API Documentation (short)

Product / Project: TELID® 3xx native driver

Customer / Project Code: -
Product: MICROSENSYS TELID®3/iID®3000 reader interfaces
Product Code: -
Revision: **16.0.0.0**
Date: **2018-03-27**

API - Definition for PC Windows®

This document describes the extension to the microsensys's API of TELID® 3xx native driver for MICROSOFT Windows® based PC systems, in order to support the TELID® 343.02 product line. TELID® 3xx native driver is a Borland Delphi dll based on the native iID®3000 (mobile) driver engine, up from Ver10.1.

The TELID® 3xx histogram functionality is not supported by this native driver, therefore please see the .Net class library TELID® (mobile) driver engine.

Library name:	**TELID3xx_Native.dll**
Tested devices:	Windows®10 x86, Windows®7 x64
Drivername:	**iiddrv30_pro.dll**
Supported host interface:	RS232 serial, USB
Release date:	2018-03-27

Charset: Windows® ANSI charset

Content

1. Short history

This chapter includes a short history of modifications of both iiddrv30_pro.dll and TELID3xx native library.

Date	Reason	Modification	Release Date / Version	FileName
2007-02-26	- first release with sample Delphi application based on TELID® 311 and TELID® 312 up to Ver 2		2007-02-26	TELID3xx_native.dll
2007-06	- added support for TELID calibration		2007-09-03	TELID3xx_native.dll, V10.5.0.1
2007-09	- upgrade to base driver iiddrv30_pro.dll V0x10.0xB		2007-09-03	TELID3xx_native.dll, V10.5.0.3
2007-10-15	- correction of TELID_ReadTemperaturProtocol, error during check of last datasets	TELID_ReadTemperaturProtocol – internal modification	2007-10-15	TELID3xx_native.dll, V10.6.0.5
2007-10-29	-implementation of TELID® 311 125° functionality	TELID_ReadTemperatureProtocol – internal modification TELID_ProgramParameters – internal modification	2007-11-07	TELID3xx_native.dll, V10.7.0.6
2007-11-07	- correction of second mode for TELID® 125°	TELID_ProgramParameters – internal modification TELID_ReadParameters – internal modification	2007-11-08	TELID3xx_native.dll, V10.8.0.2
2008-04-17	- added new TELID® 311 and TELID® 312 extension (supported TELID: version > 1)	TELID_ProgramParameters - internal modification	2008-04-17	TELID3xx_native.dll, V10.8.1.1
2008-12-04	- correction for analyzation of datasets in roll over mode - round TELIDStartTime to full minutes while programming	TELID_ProgramParameters - internal modification TELID_ReadTemperatureProtocol	2008-12-04	TELID3xx_native.dll, V10.8.4.2
2009-04-28	- correction of programming time if a TELID is programmed outside of TELIDsoft + compatible	TELID_ReadParameters - internal modification	2009-04-28	TELID3xx_native.dll, V10.8.6.1
2010-07-30	Optional programming with sampling rate 5sec possible – **please note, that frequently usage of this mode will lower the products life time because of enhanced energy requirement**	TELID_ProgramParameters - Internal modification	2010-07-30	TELID3xx_native.dll, V11.0.1.2
2012-02-06	Optional programming with sampling rate 1sec possible – **please note, that frequently usage of this mode will lower the products life time because of enhanced energy requirement**	TELID_ProgramParameters - Internal modification	2012-02-06	

2016-07-26	- x64 platform adaptions	- diverse	2016-07-26	TELID3xx_native.dll, V13.0.0.0
2017-05-05	- internal modifications for sample counter	- diverse	2017-05-05	TELID3xx_native.dll, V13.1.0.0
2017-08-03	- added support for temperature/pressure data logger TELID343.02	- diverse	2017-08-03	TELID3xx_native.dll, V14.0.0.0
2018-03-27	- added Stop Time parameter for TELID®343.02 - modified read pressure log functions	- diverse	2018-03-27	TELID3xx_native.dll, V16.0.0.0

2. Implementation of TELID®343 data logger

Up from version 14.x the existing TELID® native library has been extended to support the TELID® 343.02 Pressure logger product line. This means, that current implementations of TELID® 3xx native library based on library versions below 14.0.0.0 have to be checked for compatibility, looking at the sample code NativeTest_TELID3xx. The modifications are as follows:

1. Defining the new data structures and function prototypes (See below).
2. Defining a new constant to support the new product codes. Currently there is only a product code (0x50), but in the sample application a variable has been defined as set of bytes, in order to accommodate future product enhancements.

Up from version 16.x, the interface of the sample application changes to accommodate new functions and parameters needed for handling the TELID® 343.02. On the other hand, clicking the "*Read TEMPERATURE PROTOCOL from TELID*"-Button in NativeTest_TELID3xx causes calling the corresponding functions, depending on *TELID_ProductCode.* Please note, that reading the product code is done when calling *TELID_ReadParameters.*

3. Common functions

This chapter includes a description of functions used for communication with the host port and reader interface.

```
procedure Set_Port_Type(Porttype_:integer);stdcall;
```

This procedure sets the port type for iID® driver functions. See actual API documentation iID® (mobile) driver engine for more details.

```
procedure Set_Port_Name(Portname_:string);stdcall;
```

This procedure sets the port name for iID® driver functions. See actual API documentation iID® (mobile) driver engine for more details.

```
procedure Set_Interface_Type(InterfaceType_:integer);stdcall;
```

This procedure sets the interface type for iID® driver functions. See actual API documentation iID® (mobile) driver engine for more details.

```
procedure Reader_OpenInterface;stdcall;
```

This procedure tries to establish reader interface communication on the selected port. Following function calls are required prior calling Reader_OpenInterface():

- `procedure Set_Port_Type(Porttype_:integer);stdcall;`
- `procedure Set_Port_Name(Portname_:string);stdcall;`
- `procedure Set_Interface_Type(InterfaceType_:integer);stdcall;`

```
procedure Reader_CloseInterface;stdcall;
```

This procedure tries to close reader interface communication on the selected port.

```
procedure Reader_GetState;stdcall;
```

Reader_GetState is designed to check Reader- and Port communication. Results can be obtained using

- `Get_Port_Initialized`
- `Get_Reader_Connected`
- `Get_Reader_Id`

```
function Get_Port_Initialized:integer;stdcall;
```

This function returns 1, if communication port is open, otherwise 0. See *Reader_OpenInterface* and *Reader_CloseInterface* for more information.

```
function Get_Reader_Connected:integer;stdcall;
```

This function returns 1, if communication with reader is established, otherwise 0. See *Reader_OpenInterface, Reader_CloseInterface, Reader_GetState* for more information.

```
function Get_Reader_Id:integer;stdcall;
```

See *Reader_GetState, Reader_Read_Inteface_Id* for more information.

This function returns the internal ID number of a connected reader.

```
function
Reader_Read_Inteface_Id(PReaderId:PInteger;PReadArray:PByteArray):integer;s
tdcall;
```

This function reads ID-number of connected reader interface. Following variables are filled by this function:

```
PByteArray          PReadArray;
PInteger            PReaderId;
```

4. TELID® functions

This chapter includes TELID® system specific functions.

```
function TELID_ReadROCode(PData:PByteArray;
P_TELID_ID:PInteger):Integer;stdcall;
```

This function reads ReadOnly-Code of TELID device. The code with length 8byte is placed into the variable *PData, P_TELID_ID* contains the internal TEILD ID number. See Error-CodeTable for return values.

```
function TELID_ReadRemarks(PFrom,PTo,PRemark:PByteArray):Integer;stdcall;
```

This function reads the logistic information of TELID device. See Error-CodeTable for return values. Following variables are filled by this function:

```
PByteArray        TELID_From;  (PbyteArray^[0]=length+up to 16byte data)
PByteArray        TELID_To;  (PbyteArray^[0]=length+up to 16byte data)
PByteArray        TELID_Remark;  (PbyteArray^[0]=length+up to 32byte data)
```

```
function
TELID_ProgramRemarks(Array1,Array2,Array3:PByteArray):Integer;stdcall;
```

This function sets the logistic information to TELID device. See Error-CodeTable for return values. Following variables are filled by this function:

```
PByteArray        TELID_From;  (PbyteArray^[0]=length+up to 16byte data)
PByteArray        TELID_To;  (PbyteArray^[0]=length+up to 16byte data)
PByteArray        TELID_Remark;  (PbyteArray^[0]=length+up to 32byte data)
```

5. TELID® 311 specific functions

This chapter includes TELID® 311 temperature data logger specific functions.

```
function TELID_ReadParameters(
            PT_ID : Pinteger;
            PT_Min, PT_Max : PDouble;
            PT_ProductCode, PT_Version  : Pbyte;
            PT_Nr_Of_DataSets, PT_Max_Nr_Of_DataSets,
           PT_MemCount  : Pinteger;
            PT_Mode, PT_Mode_Advanced, PT_Status_Fail : Pbyte;
            PT_HTModeEnable, PT_HistogramEnable, PT_LimitWatchEnable,
           PT_OutOfLimit : Pbyte;
            PT_StartTime, PT_ProgramTime : PDateTime;
            PT_SampleTime_Min, PT_SampleTime_Sec : Pinteger;
            PT_LED1, PT_LED2, PT_LED3 : PByte):Integer;stdcall;
```

This function reads start-up parameters from the TELID device. Following variables are filled by this function:

PT_ID – this variable holds the ID-string of TELID® device.

PT_Min, PT_Max – these variables contain limit information of the TELID® device. Limit information is not used during the measurement process, only for evaluation.

PT_ProductCode – This variable holds the product code of the TELID® device. Up from version 16.0.0.0, only TELID® 311 product codes are supported in this function.

PT_Version – This variable holds the product version of the TELID® device.

PT_Nr_Of_DataSets – This variable holds the number of datasets collected in the TELID® device. The value is necessary for information and for interpretation of the data table.

PT_Max_Nr_Of_DataSets – This variable sets/holds the maximum number of datasets, which can be collected in the TELID® device and should be a multiple of 64.

PT_Mode – This variable contains the operation mode of the TELID® device.

Bit	Value = 0	Value = 1
0	**Device in Sleep Mode**	**Device in Run Mode**
1	**Stop Full Mode**	**Roll Over Mode**
2	**Reserved**	**Reserved**
3	**Reserved**	**Reserved**
4	**Reserved**	**Reserved**
5	**Reserved**	**Reserved**
6	**Device in measurement mode**	**Device in start delay**
7	**Free memory available**	**Memory full**

PT_MemCount – This variable holds the number of memory turns, if TELID is in Roll Over mode.

PT_Mode_Advanced – Internal use only

PT_Status_Fail – This variable should be evaluated within the application software for error detection, values <>0 generally define an error. For further details see the TELID® 3xx hardware documentations.

Bit	Description
0	**BATT-LO_0**
1	**BATT-LO_1**
2	**Internal use**
3	**Internal use**
4	**Internal use**
5	**Internal use**
6	**Internal use**
7	**Internal use**

PT_HTModeEnable (not supported) – This variable enables/disables the high temperature mode when programming TELID® devices. Leaving this variable “0” disables HT functionality, value “1” enables HT functionality depending on device type. Using this mode is not supported together with TELID_Mode=2 (Roll Over Mode).
PT_LimitWatchEnable – This variable specifies/shows, if the TELID® device’s LimitWatch is enabled. Leaving this variable “0” disables LimitWatch functionality, values different from “0” enable LimitWatch functionality depending on device type.
PT_HistogramEnable – This variable specifies/shows, if the TELID® device’s histogram function is enabled. Leaving this variable “0” disables histogram functionality, values different from “0” enable histogram functionality depending on device type.
PT_OutOfLimit – This variable shows, if the TELID® device detected a temperature limit overflow. Using this value, the user can decide of reading the whole data memory. Value 0 means no detected limit overflow, value 1 shows at minimum one limit overflow. This variable should only be used, when TELID LimitWatch is enabled.
PT_StartTime – This variable holds/sets the start time for TELID® operations.
PT_ProgramTime – This variable holds the program time for TELID® operations.
PT_SampleTime_Min, PT_SampleTime_Sec – These variables hold/set the sample time for TELID® operations. The value of *PT_SampleTime_Min* should be between 1 and 255, TELID_SampleTime_Sec has to be set 0. Special devices support a sample time < 1 min, therefore please set TELID_SampleTime_Min=0, TELID_SampleTime_Sec 15..59.
PT_LED1, PT_LED2, PT_LED3 – This variable contains functionality information of up to three LEDs attached to the TELID® device. Leaving these variables “0” disables LED functionality, values different from “0” enable LED functionality depending on device type.

```
function TELID_ProgramParameters(
            PT_ID : PInteger;
            T_Min, T_Max : Double;
            PT_ProductCode, PT_Version  : PByte;
            T_Max_Nr_Of_DataSets  : integer;
            T_Mode, T_Service_Mode : byte;
            T_HTModeEnable, T_HistogramEnable, T_LimitWatchEnable : byte;
            PT_StartTime, PT_ProgramTime : PDateTime;
            T_SampleTime_Min, T_SampleTime_Sec : integer;
            T_LED1, T_LED2, T_LED3 : byte):Integer;stdcall;
```

This function sets the start-up parameters of the TELID device. See Error-CodeTable for return values. Following variables should be provided when calling the function:
PT_ID – This variable holds the ID-string of TELID® device.
T_Min, T_Max – These variables contain limit information of the TELID® device. Limit information is not used during the measurement process, only for evaluation.
PT_ProductCode – This variable holds the product code of the TELID® device
PT_Version – This variable holds the product version of the TELID® device.
T_Max_Nr_Of_DataSets – This variable sets/holds the maximum number of datasets, which can be collected in the TELID® device and should be a multiple of 64.
T_Mode – This variable contains the operation mode of the TELID® device. Following modes are supported:

Bit	Value = 0	Value = 1
0	**Sleep Mode**	**Run Mode**
1	**Stop Full Mode**	**Roll Over Mode**
2	**Reserved**	**Reserved**
3	**Reserved**	**Reserved**
4	**Reserved**	**Reserved**
5	**Reserved**	**Reserved**
6	**Reserved**	**Reserved**
7	**Reserved**	**Reserved**

T_Service_Mode – Internal use only
T_HTModeEnable - This variable enables/disables the high temperature mode when programming TELID® devices. Leaving this variable "0" disables HT functionality, value "1" enables HT functionality depending on device type. Using this mode is not supported together with TELID_Mode=2 (Roll Over Mode), measurement interval is always 1min.
See *TELID_ReadParameters* and *TELID_ProgramParameters* for more details.
T_LimitWatchEnable - This variable specifies/shows, if the TELID® device's LimitWatch is enabled. Leaving this variable "0" disables LimitWatch functionality, values different from "0" enable LimitWatch functionality depending on device type.
T_HistogramEnable - This variable specifies/shows, if the TELID® device's histogram function is enabled. Leaving this variable "0" disables histogram functionality, values different from "0" enable histogram functionality depending on device type.
PT_StartTime - This variable holds/sets the start time for TELID® operations.
PT_ProgramTime - This variable holds the program time for TELID® operations.
T_SampleTime_Min, T_SampleTime_Sec - These variables hold/set the sample time for TELID® operations. The time should be a multiple of one minute (between 1 and 255), TELID_SampleTime_Sec has to be set 0. Special devices support a sample time < 1 min, therefore please set TELID_SampleTime_Min=0, TELID_SampleTime_Sec 1..59.
T_LED1, T_LED2, T_LED3 - This variable contains functionality information of up to three LEDs attached to the TELID® device. Leaving these variables "0" disables LED functionality, values different from "0" enable LED functionality depending on device type.

```
function TELID_ReadTemperatureProtocol(PNrOfReadDatasets:PInteger):
Integer;stdcall;
```

This function reads the temperature protocol from the TELID device. Calling TELID_ReadParameters is required before using this function. See Error-CodeTable for return values. Following variables are filled by this function:
PNrOfReadDatasets – Number of datasets already read from device;

```
function
TELID_GetTemperatureData(NrOfData:Integer;PData:PTELID_Temperature_record):
Integer;stdcall;
```

This function loads one selected temperature dataset from the library into the application. Prior call of *TELID_ReadTemperatureProtocol* is required. For definition of the *State* byte see *TELID_ReadParameters/TELID_Status_Fail*.

NrOfData – Number of dataset to load from library
Pdata – Pointer to record to be filled with time/temperature dataset

```
TTELID_Temperature_Record =
      record
        Time:        TDateTime;
        Temperature: Double;
        State:       Integer;
      end;
PTELID_Temperature_Record = ^TTELID_Temperature_Record;
```

6. TELID® 343 specific functions

This chapter includes TELID® 343 temperature/pressure data logger specific functions.

```
function TELID_ReadPressureParameters(
            PT_ID : Pinteger;
            PT_Min, PT_Max : PDouble;
            PT_ProductCode, PT_Version  : Pbyte;
            PT_Nr_Of_DataSets, PT_MemCount  : Pinteger;
            PT_Mode, PT_Mode_Advanced, PT_Status_Fail : Pbyte;
            PT_HTModeEnable, PT_HistogramEnable, PT_LimitWatchEnable,
           PT_OutOfLimit : Pbyte;
            PT_StartTime, PT_StopTime, PT_ProgramTime : PDateTime;
            PT_SampleTime_Min, PT_SampleTime_Sec : Pinteger;
            PT_LED1, PT_LED2, PT_LED3 : PByte):Integer;stdcall;
```

This function reads start up parameters from the TELID® 343.02 device. Following variables are filled by this function:

PT_ID – this variable holds the ID-string of TELID® device.

PT_Min, PT_Max – these variables contain limit information of the TELID® device. Limit information is not used during the measurement process, only for evaluation.

PT_ProductCode – This variable holds the product code of the TELID® device. This function only supports TELID® 343.02 product codes

PT_Version – This variable holds the product version of the TELID® device.

PT_Nr_Of_DataSets – This variable holds the number of datasets collected in the TELID® device. The value is necessary for information and for interpretation of the data table.

PT_Mode – This variable contains the operation mode of the TELID® device.

Bit	Value = 0	Value = 1
0	**Device in Sleep Mode**	**Device in Run Mode**
1	**Reserved**	**Reserved**
2	**Reserved**	**Reserved**
3	**Reserved**	**Reserved**
4	**Reserved**	**Reserved**
5	**Reserved**	**Reserved**
6	**Device in measurement mode**	**Device in start delay**
7	**Reserved**	**Reserved**

PT_MemCount – This variable holds the number of memory turns, if TELID is in Roll Over mode.

PT_Mode_Advanced – Internal use only

PT_Status_Fail – This variable should be evaluated within the application software for error detection, values <>0 generally define an error. For further details see the TELID® 3xx hardware documentations.

PT_HTModeEnable (not supported) – This variable enables/disables the high temperature mode when programming TELID® devices. Leaving this variable “0” disables HT functionality, value “1” enables HT functionality depending on device type. Using this mode is not supported together with TELID_Mode=2 (Roll Over Mode).

PT_LimitWatchEnable (not supported) – This variable specifies/shows, if the TELID® device’s LimitWatch is enabled. Leaving this variable “0” disables LimitWatch functionality, values different from “0” enable LimitWatch functionality depending on device type.

PT_HistogramEnable (not supported) – This variable specifies/shows, if the TELID® device’s histogram function is enabled. Leaving this variable “0” disables histogram functionality, values different from “0” enable histogram functionality depending on device type.

PT_OutOfLimit (not supported) – This variable shows, if the TELID® device detected a temperature limit overflow. Using this value, the user can decide of reading the whole data memory. Value 0 means no detected limit overflow, value 1 shows at minimum one limit overflow. This variable should only be used, when TELID LimitWatch is enabled.

PT_StartTime – This variable holds/sets the start time for TELID® operations.
PT_StopTime – This variable holds/sets the stop time for TELID® operations.
PT_ProgramTime – This variable holds the program time for TELID® operations.
PT_SampleTime_Min, PT_SampleTime_Sec – These variables hold/set the sample time for TELID® operations. Special devices support a sample time < 1 min, for this just set TELID_SampleTime_Min=0, TELID_SampleTime_Sec 15..59.
PT_LED1, PT_LED2, PT_LED3 (not supported) – This variable contains functionality information of up to three LEDs attached to the TELID® device. Leaving these variables "0" disables LED functionality, values different from "0" enable LED functionality depending on device type.

```
function TELID_ProgramParameters(
            PT_ID : PInteger;
            T_Min, T_Max : Double;
            PT_ProductCode, PT_Version  : PByte;
            T_Mode, T_Service_Mode : byte;
            T_HTModeEnable, T_HistogramEnable, T_LimitWatchEnable : byte;
            PT_StartTime, PT_StopTime, PT_ProgramTime : PDateTime;
            T_SampleTime_Min, T_SampleTime_Sec : integer;
            T_LED1, T_LED2, T_LED3 : byte):Integer;stdcall;
```

This function sets the start-up parameters of the TELID device. See Error-CodeTable for return values. Following variables are filled by this function:
PT_ID – This variable holds the ID-string of TELID® device.
T_Min, T_Max – These variables contain limit information of the TELID® device. Limit information is not used during the measurement process, only for evaluation.
PT_ProductCode – This variable holds the product code of the TELID® device
PT_Version – This variable holds the product version of the TELID® device.
T_Mode – This variable contains the operation mode of the TELID® device. Following modes are supported:

Bit	Value = 0	Value = 1
0	**Sleep Mode**	**Run Mode**
1	**Reserved**	**Reserved**
2	**Reserved**	**Reserved**
3	**Reserved**	**Reserved**
4	**Reserved**	**Reserved**
5	**Reserved**	**Reserved**
6	**Reserved**	**Reserved**
7	**Reserved**	**Reserved**

T_Service_Mode – Internal use only
T_HTModeEnable (not supported) – This variable enables/disables the high temperature mode when programming TELID® devices. Leaving this variable "0" disables HT functionality, value "1" enables HT functionality depending on device type. Using this mode is not supported together with TELID_Mode=2 (Roll Over Mode), measurement interval is always 1min.
See *TELID_ReadParameters* and *TELID_ProgramParameters* for more details.
T_LimitWatchEnable (not supported) – This variable specifies/shows, if the TELID® device's LimitWatch is enabled. Leaving this variable "0" disables LimitWatch functionality, values different from "0" enable LimitWatch functionality depending on device type.
T_HistogramEnable (not supported) – This variable specifies/shows, if the TELID® device's histogram function is enabled. Leaving this variable "0" disables histogram functionality, values different from "0" enable histogram functionality depending on device type.
PT_StartTime – This variable holds/sets the start time for TELID® operations.
PT_StopTime – This variable holds/sets the stop time for TELID® operations.
PT_ProgramTime – This variable holds the program time for TELID® operations.
T_SampleTime_Min, T_SampleTime_Sec – These variables hold/set the sample time for TELID® operations. Special devices support a sample time < 1 min, for this just set TELID_SampleTime_Min=0, TELID_SampleTime_Sec 1..59.

T_LED1, T_LED2, T_LED3 (not supported) – This variable contains functionality information of up to three LEDs attached to the TELID® device. Leaving these variables "0" disables LED functionality, values different from "0" enable LED functionality depending on device type.

```
function TELID_ReadPressureProtocol(PNrOfReadDatasets:PInteger;
PNrOfReadMeasurements:PInteger): Integer;stdcall;
```

This function reads the pressure protocol from the TELID device. Calling TELID_ReadParameters is required before using this function. See Error-CodeTable for return values. Following variables are filled by this function:

PNrOfReadDatasets – Number of datasets already read from device

PNrOfReadMeasurements – Number of measurements already read from device

```
Function
TELID_GetPressureData(NrOfData:Integer;PData:PTELID_Pressure_record):
Integer;stdcall;
```

This function loads one selected pressure measurement data from the library into the application. Prior call of *TELID_ReadPressureProtocol* is required.

NrOfData – Number of dataset to load from library

Pdata – Pointer to record to be filled with pressure measurement

```
TTELID_Pressure_Record =
      record
        Time:           TDateTime;
        Status:         Integer;
        Pressure:       Double;
        Temperature:    Double;
      end;
PTELID_Pressure_record = ^TTELID_Pressure_record;
```

7. Error - Codetable

State	Error
0x00	no error
0x01	protocol read and more datasets available
0x08	identifiers do not match
0x11	range error
0x23	TAG-error
0x24	no TAG near antenna, no data received
0x26	OP-CODE unknown
0x28	protocol failure
0x29	unknown TAG instruction
0x2A	unknown TAG-error
0x2B	error writing to TAG
0x2C	error reading from TAG
0x2E	error control-reading TAG
0x2F	wrong data control-reading TAG (RAW)
0x30	error in CRC-CheckSum
Additional Errors	
0x3F	Communication or port error
0x4E	programming error, format convert error
0x4F	unknown/not supported option detected
0xFF	general communication or driver error

All not used / internal variables have to be zero-set.