



# PicoLog CM3 Data Logger

Programmer's Guide



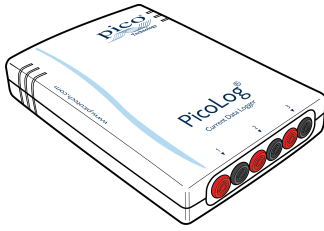
# Contents

1 Introduction .....	1
1 Overview .....	1
2 Legal information .....	2
3 Minimum PC requirements .....	3
4 Company details .....	3
2 Driver information .....	4
1 Introduction .....	4
2 Installing the driver .....	4
3 PLCM3CloseUnit .....	5
4 PLCM3Enumerate .....	6
5 PLCM3GetUnitInfo .....	7
6 PLCM3GetValue .....	8
7 PLCM3IpDetails .....	9
8 PLCM3OpenUnit .....	10
9 PLCM3OpenUnitVialp .....	11
10 PLCM3SetChannel .....	12
11 PLCM3SetMains .....	13
12 Constants and enumerated types .....	13
13 Windows .....	13
3 Writing your own programs .....	14
1 C .....	14
2 Excel .....	14
3 LabVIEW .....	14
Index.....	15



# 1 Introduction

## 1.1 Overview



The PicoLog CM3 is a three-channel, high-resolution data logger for use with current clamps.

This manual explains how to use the API (application programming interface) functions, so that you can develop your own programs to collect and analyse data from the data logger.

### Additional information

For instructions on connecting and using the device, and setting it up with the PicoLog software, please see:

- [PicoLog CM3 Data Logger User's Guide \(plcm3.en.pdf\)](#).

## 1.2 Legal information

The material contained in this release is licensed, not sold. Pico Technology Limited grants a licence to the person who installs this software, subject to the conditions listed below.

### Access

The licensee agrees to allow access to this software only to persons who have been informed of these conditions and agree to abide by them.

### Usage

The software in this release is for use only with Pico products or with data collected using Pico products.

### Copyright

Pico Technology Ltd. claims the copyright of, and retains the rights to, all material (software, documents, etc.) contained in this SDK except the example programs. You may copy and distribute the SDK without restriction, as long as you do not remove any Pico Technology copyright statements. The example programs in the SDK may be modified, copied and distributed for the purpose of developing programs to collect data using Pico products.

### Liability

Pico Technology and its agents shall not be liable for any loss, damage or injury, howsoever caused, related to the use of Pico Technology equipment or software, unless excluded by statute.

### Fitness for purpose

As no two applications are the same, Pico Technology cannot guarantee that its equipment or software is suitable for a given application. It is your responsibility, therefore, to ensure that the product is suitable for your application.

### Mission-critical applications

This software is intended for use on a computer that may be running other software products. For this reason, one of the conditions of the licence is that it excludes use in mission-critical applications, for example life support systems.

### Viruses

This software was continuously monitored for viruses during production, but you are responsible for virus-checking the software once it is installed.

### Support

If you are dissatisfied with the performance of this software, please contact our technical support staff, who will try to fix the problem within a reasonable time. If you are still dissatisfied, please return the product and software to your supplier within 28 days of purchase for a full refund.

### Upgrades

We provide upgrades, free of charge, from our web site at [www.picotech.com](http://www.picotech.com). We reserve the right to charge for updates or replacements sent out on physical media.

### Trademarks

Windows is a trademark or registered trademark of Microsoft Corporation. Pico Technology Limited and PicoLog are internationally registered trademarks.

### 1.3 Minimum PC requirements

To ensure that your PicoLog CM3 Data Logger operates correctly, you must have a computer with at least the minimum system requirements to run one of the supported operating systems, as shown in the following table. The performance of the data logger will be better with a more powerful PC. Please note the PicoLog software is not installed as part of the SDK.

Item	Specification
Operating system	Windows XP SP2 Windows Vista Windows 7
	32 bit and 64* bit versions supported
Processor	As required by Windows
Memory	
Free disk space*	
Ports	USB and optional Ethernet ports

\* While the driver will run on a 64 bit operating system, the driver itself is 32 bit, and therefore will run as 32 bit.

### 1.4 Company details

You can obtain technical assistance from Pico Technology at the following address:

Address: Pico Technology  
James House,  
Colmworth Business Park,  
Eaton Socon,  
St Neots,  
Cambridgeshire PE19 8YP  
United Kingdom

Phone: +44 (0) 1480 396 395  
Fax: +44 (0) 1480 396 296

Email:  
Technical Support: [support@picotech.com](mailto:support@picotech.com)  
Sales: [sales@picotech.com](mailto:sales@picotech.com)

Web site: [www.picotech.com](http://www.picotech.com)

## 2 Driver information

### 2.1 Introduction

The PicoLog CM3 is supplied with driver routines that you can build into your own programs.










Once you have installed the software, the [Drivers](#) directory contains the drivers and a selection of examples of how to use the drivers. It also contains a copy of this manual as a PDF file.

The driver routine is supplied as a Windows DLL.

The Windows DLL can be used with C, C++, Delphi and Visual Basic programs: it can also be used with programs like Microsoft Excel, which uses Visual Basic for Applications (VBA) as its macro programming language.

Using Ethernet (data and power), or Ethernet (data) and USB (power) connections, more than one application can access the PicoLog CM3 at the same time, with each application having its own instance of the driver. However, when using USB for both power and data, only one application can access the PicoLog CM3 at a time.

These are the routines in the driver:

 <a href="#">PLCM3CloseUnit</a>	Close the port (do this each time you finish using the device!)
 <a href="#">PLCM3Enumerate</a>	Get list of attached devices.
 <a href="#">PLCM3GetUnitInfo</a>	Get the batch number and serial number, or the calibration date, of this PicoLog CM3 Data Logger.
 <a href="#">PLCM3GetValue</a>	Get the most recent data reading from a channel.
 <a href="#">PLCM3IpDetails</a>	Read or write IP settings.
 <a href="#">PLCM3OpenUnit</a>	Open the device through its USB interface.
 <a href="#">PLCM3OpenUnitViaIp</a>	Open the device through its Ethernet interface.
 <a href="#">PLCM3SetChannel</a>	Specify the sensor type and filtering for a channel.
 <a href="#">PLCM3SetMains</a>	Change the mains noise filtering setting to 60 Hz. The default is 50 Hz.

The normal calling sequence for these routines is as follows:

1. Open Driver
2. Set Channels
3. While you want to read data
4. Get data
5. End While
6. Close Unit
7. Close Driver

### 2.2 Installing the driver

The driver is installed automatically when you install the PicoLog software. Alternatively, you can download the driver from our website at:

<http://www.picotech.com>.



## 2.3 PLCM3CloseUnit

```
PICO_STATUS PLCM3CloseUnit (  
    short handle  
)
```

This routine disconnects the driver.

Arguments:	<code>handle</code> , identifies the device to close
Returns:	defined in <code>picoStatus.h</code>

## 2.4 PLCM3Enumerate

```
PICO_STATUS PLCM3Enumerate (  
    char                * details,  
    unsigned long       * length,  
    PLCM3_COMMUNICATION_TYPE type  
)
```

This routine returns a list of all the attached PicoLog CM3 devices of the specified port type.

Arguments:	<p><code>details</code>, a string buffer to receive a maximum of <code>length</code> characters</p> <p><code>length</code>,     input:     the length of the <code>string</code> buffer               output:   the length of the information string returned</p> <p><code>type</code>, the communication type used by the PicoLog CM3. Can be any of the following enumerated types:</p> <pre>PLCM3_CT_USB      = 0x00000001 PLCM3_CT_ETHERNET = 0x00000002 PLCM3_CT_ALL      = 0xFFFFFFFF</pre>
Returns:	defined in <code>picoStatus.h</code>

## 2.5 PLCM3GetUnitInfo

```
PICO_STATUS PLCM3GetUnitInfo (
    short      handle,
    char       * string,
    short      * stringLength,
    short      * requiredSize,
    PICO_INFO  info
)
```

This routine obtains information on a specified device.

Arguments:	<p><code>handle</code>, identifies the device whose information is required</p> <p><code>string</code>, output: the information requested</p> <p><code>stringLength</code>, input: the length of the <code>string</code> buffer</p> <p><code>requiredSize</code>, output: the length of the information string requested. If this is longer than <code>stringLength</code> then only the first <code>stringLength</code> characters of the requested information are written to <code>string</code>.</p> <p><code>info</code>, the type of information required. The following types are defined in <code>picoStatus.h</code>:</p> <pre>PICO_DRIVER_VERSION PICO_USB_VERSION PICO_HARDWARE_VERSION PICO_VARIANT_INFO PICO_BATCH_AND_SERIAL PICO_CAL_DATE PICO_KERNEL_DRIVER_VERSION</pre>
Returns:	defined in <code>picoStatus.h</code>

## 2.6 PLCM3GetValue

```
PICO_STATUS PLCM3GetValue (  
    short          handle,  
    PLCM3\_CHANNELS channel,  
    long           * value,  
)
```

Once you open the driver and define some channels, the driver begins to take continuous readings from the PicoLog CM3. When you call this routine, it immediately sets data to the most recent reading for the specified channel.

Arguments:	<a href="#">handle</a> , identifies the device from which to get data  <a href="#">channel</a> , the number of the channel to read, from 1 to 3  <a href="#">value</a> , output: an array where the sample values will be stored
Returns:	defined in <a href="#">picoStatus.h</a>

## 2.7 PLCM3IpDetails

```
PICO_STATUS PLCM3IpDetails (
    short          handle,
    short          * enabled,
    char           * ipaddress,
    unsigned short * length,
    unsigned short * listeningPort,
    PLCM3_IP_DETAILS_TYPE type
)
```

This routine either reads or writes the the IP details of a specified device. The `type` argument controls whether the operation is a read or a write.

Arguments:	<p><code>handle</code>, identifies the device that is the target of the operation</p> <p><code>enabled</code>, input: 1 to enable the device, 0 to disable output: 1 if the device is enabled, 0 if disabled</p> <p><code>ipaddress</code>, input or output: the IP address of the device</p> <p><code>length</code>, input or output: the length of the IP address string</p> <p><code>listeningPort</code>, input or output: the local IP port connected to the device</p> <p><code>type</code>, the type of operation to be performed. Can be either of the following types:</p> <p style="padding-left: 40px;"><code>PLCM3_IDT_GET</code>, to read information from the driver</p> <p style="padding-left: 40px;"><code>PLCM3_IDT_SET</code>, to write information to the driver</p>
Returns:	defined in <code>picoStatus.h</code>

## 2.8 PLCM3OpenUnit

```
PICO_STATUS PLCM3OpenUnit (  
    short * handle,  
    char * serial  
)
```

This routine obtains a handle for the PicoLog CM3 device with the given serial number.

If you wish to use more than one PicoLog CM3, you must call the routine once for each device.

Arguments:	<code>handle</code> , output: handle of the device that was opened. This value is used to identify the device in all further function calls.  <code>serial</code> , input: serial number string of device, null-terminated.
Returns:	defined in <code>picoStatus.h</code>

## 2.9 PLCM3OpenUnitViaIp

```
PICO_STATUS PLCM3OpenUnitViaIp (  
    short * handle,  
    char * serial,  
    char * ipAddress  
)
```

This routine obtains a handle for the Ethernet-connected PicoLog CM3 device, identified by either its IP address or its serial number.

- Using IP address identification, a device anywhere on the internet or local network can be opened.
- Using serial number identification, only a device on the local network can be opened.

If you wish to use more than one PicoLog CM3, you must call the routine once for each device.

Arguments:	<p><code>handle</code>, output: handle of the device that was opened. This value is used to identify the device in all further function calls.</p> <p><code>serial</code>, input: serial number of device as a null-terminated string, or a null pointer if <code>ipAddress</code> is used.</p> <p><code>ipAddress</code>, input: the IP address of the device as a null-terminated string, or a null pointer if <code>serial</code> is used.</p>
Returns:	defined in <code>picoStatus.h</code>

## 2.10 PLCM3SetChannel

```
PICO_STATUS PLCM3SetChannel (
    short      handle,
    PLCM3\_CHANNELS channel,
    PLCM3\_DATA\_TYPES type,
)
```

This routine configures a single channel of the specified PicoLog CM3. It can be called any time after calling [PLCM3OpenUnit](#).

The fewer channels selected, the more frequently they will be updated. Measurement takes around 720 ms per active channel.

Arguments:	<p><a href="#">handle</a>, identifies the device to be configured</p> <p><a href="#">channel</a>, which channel you want to set the details for. It should be between 1 and 3.</p> <p><a href="#">type</a>, the type of reading you require. Choose from the table below.</p>
Returns:	defined in <a href="#">picoStatus.h</a>

PLCM3_DATA_TYPES		Data type
<a href="#">PLCM3_OFF</a>	0	disable channel
<a href="#">PLCM3_1_MILLIVOLT</a>	1	1 mV range (1 mV/A)
<a href="#">PLCM3_10_MILLIVOLTS</a>	2	10 mV range (10 mV/A)
<a href="#">PLCM3_100_MILLIVOLTS</a>	3	100 mV range (100 mV/A)
<a href="#">PLCM3_VOLTAGE</a>	4	( $\mu$ V)



## 2.11 PLCM3SetMains

```
PICO_STATUS PLCM3SetMains (
    short      handle
    unsigned    short sixty_hertz
)
```

This routine is used to inform the driver of the local mains (line) frequency. This helps the driver to filter out electrical noise.

Arguments:	<code>handle</code> , identifies the device to be configured  <code>sixty_hertz</code> , for 50 Hz set to 0; for 60 Hz set to 1
Returns:	defined in <code>picoStatus.h</code>

## 2.12 Constants and enumerated types

```
typedef enum enPLCM3Channels
{
    PLCM3_CHANNEL_1 = 1,
    PLCM3_CHANNEL_2,
    PLCM3_CHANNEL_3,
    PLCM3_MAX_CHANNELS = PLCM3_CHANNEL_3
} PLCM3_CHANNELS;
```

```
typedef enum enPLCM3DataType
{
    PLCM3_OFF,
    PLCM3_1_MILLIVOLT,
    PLCM3_10_MILLIVOLTS,
    PLCM3_100_MILLIVOLTS,
} PLCM3_DATA_TYPES;
```

```
typedef enum enIpDetailsType
{
    PLCM3_IDT_GET,
    PLCM3_IDT_SET,
} PLCM3_IP_DETAILS_TYPE;
```

```
typedef enum enCommunicationType
{
    PLCM3_CT_USB = 0x00000001,
    PLCM3_CT_ETHERNET = 0x00000002,
    PLCM3_CT_ALL = 0xFFFFFFFF
} COMMUNICATION_TYPE;
```

## 2.13 Windows

The 32-bit Windows driver is the file `plcm3.dll`, which is included in the SDK. If an application is unable to find the DLL, try moving the DLL to `c:\windows\system`.

## 3 Writing your own programs

### 3.1 C

The C example is a console mode program that demonstrates the facilities of the driver.

To compile the program, create a new project containing the following files from the PicoLog CM3 SDK:

● `PLCM3con.c`

and:

● `PLCM3bc.lib` (Borland 32-bit applications) or

● `PLCM3.lib` (Microsoft Visual C 32-bit applications)

The following file must be in the compilation directory:

● `PLCM3Api.h`

and the following file must be in the same directory as the executable:

● `PLCM3.dll`

### 3.2 Excel

The easiest way to transfer data into Excel is to use PicoLog.

If, however, you need to do something that is not possible using PicoLog, you can write an Excel macro that calls `PLCM3.dll` to read in a set of data values. The Excel Macro language is similar to Visual Basic.

The example `PLCM3.xls` reads values from all three channels every second and assigns them to cells in the spreadsheet.

### 3.3 LabVIEW

The routines described here were created using LabVIEW 8.2 on Windows XP.

To use these routines, copy `plcm3.dll` to your LabVIEW `user.lib` directory.

`PLCM3.vi` is a fully functional example LabVIEW application. It demonstrates how to connect to the device using both USB and Ethernet. It also demonstrates reading all possible measurement types from the three channels.

# Index

## A

Address 3

## B

Batch number 7

## C

C 14

Calibration date 7

Channel setup 12

Closing a unit 5

COMMUNICATION\_TYPE type 13

Contact details 3

## D

Driver version 7

## E

Email 3

Ethernet port 11

Excel 14

## F

Fax 3

## H

Handle, obtaining 10

Hardware version 7

## I

Installation 4

IP address 11

IP details 9

IP\_DETAILS\_TYPE type 13

## K

Kernel driver version 7

## L

LabVIEW 14

Legal information 2

## M

Mains frequency 13

## O

Opening a unit 10

## P

PC requirements 3

PLCM3\_CHANNELS type 13

PLCM3\_DATA\_TYPES type 13

PLCM3CloseUnit 5

PLCM3Enumerate 6

PLCM3GetUnitInfo 7

PLCM3GetValue 8

PLCM3IpDetails 9

PLCM3OpenUnit 10

PLCM3OpenUnitViaIp 11

PLCM3SetChannel 12

PLCM3SetMains 13

## S

Serial number 7

## T

Technical assistance 3

Telephone 3

## U

USB 3

USB version 7

## V

Variant information 7

## W

Website 3

Windows 13







## Pico Technology

James House  
Colmworth Business Park  
ST. NEOTS  
Cambridgeshire  
PE19 8YP  
United Kingdom  
Tel: +44 (0) 1480 396 395  
Fax: +44 (0) 1480 396 296  
[www.picotech.com](http://www.picotech.com)

plcm3pg.en-1

23.5.11

Copyright © 2011 Pico Technology Ltd. All rights reserved.