The Embedded I/O Company

TEWS TECHNOLOGIES

TXMC590-SW-82

Linux Device Driver

16 Channel Thermo-/Strain Measurement

Version 1.0.x

User Manual

Issue 1.0.0

March 2019

powerBridge Computer

Ehlbeek 15a
30938 Burgwedel
fon 05139-9980-0
fax 05139-9980-49

www.powerbridge.de
info@powerbridge.de

TEWS TECHNOLOGIES GmbH
Am Bahnhof 7 25469 Halstenbek, Germany
(0) 4101 4058 0 Fax: +49 (0) 4101 4058 19
info@tews.com www.tews.com

TXMC590-SW-82

Linux Device Driver

16 Channel Thermo-/Strain Measurement

Supported Modules:
TXMC590

Issue	Description	Date
1.0.0	First Issue	March 11, 2019

Table of Contents

1 Introduction

The TXMC590-SW-82 Linux device driver allows the operation of a TXMC590 Thermo-/Strain Measurement XMC on Linux operating systems.

The TXMC590-SW-82 device driver software includes the following features:

- Configure channels
- Read temperature value (RTD/Thermocouple)
- Read strain value (Strain gauge)
- Calibrate channel (local ADC calibration)
- Configure cold junction channel
- Load configuration tables for custom devices

The TXMC590-SW-82 device driver supports the modules listed below:

TXMC590	16 channel temperature acquisition board	XMC

To get more information about the features and use of the supported devices it is recommended to read the manuals listed below.

TXMC590 Hardware User Manual

2 Installation

Following files are located on the distribution media:

Directory path 'TXMC590-SW-82':

TXMC590-SW-82-1.0.0.pdf	This manual in PDF format
TXMC590-SW-82-SRC.tar.gz	GZIP compressed archive with driver source code
Release.txt	Release information
ChangeLog.txt	Release history

The GZIP compressed archive TXMC590-SW-82-SRC.tar.gz contains the following files and directories:

Directory path './txmc590/':

txmc590.c	Driver source code
txmc590def.h	Driver include file
txmc590.h	Driver include file for application program
Makefile	Device driver make file
makenode	Script for device node creation
COPYING	Copy of the GNU Public License (GPL)
api/txmc590api.h	API include file
api/txmc590api.c	API source file
include/config.h	Include of system dependent config.h
include/tpxxxhwdep.c	Low level hardware access functions source file
include/tpxxxhwdep.h	Access functions header file
include/tpmodule.c	Driver independent library
include/tpmodule.h	Driver independent library header file
example/txmc590exa.c	Example application
example/Makefile	Example application makefile

In order to perform an installation, extract all files of the archive TXMC590-SW-82-SRC.tar.gz to the desired target directory. The command 'tar -xzvf TXMC590-SW-82-SRC.tar.gz' will extract the files into the local directory.

- Login as *root* and change to the target directory
- Copy txmc590.h and api/txmc590api.h to */usr/include*

2.1 Build and install the Device Driver

- Login as *root*
- Change to the target directory
- To create and install the driver in the module directory */lib/modules/<version>/misc* enter:

  # make install

- To update the device driver's module dependencies, enter:

  # **depmod -a**

2.2 Uninstall the Device Driver

- Login as *root*
- Change to the target directory
- To remove the driver from the module directory */lib/modules/<version>/misc* enter:

  # make uninstall

2.3 Install Device Driver into the running Kernel

- To load the device driver into the running kernel, login as root and execute the following commands:

  # modprobe txmc590drv

- After the first build or if you are using dynamic major device allocation it is necessary to create new device nodes on the file system. Please execute the script file *makenode* to do this. If your kernel has enabled a device file system (devfs or sysfs with udev) then you have to skip running the *makenode* script. Instead of creating device nodes from the script the driver itself takes creating and destroying of device nodes in its responsibility.

  # sh makenode

On success, the device driver will create a minor device for each TXMC590 module found. The first module of the first TXMC590 module can be accessed with device node /dev/txmc590_0, the second module with device node /dev/txmc590_1, the third TXMC590 module with device node /dev/txmc590_2 and so on.

The assignment of device nodes to physical TXMC590 modules depends on the search order of the PCI bus driver.

2.4 Remove Device Driver from the running Kernel

- To remove the device driver from the running kernel login as root and execute the following command:

  # modprobe –r txmc590drv

If your kernel has enabled devfs or sysfs (udev), all /dev/txmc590_x nodes will be automatically removed from your file system after this.

Be sure that the driver isn't opened by any application program. If opened you will get the response "*txmc590drv: Device or resource busy*" and the driver will still remain in the system until you close all opened files and execute *modprobe –r* again.

2.5 Change Major Device Number

This paragraph is only for Linux kernels without dynamic device management. The TXMC590 driver uses dynamic allocation of major device numbers per default. If this isn't suitable for the application it's possible to define a major number for the driver.

To change the major number edit the file txmc590def.h, change the following symbol to appropriate value and enter `make install` to create a new driver.

TXMC590_MAJOR	Valid numbers are in range between 0 and 255. A value of 0 means dynamic number allocation.

Example:

```
#define TXMC590_MAJOR   122
```

Be sure that the desired major number is not used by other drivers. Please check */proc/devices* to see which numbers are free.

Keep in mind that it is necessary to create new device nodes if the major number for the TXMC590 driver has changed and the makenode script is not used.

3 API Documentation

3.1 General Functions

3.1.1 txmc590Open

NAME

txmc590Open – opens a device.

SYNOPSIS

```
TXMC590_HANDLE txmc590Open
(
      char          *DeviceName
)
```

DESCRIPTION

Before I/O can be performed to a device, a device descriptor must be opened by a call to this function.

PARAMETERS

DeviceName

This parameter points to a null-terminated string that specifies the name of the device. The first TXMC590 device is named “/dev/txmc590_0”, the second device is named “/dev/txmc590_1”, and so on.

EXAMPLE

```
#include “txmc590api.h”

TXMC590_HANDLE   hdl;

/*
** open the specified device
*/
hdl = txmc590Open(“/txmc590/0”);
if (hdl == NULL)
{
   /* handle open error */
}
```

RETURNS

A device handle, or NULL if the function fails. An error code will be stored in *errno*.

ERROR CODES

The error codes are stored in *errno.*

The error code is a standard error code set by the I/O system.

3.1.2 txmc590Close

NAME

txmc590Close – closes a device.

SYNOPSIS

```
TXMC590_STATUS txmc590Close
(
      TXMC590_HANDLE        hdl
)
```

DESCRIPTION

This function closes previously opened devices.

PARAMETERS

hdl

This value specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

EXAMPLE

```
#include "txmc590api.h"

TXMC590_HANDLE   hdl;
TXMC590_STATUS   result;

/*
** close the device
*/
result = txmc590Close(hdl);
if (result != TXMC590_OK)
{
   /* handle close error */
}
```

RETURNS

On success, TXMC590_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TXMC590_ERR_INVALID_HANDLE	The specified device handle is invalid

3.1.3 txmc590GetModuleInfo

NAME

txm590GetModuleInfo – get information of the module

SYNOPSIS

```
TXMC590_STATUS txmc590GetModuleInfo
(
    TXMC590_HANDLE      hdl,
    TXMC590_INFO_BUF    *pPciInfoBuf
)
```

DESCRIPTION

This function returns information about the module, including PCI header as well as the PCI localization.

PARAMETERS

hdl

This argument specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

pPciInfoBuf

This argument is a pointer to the structure TXMC590_INFO_BUF that receives information of the module PCI header.

```
typedef struct
{
    unsigned int        pciBusNo;
    unsigned int        pciDevNo;
    unsigned int        pciFuncNo;
    unsigned short      vendorId;
    unsigned short      deviceId;
    unsigned short      subSystemId;
    unsigned short      subVendorId;
    unsigned int        variant;
    unsigned int        fpgaRev;
    unsigned int        aducRev[16];
} TXMC590_INFO_BUF;
```

pciBusNo

Number of the PCI bus, where the module resides.

pciDevNo

PCI device number

pciFuncNo

PCI function number

vendorId

PCI module vendor ID.

deviceId

PCI module device ID

subSystemId

PCI module sub system ID

subVendorId

PCI module sub vendor ID

variant

Module variant (e.g. 10 for TXMC590-10)

fpgaRev

FPGA Code revision. Format: MMmmRRbb

MM - major version

mm - minor version

RR – revision

bb – reserved - internal build code

aducRev[]

ADuC-Firmware version for each channel. Format: MMmmRRbb

MM - major version

mm - minor version

RR – revision

bb – reserved - internal build code

EXAMPLE

```
#include "txmc590api.h"

TXMC590_HANDLE     hdl;
TXMC590_STATUS     result;
TXMC590_INFO_BUF   infoBuf;

/*
** get module information
*/
result = txmc590GetModuleInfo( hdl, &infoBuf );
if (result != TXMC590_OK)
{
  /* handle error */
}
```

RETURN VALUE

On success, TXMC590_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TXMC590_ERR_INVALID_HANDLE	The specified device handle is invalid
TXMC590_ERR_INVAL	Specified pointer is invalid.

3.2 Device Access Functions

3.2.1 txmc590ReadTemperature

NAME

txmc590ReadTemperatue – Read temperature from specified channel

SYNOPSIS

```
TXMC590_STATUS txmc590ReadTemperature
(
        TXMC590_HANDLE          hdl,
        unsigned int            channel,
        unsigned int            unit,
        unsigned int            decPlaces,
        unsigned int            immediateRead,
        int                     *value
)
```

DESCRIPTION

This function reads a temperature from the specified channel. The unit of the returned value can be chosen.

PARAMETERS

hdl

This argument specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

channel

This argument specifies the input channel which shall be read. Possible values are 0 up 15.

unit

This argument specifies the unit of the returned temperature value. The following values are defined:

Value	Description
TXMC590_UNIT_RAW	The value read from the channel will be returned. The unit is defined in the configuration table of the channel.
TXMC590_UNIT_CELSIUS	The temperature will be returned in degrees Celsius.
TXMC590_UNIT_FAHRENHEIT	The temperature will be returned in degrees Fahrenheit.
TXMC590_UNIT_KELVIN	The temperature will be returned in Kelvin.

decPlaces

This argument specified the decimal places of the returned value. This argument is not used if the argument unit is TXMC590_RAW.

The table below shows how the conversion works.

Temperature	Specified Unit	Decimal Places	Returned Value
20.1638°C	°C	0	20
20.1638°C	°C	1	202
20.1638°C	°C	2	2016
20.1638°C	°C	3	20164
20.1638°C	°C	5	2016380
20.1638°C	K	2	29331
20.1638°C	°F	2	6829

immediateRead

This argument specifies if the function should wait for a new value. This flag is only used if the channel is configured in clock mode (TXMC590_TRMODE_CLOCK_xxx).

Value	Description
TXMC590_RDMODE_WAIT	The function will wait until the current clock period expires and the value is returned.
TXMC590_RDMODE_IMMEDIATE	The function returns the last converted value. The read is asynchronous to the update cycle.

value

This argument returns the temperature in the specified unit or if the raw unit is specified the value is returned in the unit defined in the configuration table.

EXAMPLE

```
#include "txmc590api.h"

TXMC590_HANDLE  hdl;
TXMC590_STATUS  result;
int       temperature;
double    fTemperature;
int       decPlaces = 3;     /* number of decimal places */

/* Wait for a new value */
result = txmc590ReadTemperature( hdl,
                 1,
                 TXMC590_UNIT_CELSIUS,
                 decPlaces,
                 TXMC590_RDMODE_WAIT,
                 &temperature);
if (result != TXMC590_OK)
{
   /* handle error */
}
else
{
   fTemperature = temperature / pow(10, decPlaces);
   printf("Temperature (#1): %8.5f°C\n", fTemperature);
}
```

RETURNS

On success, TXMC590_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TXMC590_ERR_INVALID_HANDLE	The specified TXMC590_HANDLE is invalid
TXMC590_ERR_INVAL	Invalid pointer or value specified
TXMC590_ERR_INVALID_CHANNEL	Invalid channel number selected
TXMC590_ERR_INVALID_MODE	Channel not configured to read temperatures
TXMC590_ERR_INVALID_TABLE	Invalid value detected in correction table
TXMC590_ERR_BUSY	Channel is busy
TXMC590_ERR_TIMEOUT	Access timed out
TXMC590_ERR_CONFIG	Channel announced a configuration error
TXMC590_ERR_CHANNEL	Channel announced a channel error

3.2.2 txmc590ReadColdJunction

NAME

txmc590ReadColdJunction – Read the cold junction temperature

SYNOPSIS

```
TXMC590_STATUS txmc590ReadColdJunction
(
        TXMC590_HANDLE          hdl,
        unsigned int            channel,
        unsigned int            unit,
        unsigned int            decPlaces,
        int                     *value
)
```

DESCRIPTION

This function reads a cold junction temperature (local input of TXMC590). The unit of the returned value can be chosen.

PARAMETERS

hdl

This argument specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

channel

This argument specifies the cold junction channel that shall be read. The following values are valid:

Channel	Description
TXMC590_INTERNAL_CJ_SENSOR	Read the cold junction sensor locally mounted on the TXMC590.
TXMC590_EXTERNAL_CJ_SENSOR	Read the cold junction sensor externally connected to the TXMC590.

unit

This argument specifies the unit of the returned temperature value. The following values are defined:

Value	Description
TXMC590_UNIT_CELSIUS	The temperature will be returned in degrees Celsius.
TXMC590_UNIT_FAHRENHEIT	The temperature will be returned in degrees Fahrenheit.
TXMC590_UNIT_KELVIN	The temperature will be returned in Kelvin.
TXMC590_UNIT_RAW	Read raw value from TXMC590.

decPlaces

This argument specifies the decimal places of the returned value. This argument is not used if the argument unit is TXMC590_UNIT_RAW.

value

This argument returns the cold junction temperature in the specified unit.

EXAMPLE

```
#include "txmc590api.h"

TXMC590_HANDLE  hdl;
TXMC590_STATUS  result;
int       temperature;
double    fTemperature;
int       decPlaces = 3;     /* number of decimal places */

result = txmc590ReadColdJunction(hdl,
                TXMC590_INTERNAL_CJ_SENSOR,
                TXMC590_UNIT_CELSIUS,
                decPlaces,
                &temperature);
if (result != TXMC590_OK)
{
   /* handle error */
}
else
{
   fTemperature = temperature / pow(10, decPlaces);
   printf("CJ-Temperature): %8.5f°C\n", fTemperature);
}
```

RETURNS

On success, TXMC590_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TXMC590_ERR_INVALID_HANDLE	The specified TXMC590_HANDLE is invalid
TXMC590_ERR_INVAL	Invalid pointer or value specified
TXMC590_ERR_INVALID_CHANNEL	Invalid channel number selected
TXMC590_ERR_INVALID_MODE	Channel not configured

3.2.3 txmc590ReadStrain

NAME

txmc590ReadStrain – Read strain value from specified channel

SYNOPSIS

```
TXMC590_STATUS txmc590ReadStrain
(
        TXMC590_HANDLE          hdl,
        unsigned int            channel,
        unsigned int            decPlaces,
        unsigned int            immediateRead,
        int                     *value
)
```

DESCRIPTION

This function reads a strain value from the specified channel.

PARAMETERS

hdl

This argument specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

channel

This argument specifies the input channel which shall be read. Possible values are 0 up 15.

decPlaces

This argument specified the decimal places of the returned value.

immediateRead

This argument specifies if the function should wait for a new value. This flag is only used if the channel is configured in clock mode (TXMC590_TRMODE_CLOCK_xxx).

Value	Description
TXMC590_RDMODE_WAIT	The function will wait until the current clock period expires and the value is returned.
TXMC590_RDMODE_IMMEDIATE	The function returns the last converted value. The read is asynchronous to the update cycle.

value

This argument returns the strain value. The value unit is returned according the specification of the configuration table.

EXAMPLE

```
#include "txmc590api.h"

TXMC590_HANDLE  hdl;
TXMC590_STATUS  result;
int       strain;

/* Do not wait for conversion, read last available value */
result = txmc590ReadStrain(hdl, 2, 3, TXMC590_RDMODE_IMMEDIATE, &strain);
if (result != TXMC590_OK)
{
  /* handle error */
}
else
{
  printf("Strain value (x1000) (#2): %d \n", strain);
}
```

RETURNS

On success, TXMC590_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TXMC590_ERR_INVALID_HANDLE	The specified TXMC590_HANDLE is invalid
TXMC590_ERR_INVAL	Invalid pointer or value specified
TXMC590_ERR_INVALID_CHANNEL	Invalid channel number selected
TXMC590_ERR_INVALID_MODE	Channel not configured to read strain
TXMC590_ERR_INVALID_TABLE	Invalid value detected in correction table
TXMC590_ERR_BUSY	Channel is busy
TXMC590_ERR_TIMEOUT	Access timed out
TXMC590_ERR_CONFIG	Channel announced a configuration error
TXMC590_ERR_CHANNEL	Channel announced a channel error

3.2.4 txmc590ConfigureChannel

NAME

txmc590ConfigureChannel – Configure an input channel

SYNOPSIS

```
TXMC590_STATUS txmc590ConfigureChannel
(
        TXMC590_HANDLE          hdl,
        unsigned int            channel,
        unsigned int            correctionTableNo,
        unsigned int            conversionClockMode,
        unsigned short          conversionClockRate,
        unsigned int            coldJunctionSource
)
```

DESCRIPTION

This function configures the specified channel.

PARAMETERS

hdl

This argument specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

channel

This argument specifies the input channel which shall be read. Possible values are 0 up 15.

correctionTableNo

This argument specifies the correction table number that shall be used for the channel. The configuration table contains all information which is necessary for the input channel to work. For more detailed information about the content of the table see TXMC590-UM.

There are predefined tables for the most common probes (0…8 - Thermocouples, 9…14 - RTDs, 15 - Strain Gauges) and user defined tables (16…31), which can be loaded by the application for custom probes or special application.

conversionClockMode

This argument specifies the mode how conversions are initiated. The following modes are defined:

Value	Description
TXMC590_TRMODE_OFF	The channel is disabled
TXMC590_TRMODE_CONVERTONREAD	A new conversion will be started on a read access. The reading function will have to wait for completion of the conversion.
TXMC590_TRMODE_CLOCK_100US	The conversion will be started repeatedly with a defined rate. The rate is defined in conversionClockRate in steps of 100us. The reading function can use the value immediately.
TXMC590_TRMODE_CLOCK_100MS	The conversion will be started repeatedly with a defined rate. The rate is defined in conversionClockRate in steps of 1ms. The reading function can use the value immediately.
TXMC590_TRMODE_CLOCK_1S	The conversion will be started repeatedly with a defined rate. The rate is defined in conversionClockRate in steps of 1s. The reading function can use the value immediately.

conversionClockRate

This argument specifies the value of the conversion timer. The value is only used for conversionClockMode TXMC590_CLOCK_xxx.

The value is specified in units defined in conversionClockMode, e.g. a refresh cycle of 10ms will be generated with a value of 100 and conversionClockMode = TXMC590_TRMODE_CLOCK_100us.

coldJunctionSource

This argument specifies the channel that will be used to measure the cold junction temperature. This value will just be used for Thermocouple probes.

coldJunctionSource	Description
0..15	Use the appropriate channel as Cold Junction input. The channels must be configured before it is used as cold junction input.
TXMC590_INTERNAL_CJ_SENSOR	Use the cold junction sensor locally mounted on the TXMC590. The cold junctions channel must be configured before it is used.
TXMC590_EXTERNAL_CJ_SENSOR	Use the cold junction sensor externally connected to the TXMC590. The cold junctions channel must be configured before it is used.
TXMC590_NONE_CJ_SENSOR	Do not use a cold junction sensor. If this cold junction mode is selected, the temperature difference between hot and cold junction.

EXAMPLE

```
#include "txmc590api.h"

TXMC590_HANDLE   hdl;
TXMC590_STATUS   result;

/* Use channel 1, whith table 5 update repeately every 10 seconds */
/* use the TXMC590 Cold Junction Temperature */
result = txmc590ConfigureChannel (   hdl,
                 1,
                 5,
                 TXMC590_CLOCK_1S,
                 10,
                 TXMC590_INTERNAL_CJ_SENSOR);
if (result != TXMC590_OK)
{
   /* handle error */
}
```

RETURNS

On success, TXMC590_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TXMC590_ERR_INVALID_HANDLE	The specified TXMC590_HANDLE is invalid.
TXMC590_ERR_INVALID_CHANNEL	Invalid channel number
TXMC590_ERR_INVALID_TABLE	Invalid correction table number specified, or content of table is invalid
TXMC590_ERR_CONFIG	Channel announced a configuration error
TXMC590_ERR_CHANNEL	Channel announced a channel error
TXMC590_ERR_INVAL	Invalid parameter pointer or value
TXMC590_ERR_TIMEOUT	Configuration timed out, configuration did not complete

3.2.5 txmc590ConfigureColdJunction

NAME

txmc590ConfigureColdJunction – Configure the TXMC590 Cold Junction channel

SYNOPSIS

```
TXMC590_STATUS txmc590ConfigureColdJunction
(
    TXMC590_HANDLE      hdl,
    unsigned int        channel,
    unsigned int        conversionClockMode,
    unsigned short      conversionClockRate
)
```

DESCRIPTION

This function configures the specified cold junction channel.

PARAMETERS

hdl

This argument specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

channel

This argument specifies the cold junction channel that shall be configured. The following values are valid:

Channel	Description
TXMC590_INTERNAL_CJ_SENSOR	Configure the cold junction sensor locally mounted on the TXMC590.
TXMC590_EXTERNAL_CJ_SENSOR	Configure the cold junction sensor externally connected to the TXMC590.

conversionClockMode

This argument specifies the mode how conversions are initiated. The following modes are defined.

Value	Description
TXMC590_TRMODE_OFF	The channel is disabled
TXMC590_TRMODE_CLOCK_100US	The conversion will be started repeatedly with a defined rate. The rate is defined in conversionClockRate in steps of 100us. The reading function can use the value immediately.
TXMC590_TRMODE_CLOCK_100MS	The conversion will be started repeatedly with a defined rate. The rate is defined in conversionClockRate in steps of 1ms. The reading function can use the value immediately.
TXMC590_TRMODE_CLOCK_1S	The conversion will be started repeatedly with a defined rate. The rate is defined in conversionClockRate in steps of 1s. The reading function can use the value immediately.

conversionClockRate

This argument specifies the value of the conversion timer.

The value is specified in units defined in conversionClockMode, e.g. a refresh cycle of 10ms will be generated with a value of 100 and conversionClockMode = TXMC590_TRMODE_CLOCK_100us.

EXAMPLE

```
#include "txmc590api.h"

TXMC590_HANDLE  hdl;
TXMC590_STATUS  result;

/* Use internal cold junction sensor, update every 10 seconds */
result = txmc590ConfigureColdJunction (hdl,
                  TXMC590_INTERNAL_CJ_SENSOR,
                  TXMC590_TRMODE_CLOCK_1S,
                  10);
if (result != TXMC590_OK)
{
   /* handle error */
}
```

RETURNS

On success, TXMC590_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TXMC590_ERR_INVALID_HANDLE	The specified TXMC590_HANDLE is invalid.
TXMC590_ERR_INVALID_CHANNEL	Invalid channel number
TXMC590_ERR_INVAL	Invalid parameter pointer or value

3.2.6 txmc590Calibrate

NAME

txmc590Calibrate – Calibrate channel – Execute internal ADC calibration cycle

SYNOPSIS

```
TXMC590_STATUS txmc590Calibrate
(
      TXMC590_HANDLE          hdl,
      unsigned int            channel
)
```

DESCRIPTION

This function executes an ADC calibration cycle on the specified channel.

PARAMETERS

hdl

This argument specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

channel

This argument specifies the input channel which shall be read. Possible values are 0 up 15.

EXAMPLE

```
#include "txmc590api.h"

TXMC590_HANDLE   hdl;
TXMC590_STATUS   result;

result = txmc590Calibrate(hdl, 2);
if (result != TXMC590_OK)
{
   /* handle error */
}
```

RETURNS

On success, TXMC590_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TXMC590_ERR_INVALID_HANDLE	The specified TXMC590_HANDLE is invalid.
TXMC590_ERR_INVALID_CHANNEL	Invalid channel number
TXMC590_ERR_CONFIG	Channel announced a configuration error
TXMC590_ERR_CHANNEL	Channel announced a channel error

3.2.7 txmc590LoadUserTable

NAME

txmc590LoadUserTable – Load a custom defined table

SYNOPSIS

```
TXMC590_STATUS txmc590UserTable
(
        TXMC590_HANDLE          hdl,
        unsigned int            tableNo,
        char                    *fileName
)
```

DESCRIPTION

This function loads a new table into a specified custom table space. The table allows to use custom and application adapted tables, e.g. for special probes or to use a more detailed resolution in the target temperature range.

For more detailed information about the content of the table see TXMC590-UM.

PARAMETERS

hdl

This argument specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

tableNo

This argument specifies the correction table number that shall be loaded. Only user definable tables (16…31) are selectable.

fileName

This argument specifies file name (path) of the table file.

EXAMPLE

```
#include "txmc590api.h"

TXMC590_HANDLE  hdl;
TXMC590_STATUS  result;

result = txmc590LoadUserTable (hdl, 18, "./newTable.hex");
if (result != TXMC590_OK)
{
   /* handle error */
}
```

RETURNS

On success, TXMC590_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TXMC590_ERR_INVALID_HANDLE	The specified TXMC590_HANDLE is invalid.
TXMC590_ERR_INVAL	Invalid parameter pointer or value
TXMC590_ERR_INVFILE	Invalid file format
TXMC590_ERR_NOMEM	Cannot allocate memory
TXMC590_ERR_INVALID_TABLE	Invalid correction table number specified
TXMC590_ERR_NOFILE	Can't access file
TXMC590_ERR_TIMEOUT	Loading table into or programming table from correction table space failed

4 Diagnostic

If the TXMC590 does not work properly, it is helpful to get some status information from the driver respective kernel. To get debug output from the driver, enable the following symbols in '*txmc590.c*' by replacing "#undef" with "#define":

```
#define DEBUG_TXMC590
```

The Linux */proc* file system provides information about kernel, resources, driver, devices, and so on. The following screen dumps display information of a correct running TXMC590 driver (see also the proc man pages).

```
# tail –f /var/log/messages
<log-time> linux kernel: TEWS TECHNOLOGIES - TXMC590 Device Driver: version
<version> (<version date>)
<log-time> linux kernel: TXMC590: Probe new device (vendor=0x1498, device=0x924E)
…
# lspci -v
…
03:00.0 Signal processing controller: TEWS Technologies GmbH Device 924e
        Subsystem: TEWS Technologies GmbH Device 900a
        Flags: fast devsel, IRQ 18
        Memory at feaff000 (32-bit, non-prefetchable) [size=4K]
        Memory at feafe000 (32-bit, non-prefetchable) [size=4K]
        Memory at feae0000 (32-bit, non-prefetchable) [size=64K]
        Capabilities: [40] Power Management version 3
        Capabilities: [48] MSI: Enable- Count=1/1 Maskable- 64bit+
        Capabilities: [60] Express Endpoint, MSI 00
        Capabilities: [100] Device Serial Number 00-00-00-00-00-00-00-00
        Kernel driver in use: TEWS TECHNOLOGIES - TXMC590 Device Driver
        Kernel modules: txmc590drv
…

# cat /proc/devices
Character devices:
…
226 drm
240 txmc590drv
…
```

```
# cat /proc/interrupts
           CPU0       CPU1       CPU2       CPU3
  0:        110          0          0          0   IO-APIC   2-edge      timer
  1:         13          0          0          0   IO-APIC   1-edge      i8042
  7:          0          0          0          0   IO-APIC   7-edge      parport0
  8:          0          1          0          0   IO-APIC   8-edge      rtc0
  9:          0          0          0          0   IO-APIC   9-fasteoi   acpi
 12:          0          0          0        163   IO-APIC  12-edge      i8042
 16:          0          0          0          0   IO-APIC  16-fasteoi   uhci_hcd:usb5
 18:          0          0       4927          0   IO-APIC  18-fasteoi   ata_piix, TXMC590
 19:          0     229696          0          0   IO-APIC  19-fasteoi   enp5s0
 22:          0          0          0      68620   IO-APIC  22-fasteoi   ata_piix, i801_smbus
…
```