



Description of the S30c DLL

for digital transmitters and autonomous data loggers from KELLER

1	Introduction.....	1
1.1	Description of the software driver (DLL)	1
1.1.1	General.....	1
1.1.2	The functions of the DLL	2
1.1.2.1	Port functions.....	3
1.1.2.2	Echo function.....	3
1.1.2.3	Protocol functions	4
1.2	Changes	5
1.3	Support.....	5

1 Introduction

This document describes all procedures and functions of the S30c.dll which are available for your application programs. Using these procedures and functions performed by the DLL enables you to write custom Windows applications (Windows XP, Windows 7 and Windows 8.1) in Delphi, Visual Basic, C++ Builder or any other Windows utility program for applications that support DLL. The DLL is available as 32bit (s30c.dll) and 64bit (s30c_64bit.dll) version.

Find further in this document a complete list of all functions and methods that are exported by the S30c.dll. KELLER offers a number of program examples which allows you to insight into the way how you can develop your own applications. You find those examples and additional the communication protocols to all digital products of KELLER, either on the KELLER Software CD or on our webpage: <http://www.keller-druck.com/>.

1.1 Description of the software driver (DLL)

1.1.1 General

The available DLL s30c.dll and s30c_64bit.dll have been tested on the Windows XP, Windows7 64bit and Windows 8.1 64bit operating systems.

Examples of the use of this DLL are available for the following programming languages:

- LabVIEW
- C++
- Delphi
- .Net
- VB
- VBA



The call convention stdcall is used for assigning the parameters to the functions. This means that:

- all parameters are passed via the stack
- the parameter furthest to the right is calculated and passed first, the parameter furthest to the left is calculated and passed last
- the function itself deletes the parameters from the stack

As the declarations for the functions presented below show, many variables are declared with the prefixed word *var*. This means that these variables are passed as pointers and not as values.

The following types are used:

Type	Range	Format
Byte	0..255	8-bit without sign
Word	0..65535	16-bit without sign
Smallint	-32768..32767	16-bit with sign
Longint	-2147483648.. 2147483647	32-bit with sign
Pbyte		Pointer to byte
Single	+/- 1.5x10 ⁻⁴⁵ ..3.4x10 ³⁸	32-bit

1.1.2 The functions of the DLL

Each function has a return value indicating if the call has been successfully executed or not. Please find in the table below all possible return values. The returned parameters are valid only, when any function call returns with "RS_OK". The parameter values mustn't be processed when function result differs from "RS_OK".

Return value		Description
RS_OK	0	Function successfully executed; return parameters are valid
RS_EX1	1	Function successfully executed; but exception error 1 has occurred
RS_EX2	2	Function successfully executed; but exception error 2 has occurred
RS_EX3	3	Function successfully executed; but exception error 3 has occurred
RS_EX32	32	Function successfully executed; but exception error 32 has occurred
RS_BROADCAST	100	Broadcast
RS_ERROR	-1	General error
RS_TXERROR	-2	Transmit error
RS_RXERROR	-3	Receive error in UART
RS_TIMEOUT	-4	No data or insufficient data received
RS_BADDATA	-5	Data erroneous (e.g. CRC16 erroneous)



1.1.2.1 Port functions

The devices are connected to the PC via a serial interface. The port functions serve to open and close this interface. The standard setting should be used for the timeout time (Timeout = 0). When the desired port has been successfully opened, the **OpenComPort** function returns the value RS_OK, otherwise RS_ERROR. An open port is closed automatically on ending the program.

The **OpenComExt** function allows to specify baud rate and parity. Please note factory default setting for baud rate is 9600baud. Series 30 pressure transmitters with firmware 5.20 or newer support faster communication with 115'200 baud as well. Please use the Keller ControlCenterSeries30 software to configure the device accordingly.

The ControlCenterSeries30 software can be downloaded here:

<http://www.keller-druck2.ch/swupdate/InstallerControlCenterSeries30/InstallerControlCenterSeries30.zip>

As a standard setting, **no** parity is used (none). This results in a data format of 10 bits per byte. If parity is active, the data format is 11 bits per byte.

```
function OpenComPort( intPort, intTimeout: Smallint ): Smallint; stdcall; export;

function OpenComExt( intPort, intTimeout: Smallint; longBaud: Longint; intParity: Smallint
): Smallint; stdcall; export;

intParity: 0: no parity bit (Standard), 1: odd parity bit, 2: even parity bit
longBaud: 9600 for 9600 baud, 115'200 for 115'200 baud (devices with firmware 5.20 or higher)

function CloseComPort : Smallint; stdcall; export;
```

1.1.2.2 Echo function

Interface converters from KELLER Druckmesstechnik always supply an echo of the message transmitted by the PC. This function has the standard value 1 (Echo On), to enable operation with the converters supplied by KELLER. If other converters are used which do not supply a hardware echo, the function must be set to 0 = Echo Off .

```
function EchoOn( bteEcho: Byte ): Smallint; stdcall; export;
```



1.1.2.3 Protocol functions

The following functions encapsulate the Keller Bus functions described in the Communication protocol.

The description of the communication protocol can be found here: http://www.keller-druck.ch/picts/pdf/engl/comm_protocol_e.pdf

The parameter sequences are identical. The CRC16 checksum is not included here, as it is calculated and checked in the DLL.

Some parameters consist of several bytes. These are grouped together for the sake of clarity.

The different requests a and b belonging to function 95 are split into two functions: F95 and F95val.

Functions F34, F35, F64, F65 and F101 are only listed here for the sake of completeness, and are of no relevance in these devices.

```
function F30( bteDeviceAddr, bteCoeffNo: Byte; var sinCoeff: Single
): Smallint; stdcall; export;

function F31( bteDeviceAddr, bteCoeffNo: Byte; sinCoeff: Single
): Smallint; stdcall; export;

function F32( bteDeviceAddr, bteCoeffNo: Byte; var sinCoeff: Byte
): Smallint; stdcall; export;

function F33( bteDeviceAddr, bteCoeffNo: Byte; sinCoeff: Byte
): Smallint; stdcall; export;

function F34( bteDeviceAddr: Byte; wrdAddr: Word; bteAmount: Byte; pbteData: PByte
): Smallint; stdcall; export;

function F35( bteDeviceAddr: Byte; wrdAddr: Word; bteAmount: Byte; pbteData: PByte
): Smallint; stdcall; export;

function F48(
    bteDeviceAddr: Byte; var bteClass, bteGroup, bteYear, bteWeek, bteBuffer, bteState: Byte
): Smallint; stdcall; export;

function F64( bteDeviceAddr: Byte; wrdAddr: Word; bteAmount: Byte; pbteData: PByte
): Smallint; stdcall; export;

function F65( bteDeviceAddr: Byte; wrdAddr: Word; bteAmount: Byte; pbteData: PByte
): Smallint; stdcall; export;

function F66( bteDeviceAddr, bteNewAddr: Byte; var bteActualAddr: Byte
): Smallint; stdcall; export;

function F69( bteDeviceAddr: Byte; var linSN: Longint
): Smallint; stdcall; export;

function F73( bteDeviceAddr, bteChannel: Byte; var sinValue: Single; var bteStat: Byte
): Smallint; stdcall; export;

function F95( bteDeviceAddr, bteCmd: Byte
): Smallint; stdcall; export;

function F95val( bteDeviceAddr, bteCmd: Byte; sinVal: Single
): Smallint; stdcall stdcall; export;

function F100(
    bteDeviceAddr, bteIndex: Byte; var btePara0, btePara1, btePara2, btePara3, btePara4: Byte
): Smallint; stdcall stdcall; export;

function F101(
    bteDeviceAddr, bteIndex: Byte; btePara0, btePara1, btePara2, btePara3, btePara4: Byte
): Smallint; stdcall stdcall; export;
```



1.2 Changes

- Document Version 1.0, Sept. 2015 DLL-Description in separate document

1.3 DLL-Version

Base version	2005
30.08.2010	base version re-compiled
Version 2.0.0.0	new version for 32bit and 64bit

KELLER AG für Druckmesstechnik

St. Gallerstrasse 119 • CH-8404 Winterthur

Tel: ++41 52 235 25 25

<http://www.keller-druck.com>