



Open Source Used In STAR 1.0

Cisco Systems, Inc.

www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number: 78EE117C99-1106135427

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please contact us at external-opensource-requests@cisco.com.

In your requests please include the following reference number 78EE117C99-1106135427

Contents

1.1 atf-v1-2-marvell-patches 17.01.0

1.1.1 Available under license

1.2 gobi-drivers SWI_2.25

1.2.1 Available under license

1.3 arm-trusted-firmware-atf 1.2

1.3.1 Available under license

1.1 atf-v1-2-marvell-patches 17.01.0

1.1.1 Available under license :

Copyright (c) 2013-2017, ARM Limited and Contributors. All rights reserved.

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

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

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.

Neither the name of ARM 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.

The following text has been copied from Marvell Limited Use License Agreement for 7K/8K Source Code for Cisco and for this asset the license is in row 2:

```
#####  
OPEN SOURCE DELIVERABLES
```

The following are Open Source Deliverables:

[ARM Trusted Firmware (ATF) " Marvell patches] [Marvell] [Source code] [BSD3
<https://github.com/ARM-software/arm-trusted-firmware/blob/master/license.rst>]

1.2 gobi-drivers SWI_2.25

1.2.1 Available under license :

```
/*=====
```

FILE:

GobiSerial.c

DESCRIPTION:

Linux Qualcomm Serial USB driver Implementation

PUBLIC DRIVER FUNCTIONS:

GobiProbe

GobiOpen

GobiClose

GobiReadBulkCallback

GobiSerialSuspend

GobiSerialResume (if kernel is less than 2.6.24)

Copyright (c) 2011, Code Aurora Forum. All rights reserved.

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

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * 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.
- * Neither the name of Code Aurora Forum nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written

```

permission.
/*=====
FILE:
GobiUSBNet.c

DESCRIPTION:
Qualcomm USB Network device for Gobi 3000

FUNCTIONS:
GobiNetSuspend
GobiNetResume
GobiNetDriverBind
GobiNetDriverUnbind
GobiUSBNetURBCallback
GobiUSBNetTXTimeout
GobiUSBNetAutoPMThread
GobiUSBNetStartXmit
GobiUSBNetOpen
GobiUSBNetStop
GobiUSBNetProbe
GobiUSBNetModInit
GobiUSBNetModExit

```

Copyright (c) 2011, Code Aurora Forum. All rights reserved.

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

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * 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.
- * Neither the name of Code Aurora Forum nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

1.3 arm-trusted-firmware-atf 1.2

1.3.1 Available under license :

```

#ifndef _FDT_H
#define _FDT_H
/*
* libfdt - Flat Device Tree manipulation
* Copyright (C) 2006 David Gibson, IBM Corporation.
* Copyright 2012 Kim Phillips, Freescale Semiconductor.
*

```

* libfdt is dual licensed: you can use it either under the terms of
* the GPL, or the BSD license, at your option.
*
* a) This library is free software; you can redistribute it and/or
* modify it under the terms of the GNU General Public License as
* published by the Free Software Foundation; either version 2 of the
* License, or (at your option) any later version.
*
* This library is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public
* License along with this library; if not, write to the Free
* Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston,
* MA 02110-1301 USA
*
* Alternatively,
*
* b) 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.
*
* 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 OWNER 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.
*/

/*

* Portions copyright (c) 2016, ARM Limited and Contributors.

* All rights reserved.

*/

#ifndef __ASSEMBLY__

#include <libfdt_env.h>

```
struct fdt_header {
    fdt32_t magic; /* magic word FDT_MAGIC */
    fdt32_t totalsize; /* total size of DT block */
    fdt32_t off_dt_struct; /* offset to structure */
    fdt32_t off_dt_strings; /* offset to strings */
    fdt32_t off_mem_rsvmap; /* offset to memory reserve map */
    fdt32_t version; /* format version */
    fdt32_t last_comp_version; /* last compatible version */
```

/* version 2 fields below */

```
    fdt32_t boot_cpuid_phys; /* Which physical CPU id we're
        booting on */
```

/* version 3 fields below */

```
    fdt32_t size_dt_strings; /* size of the strings block */
```

/* version 17 fields below */

```
    fdt32_t size_dt_struct; /* size of the structure block */
```

```
};
```

```
struct fdt_reserve_entry {
```

```
    fdt64_t address;
```

```
    fdt64_t size;
```

```
};
```

```
struct fdt_node_header {
```

```
    fdt32_t tag;
```

```
    char name[];
```

```
};
```

```
struct fdt_property {
```

```
    fdt32_t tag;
```

```
    fdt32_t len;
```

```
    fdt32_t nameoff;
```

```
    char data[];
```

```
};
```

```
#endif /* !__ASSEMBLY */
```

```
#define FDT_MAGIC 0xd00dfeed /* 4: version, 4: total size */
```

```

#define FDT_TAGSIZE sizeof(fdt32_t)

#define FDT_BEGIN_NODE 0x1 /* Start node: full name */
#define FDT_END_NODE 0x2 /* End node */
#define FDT_PROP 0x3 /* Property: name off,
    size, content */
#define FDT_NOP 0x4 /* nop */
#define FDT_END 0x9

#define FDT_V1_SIZE (7*sizeof(fdt32_t))
#define FDT_V2_SIZE (FDT_V1_SIZE + sizeof(fdt32_t))
#define FDT_V3_SIZE (FDT_V2_SIZE + sizeof(fdt32_t))
#define FDT_V16_SIZE FDT_V3_SIZE
#define FDT_V17_SIZE (FDT_V16_SIZE + sizeof(fdt32_t))

#endif /* _FDT_H */

```

Copyright (c) 2013-2014, ARM Limited and Contributors. All rights reserved.

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

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

- * 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.

- * Neither the name of ARM 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.

#

Copyright (c) 2016, ARM Limited and Contributors. All rights reserved.

#

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

#

```
# Redistributions of source code must retain the above copyright notice, this
# list of conditions and the following disclaimer.
#
# 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.
#
# Neither the name of ARM 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.
#
```

```
LIBFDT_SRCS := $(addprefix lib/libfdt/, \
    fdt.c \
    fdt_addresses.c \
    fdt_empty_tree.c \
    fdt_ro.c \
    fdt_rw.c \
    fdt_strerror.c \
    fdt_sw.c \
    fdt_wip.c) \
```

```
INCLUDES += -Iinclude/lib/libfdt
```

```
/*
```

```
* libfdt - Flat Device Tree manipulation
* Copyright (C) 2006 David Gibson, IBM Corporation.
*
* libfdt is dual licensed: you can use it either under the terms of
* the GPL, or the BSD license, at your option.
*
* a) This library is free software; you can redistribute it and/or
* modify it under the terms of the GNU General Public License as
* published by the Free Software Foundation; either version 2 of the
* License, or (at your option) any later version.
*
* This library is distributed in the hope that it will be useful,
```


* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.

*
* You should have received a copy of the GNU General Public
* License along with this library; if not, write to the Free
* Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston,
* MA 02110-1301 USA

* Alternatively,

*
* b) 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.

* 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 OWNER 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.

*/

```
#include "libfdt_env.h"
```

```
#include <fdt.h>
```

```
#include <libfdt.h>
```

```
#include "libfdt_internal.h"
```

```
int fdt_check_header(const void *fdt)
{
    if (fdt_magic(fdt) == FDT_MAGIC) {
        /* Complete tree */
    }
}
```

```

if (fdt_version(fdt) < FDT_FIRST_SUPPORTED_VERSION)
    return -FDT_ERR_BADVERSION;
if (fdt_last_comp_version(fdt) > FDT_LAST_SUPPORTED_VERSION)
    return -FDT_ERR_BADVERSION;
} else if (fdt_magic(fdt) == FDT_SW_MAGIC) {
    /* Unfinished sequential-write blob */
    if (fdt_size_dt_struct(fdt) == 0)
        return -FDT_ERR_BADSTATE;
    } else {
        return -FDT_ERR_BADMAGIC;
    }
}

return 0;
}

```

```

const void *fdt_offset_ptr(const void *fdt, int offset, unsigned int len)
{
    const char *p;

    if (fdt_version(fdt) >= 0x11)
        if (((offset + len) < offset)
            || ((offset + len) > fdt_size_dt_struct(fdt)))
            return NULL;

    p = _fdt_offset_ptr(fdt, offset);

    if (p + len < p)
        return NULL;
    return p;
}

```

```

uint32_t fdt_next_tag(const void *fdt, int startoffset, int *nextoffset)
{
    const fdt32_t *tagp, *lenp;
    uint32_t tag;
    int offset = startoffset;
    const char *p;

    *nextoffset = -FDT_ERR_TRUNCATED;
    tagp = fdt_offset_ptr(fdt, offset, FDT_TAGSIZE);
    if (!tagp)
        return FDT_END; /* premature end */
    tag = fdt32_to_cpu(*tagp);
    offset += FDT_TAGSIZE;

    *nextoffset = -FDT_ERR_BADSTRUCTURE;
    switch (tag) {
    case FDT_BEGIN_NODE:

```

```

/* skip name */
do {
    p = fdt_offset_ptr(fdt, offset++, 1);
} while (p && (*p != '\0'));
if (!p)
    return FDT_END; /* premature end */
break;

case FDT_PROP:
    lenp = fdt_offset_ptr(fdt, offset, sizeof(*lenp));
    if (!lenp)
        return FDT_END; /* premature end */
    /* skip-name offset, length and value */
    offset += sizeof(struct fdt_property) - FDT_TAGSIZE
        + fdt32_to_cpu(*lenp);
    break;

case FDT_END:
case FDT_END_NODE:
case FDT_NOP:
    break;

default:
    return FDT_END;
}

if (!fdt_offset_ptr(fdt, startoffset, offset - startoffset))
    return FDT_END; /* premature end */

*nextoffset = FDT_TAGALIGN(offset);
return tag;
}

int _fdt_check_node_offset(const void *fdt, int offset)
{
    if ((offset < 0) || (offset % FDT_TAGSIZE)
        || (fdt_next_tag(fdt, offset, &offset) != FDT_BEGIN_NODE))
        return -FDT_ERR_BADOFFSET;

    return offset;
}

int _fdt_check_prop_offset(const void *fdt, int offset)
{
    if ((offset < 0) || (offset % FDT_TAGSIZE)
        || (fdt_next_tag(fdt, offset, &offset) != FDT_PROP))
        return -FDT_ERR_BADOFFSET;
}

```

```

return offset;
}

int fdt_next_node(const void *fdt, int offset, int *depth)
{
    int nextoffset = 0;
    uint32_t tag;

    if (offset >= 0)
        if ((nextoffset = _fdt_check_node_offset(fdt, offset)) < 0)
            return nextoffset;

    do {
        offset = nextoffset;
        tag = fdt_next_tag(fdt, offset, &nextoffset);

        switch (tag) {
            case FDT_PROP:
            case FDT_NOP:
                break;

            case FDT_BEGIN_NODE:
                if (depth)
                    (*depth)++;
                break;

            case FDT_END_NODE:
                if (depth && (--(*depth)) < 0)
                    return nextoffset;
                break;

            case FDT_END:
                if ((nextoffset >= 0)
                    || ((nextoffset == -FDT_ERR_TRUNCATED) && !depth))
                    return -FDT_ERR_NOTFOUND;
                else
                    return nextoffset;
        }
    } while (tag != FDT_BEGIN_NODE);

    return offset;
}

int fdt_first_subnode(const void *fdt, int offset)
{
    int depth = 0;

    offset = fdt_next_node(fdt, offset, &depth);
}

```

```

if (offset < 0 || depth != 1)
    return -FDT_ERR_NOTFOUND;

return offset;
}

int fdt_next_subnode(const void *fdt, int offset)
{
    int depth = 1;

    /*
     * With respect to the parent, the depth of the next subnode will be
     * the same as the last.
     */
    do {
        offset = fdt_next_node(fdt, offset, &depth);
        if (offset < 0 || depth < 1)
            return -FDT_ERR_NOTFOUND;
    } while (depth > 1);

    return offset;
}

const char *_fdt_find_string(const char *strtab, int tabsize, const char *s)
{
    int len = strlen(s) + 1;
    const char *last = strtab + tabsize - len;
    const char *p;

    for (p = strtab; p <= last; p++)
        if (memcmp(p, s, len) == 0)
            return p;
    return NULL;
}

int fdt_move(const void *fdt, void *buf, int bufsize)
{
    FDT_CHECK_HEADER(fdt);

    if (fdt_totalsize(fdt) > bufsize)
        return -FDT_ERR_NOSPACE;

    memmove(buf, fdt, fdt_totalsize(fdt));
    return 0;
}

```

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a

partnership relationship between Cisco and any other company. (1110R)

©2020 Cisco Systems, Inc. All rights reserved.