PicoLog® CM3

Current Data Logger

Programmer's Guide

plcm3pg.en r3


pico
Technology

Contents

1 Introduction 1

1 Overview 1

2 Legal information 1

3 Minimum PC requirements 2

2 Driver information 3

1 About the driver 3

2 PLCM3CloseUnit 4

3 PLCM3Enumerate 5

4 PLCM3GetUnitInfo 6

5 PLCM3GetValue 7

6 PLCM3IpDetails 8

7 PLCM3OpenUnit 9

8 PLCM3OpenUnitViaIp 10

9 PLCM3SetChannel 11

10 PLCM3SetMains 12

11 Constants and enumerated types 13

3 Programming examples 14

Index 15


pico
Technology

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 analyze 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*

1.2 Legal information

The material contained in this release is licensed, not sold. Pico Technology Limited grants a license 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 license 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 14 days of purchase for a full refund.

Upgrades. We provide upgrades, free of charge, from our web site at 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 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 that the PicoLog software is not installed as part of the SDK.

Item	Specification
Operating system	Windows 7, Windows 8 or Windows 10 32-bit and 64-bit versions supported
Processor	As required by the operating system
Memory	
Free disk space	
Ports	USB and optional Ethernet ports

2 Driver information

2.1 About the driver

The Pico Technology software development kit (SDK) is available in 32-bit and 64-bit versions and contains the drivers and a selection of examples showing how to use them.

The driver is supplied as a Windows dynamic link library (DLL), `plcm3.dll`, which can be found in the `lib` subdirectory of your SDK installation and allows you to program a PicoLog CM3 using standard C function calls. The Windows DLL exports the function definitions in standard C format, but this does not limit you to programming in C. You can use the API with any programming language that supports standard C calls.

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 device at a time.

These are the routines in the driver:

PLCM3CloseUnit	Close the port (do this each time you finish using the device!)
PLCM3Enumerate	Get list of attached devices
PLCM3GetUnitInfo	Get the batch number and serial number, or the calibration date, of this PicoLog CM3
PLCM3GetValue	Get the most recent data reading from a channel
PLCM3IpDetails	Read or write IP settings
PLCM3OpenUnit	Open the device through its USB interface
PLCM3OpenUnitViaIp	Open the device through its Ethernet interface
PLCM3SetChannel	Specify the sensor type and filtering for a channel
PLCM3SetMains	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. Load driver
2. Open driver
3. Set channels
4. While you want to read data
5. Get data
6. End While
7. Close unit
8. Unload driver (happens automatically when you terminate the application)

2.2 PLCM3CloseUnit

```
PICO_STATUS PLCM3CloseUnit
(
  int16_t   handle
)
```

This routine disconnects the driver from the device.

Arguments:	`handle`, identifies the device to close
Returns:	defined in `PicoStatus.h`

2.3 PLCM3Enumerate

```
PICO_STATUS PLCM3Enumerate
(
  int8_t                    * details,
  uint32_t                  * length,
  PLCM3_COMMUNICATION_TYPE  type
)
```

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

Arguments:	`details`, a string buffer to receive a maximum of `length` characters. The string is a comma-separated list of attached devices of the selected `type`. The buffer must be long enough to accommodate the expected string.
	
	USB devices are listed in this format:
	
	`USB:Serial Number`
	
	Example: `USB:DV045/055`
	
	Ethernet devices are listed in this format:
	
	`IP:Serial Number[IP Address:port]`
	
	Example: `IP:DV045/055[192.168.1.253:6500]`
	
	`length`, input: the length of the `string` buffer
	output: the length of the information string returned
	
	`type`, the communication type used by the PicoLog CM3. Can be any of the following enumerated types:
	
	`PLCM3_CT_USB = 0x00000001`
	`PLCM3_CT_ETHERNET = 0x00000002`
	`PLCM3_CT_ALL = 0xFFFFFFFF`

Returns:	defined in `PicoStatus.h`

2.4 PLCM3GetUnitInfo

```
PICO_STATUS  PLCM3GetUnitInfo
(
  int16_t     handle,
  int8_t      * string,
  int16_t     stringLength,
  int16_t     * requiredSize,
  PICO_INFO   info
)
```

This routine obtains information on a specified device.

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

Returns:	defined in `PicoStatus.h`

2.5 PLCM3GetValue

```
PICO_STATUS PLCM3GetValue
(
  int16_t          handle,
  PLCM3_CHANNELS   channel,
  int32_t          * 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:	`handle,` identifies the device from which to get data `channel,` the number of the channel to read, from 1 to 3 `value,` output: an array where the sample values will be stored
Returns:	defined in `PicoStatus.h`

2.6 PLCM3IpDetails

```
PICO_STATUS PLCM3IpDetails
(
  int16_t                 handle,
  int16_t                 * enabled,
  int8_t                  * ipaddress,
  uint16_t                * length,
  uint16_t                * listeningPort,
  PLCM3_IP_DETAILS_TYPE   type
)
```

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

Arguments:	`handle,`	identifies the device that is the operation target
	`enabled,`	input: 1 to enable the device, 0 to disable output: 1 if the device is enabled, 0 if disabled
	`ipaddress,`	input or output: the IP address of the device
	`length,`	input or output: the length of the IP address string
	`listeningPort,`	input: the size of the string array specified in `ipaddress` to receive the IP address string. output: the length of the IP address string
	`type,`	the type of operation to be performed. Can be either of the following types: `PLCM3_IDT_GET,` to read information from the driver `PLCM3_IDT_SET,` to write information to the driver
Returns:	defined in `PicoStatus.h`	

2.7 PLCM3OpenUnit

```
PICO_STATUS PLCM3OpenUnit
(
  int16_t    * handle,
  int8_t     * serial
)
```

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

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

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

2.8 PLCM3OpenUnitViaIp

```
PICO_STATUS PLCM3OpenUnitViaIp
(
  int16_t    * handle,
  int8_t     * serial,
  int8_t     * 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:	`handle`, output: handle of the device that was opened. This value is used to identify the device in all further function calls. `serial`, input: serial number of device as a null-terminated string, or a null pointer if `ipAddress` is used. `ipAddress`, input: the IP address of the device as a null-terminated string, or a null pointer if `serial` is used. String format: `<ipaddress>:<port>`
Returns:	defined in `PicoStatus.h`

2.9 PLCM3SetChannel

```
PICO_STATUS PLCM3SetChannel
(
  int16_t            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:	handle, identifies the device to be configured channel, which channel you want to set the details for. It should be between 1 and 3. type, the type of reading you require. Choose from the table below.
Returns:	defined in PicoStatus.h

PLCM3_DATA_TYPES		Data type
PLCM3_OFF	0	disable channel
PLCM3_1_MILLIVOLT	1	1 mV range (1 mV/A)
PLCM3_10_MILLIVOLTS	2	10 mV range (10 mV/A)
PLCM3_100_MILLIVOLTS	3	100 mV range (100 mV/A)
PLCM3_VOLTAGE	4	(µV)

2.10 PLCM3SetMains

```
PICO_STATUS PLCM3SetMains
(
  int16_t    handle
  uint16_t   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:	`handle`, identifies the device to be configured `sixty_hertz`, for 50 Hz set to 0; for 60 Hz set to 1
Returns:	defined in `PicoStatus.h`

2.11 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;
```

3 Programming examples

Your Pico Technology SDK installation includes programming examples in various languages and development environments. Please refer to the SDK for details.

Index

B

Batch number 6

C

Calibration date 6
Channel setup 11
Closing a unit 4
COMMUNICATION_TYPE type 13

D

Driver version 6

E

Ethernet port 10

H

Handle, obtaining 9
Hardware version 6

I

IP address 10
IP details 8
IP_DETAILS_TYPE type 13

K

Kernel driver version 6

L

Legal information 1

M

Mains frequency 12

O

Opening a unit 9

P

PC requirements 2
PLCM3_CHANNELS type 13
PLCM3_DATA_TYPES type 13
PLCM3CloseUnit 4
PLCM3Enumerate 5
PLCM3GetUnitInfo 6
PLCM3GetValue 7
PLCM3IpDetails 8
PLCM3OpenUnit 9
PLCM3OpenUnitViaIp 10
PLCM3SetChannel 11
PLCM3SetMains 12
Programming examples 14

S

Serial number 6

U

USB 2
USB version 6

V

Variant information 6

UK headquarters

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

sales@picotech.com
support@picotech.com

www.picotech.com

USA headquarters

Pico Technology
320 N Glenwood Blvd
Tyler
Texas 75702
United States

Tel: +1 800 591 2796
Fax: +1 620 272 0981

plcm3pg.en r3 2016-06-29