. The main purpose of this driver is to enable the EP side application to initiate EDMA transfers to the peer over PCIe. Note that since this is a sample driver, it is not integrated into kernel and needs to be applied as patch to the kernel source from release.

Driver Files

```
drivers/char/ti81xx_edma.h drivers/char/ti81xx_edma.c
```

Driver Calls

Please refer driver files and EP application code for reference.

IOCTLs supported

Please refer driver files and EP application code for reference.

Building The Module

Apply the attached patch *0001-char-ti81xx-Add-EDMA-driver-to-provide-minimal-acces.patch* by copying it to kernel source directory. Ensure that the kernel is already configured for TI81XX EP as mentioned earlier.

The download link mentioned earlier has the patch in gzipped format so it needs to be extracted to get the patch file. You can then use 'patch' utility to apply the patch as:

```
# gunzip 0001-char-ti81xx-Add-EDMA-driver-to-provide-minimal-acces.patch.gz
# patch -p 1 -i 0001-char-ti81xx-Add-EDMA-driver-to-provide-minimal-acces.patch
```

NOTE: The above step directly modifies the drivers/char/Makefile to build this sample driver always as loadable kernel module.

Build module with following command

```
make ARCH=arm modules
```

The module will be built as

```
drivers/char/ti81xx_edma.ko
```

RC Kernel Module

This sample driver needs to be applied to the RC kernel. This means it must have PCI bus support with MSI enabled.

Note: At present, only TI816X/TI814X/TI813X RC is supported. Also, to apply this patch cleanly, the kernel extracted earlier (for building EP driver) needs to be used.

Driver Files

drivers/char/ti81xx_pcie_rcdrv.h drivers/char/ti81xx_pcie_rcdrv.c

Driver Calls

Please refer driver files and RC application code for reference.

IOCTLs supported

Please refer driver files and RC application code for reference.

Building The Module

Apply the attached patch *0002-char-ti81xx-Add-driver-for-controlling-TI81XX-EP-fro.patch* by copying it to kernel source directory. Ensure that the patch for EDMA sample module is already applied (as mentioned above), this will avoid conflicts.

The download link mentioned earlier has the patch in gzipped format so it needs to be extracted to get the patch file. You can then use 'patch' utility to apply the patch as:

```
# gunzip 0002-char-ti81xx-Add-driver-for-controlling-TI81XX-EP-fro.patch.gz
# patch -p 1 -i 0002-char-ti81xx-Add-driver-for-controlling-TI81XX-EP-fro.patch
```

NOTE: The above step directly modifies the drivers/char/Makefile to build this sample driver always as loadable kernel module when PCI MSI (CONFIG_PCI_MSI) support is enabled.

Ensuring default kernel configuration

- Configure the kernel with default options. E.g., for TI816X EVM,

```
make ARCH=arm ti8168_evm_defconfig
```

- In case using TI814X EVM as RC, use,

```
make ARCH=arm ti8148_evm_defconfig
```

- For TI813X RC,

```
make ARCH=arm dm385_evm_defconfig
```

Build the kernel for RC

```
make ARCH=arm uImage
```

Build RC module

```
make ARCH=arm modules
```

The module will be built as

```
drivers/char/ti81xx_pcie_rcdrv.ko
```

Sample Applications

The attached tarball `epdrv_sample_apps.tar.gz` contains RC and EP side sample applications which use various kernel modules described above to communicate and perform data transfers between them across PCIe.

Currently the EP application requires minimum 8MB of memory reserved for PCIe transfers, thus, the kernel command line must be passed `pcie_mem=8M` at least. If the memory reserved is less, then the application will exit.

Files

```
./Makefile
./lib
./lib/Makefile
./lib/ti81xx_ep_lib.c
./lib/ti81xx_mgmt_lib.c
./lib/ti81xx_pci_info.c
./lib/ti81xx_trans.c
./include
./include/debug_msg.h
./include/ti81xx_ep_lib.h
./include/ti81xx_mgmt_lib.h
./include/ti81xx_pci_info.h
./include/ti81xx_trans.h
./application
./application/makefile
./application/ti81xx_ep_app.c
./application/ti81xx_rc_app.c
```

Building Applications

Create an empty directory and extract the attached tarball `epdrv_sample_apps.tar.gz` into that directory.

E.g., assuming the directory created is named 'apps' and `epdrv_sample_apps.tar.gz` is copied to it,

```
# cd apps
# tar -xzvf eprv_sample_apps.tar.gz
```

Build the applications for TI81XX RC and TI81XX EP using following command

```
# make KERNSRC=<absolute-path-to-TI81XX-kernel-source>
```

Remember to pass correct path to kernel source where the kernel tarball mentioned earlier is extracted.

The above command will build EP and RC applications at following paths respectively:

```
application/ep_app
application/rc_app
```

In case the RC kernel source is different than EP kernel (e.g., when using PC -> TI81XX setup), the RC and EP side application components need to be built separately in two steps as below example:

```
# make KERNSRC=<absolute-path-to-RC-kernel-source> rc
# make KERNSRC=<absolute-path-to-TI81XX-EP-kernel-source> ep
```

You may use `make clean` in between to force rebuild of the library files which are common for RC & EP. Note that this will also delete the application built in earlier step, so ensure it is copied elsewhere.

To clean the built files, you can use

```
# make clean
```

Note: Currently only default mode of applications is supported (DEMO mode). The other 2 modes - throughput and sanity mode are not supported.

Execution

Following sequence needs to be executed to run the demo with above components (commands with 'EP#' to be executed on EP size while those with 'RC#' to be executed on RC side):

- Ensure the desire PCIe setup is done - in this case, we assume TI816X <-> TI816X setup.
- Power on TI816X EP
- Power on TI816X RC
- Boot EP from RC (refer Boot Driver User Guide)

Note: It should be possible to skip this step and instead have EP booted from any non PCIe boot mode (e.g, NAND boot) but it must be ensured that PCIe h/w is initialized, enumerated (from RC) and EP kernel passed memory size to be reserved as mentioned in next step before loading EP driver.

- Make sure that at least 8 MB of PCIe memory is reserved by appending `pcie_mem=8M` (or more) to the 'bootargs' to the EP kernel. You may also need to ensure that the 'mem' argument passed to kernel ensures sufficient memory is available.
- Ensure that EP Driver, EDMA driver and EP sample application are present in a directory on EP filesystem.
- Once shell is available on EP side, navigate to the directory containing the above files. Load EP Driver and EDMA sample driver:

```
EP# insmod ti81xx_pcie_epdrv.ko
EP# insmod ti81xx_edma.ko
```

- After successful load of these modules, run EP application:

```
EP# ./ep_app
```

- You may need to add executable permission to the application before above step:

```
EP# chmod a+x ep_app
```

- The EP application will not wait for communication with peer.
- On TI816X RC side, make sure that the RC driver and application are present in working directory.
- Navigate to that directory and load the RC driver

```
RC# insmod ti81xx_pcie_rcdrv.ko
```

- Run RC application (again, you may need to run `chmod a+x rc_app` before this step)

```
RC# ./rc_app
```

Since these applications are built with DEMO mode, they will start communicating each other and do data transfer in either directions using CPU (read/write from RC) and EDMA (read/write from EP).

During execution of the applications, on RC and EP consoles, you would see chunks with 'C', 'D', 'L' characters. The meaning is explained below:

```
on RC, data represented by 'C' is sent by memcpy from EP
on RC, data represented by 'D' is sent by EDMA from EP
on EP, data represented by 'C' is sent by memcpy from RC
on EP, data represented by 'L' is data that has been read by EDMA from a buffer on RC populated by application on RC.
```

- To exit the DEMO, terminate RC and EP apps - this can be done by pressing Ctrl + C combination at shell on both RC and EP. Note that there are no log files generated for DEMO mode.

Getting Throughput from EDMA transfers

For throughput measurements, MSI interrupt from EP to RC are used for control communication.

- You can also build the applications in throughput mode to get the throughput data for EDMA transfers from EP (read/write).
- To set this mode, build the applications by passing `MODE=THPT` as argument to 'make' command.
- Assuming we are building the applications for TI816X (RC) <-> TI813X (EP), using following commands on build host in EP sample applications directory

```
# make clean
# make KERNSRC=<path-to-TI81XX-kernel-source> MODE=THPT
```

- This will build the applications `ep_app` and `rc_app` inside application directory. Transfer them to RC and EP as mentioned earlier and set as executable.
- It is assumed that the required modules are loaded on RC and EP after fresh boot as described earlier to proceed application execution
- On TI813X EP, start `ep_app` followed by starting `rc_app` on RC.
- In the example setup, following messages were seen
 - On TI813X EP

```
THPT calculated in TX direction is : 318.012421 MBPS
Executing test case THPT RX (RC read from EP)
THPT calculated in RX direction is : 302.421722 MBPS
Executing test case THPT TX/RX (simultaneous read & write from EP)
THPT calculated in TX direction is : 154.729523 MBPS
THPT calculated in RX direction is : 154.729523 MBPS
Total thpt calculated tx+rx is: 309.459045 MBPS
```

- On TI816X RC

```
root@am389x-evm:/home/hemant/hemant/ti816x_pcie# ./rc_app
***Forcing polling (interrupt communication disabled, RC will not generate interrupt)
```

References

- [1] <http://arago-project.org/git/projects/?p=linux-omap3.git;a=snapshot;h=3dbd693bc86cd3ca0b79080d06b8056f490fe947;sf=tgz>
- [2] <http://arago-project.org/git/projects/?p=linux-omap3.git;a=shortlog;h=refs/heads/ti81xx-master>
- [3] http://processors.wiki.ti.com/index.php/TI81XX_PCI_Express_Root_Complex_Driver_User_Guide
- [4] http://processors.wiki.ti.com/index.php/TI81XX_PCI_Express_Endpoint_Boot_Driver_User_Guide

Article Sources and Contributors

TI18XX PSP PCI Express Endpoint Driver User Guide *Source:* <http://processors.wiki.ti.com/index.php?oldid=119727> *Contributors:* Hemantp, Kevinsc, Kumar, RK

Image Sources, Licenses and Contributors

Image:TIBanner.png *Source:* <http://processors.wiki.ti.com/index.php?title=File:TIBanner.png> *License:* unknown *Contributors:* Nsnehaprabha

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

License

1. Definitions

- a. **"Adaptation"** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.
- b. **"Collection"** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License.
- c. **"Creative Commons Compatible License"** means a license that is listed at <http://creativecommons.org/compatlicenses> that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of adaptations of works made available under that license under this License or a Creative Commons jurisdiction license with the same License Elements as this License.
- d. **"Distribute"** means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.
- e. **"License Elements"** means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.
- f. **"Licensor"** means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
- g. **"Original Author"** means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.
- h. **"Work"** means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.
- i. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
- j. **"Publicly Perform"** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.
- k. **"Reproduce"** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights

Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
- b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
- c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
- d. to Distribute and Publicly Perform Adaptations.
- e. For the avoidance of doubt:
- i. **Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
- ii. **Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
- iii. **Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions

The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

- a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.
- b. You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the "Applicable License"), you must comply with the terms of the Applicable License generally and the following provisions: (I) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (II) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (IV) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.
- c. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.
- d. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

- a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
- b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

- a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
- b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
- c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
- d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
- e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
- f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.