

Manuale codice Picoscope

Leo Paolo

25 giugno 2021

1 Settare ambiente e librerie Picoscope

Questo capitolo è una guida ad installare il software e impostare l'ambiente di lavoro per poter utilizzare il codice d'esempio reso disponibile dagli sviluppatori Picoscope per c/c++. Parte di ciò che verrà detto in questo capitolo è presente all'indirizzo <https://github.com/picotech/picosdk-c-examples>, insieme al codice d'esempio che utilizzeremo. Se l'ambiente di lavoro è già impostato andate al capitolo successivo.

Presupponendo di avere una macchina Linux, aprire il terminale ed eseguire i seguenti comandi:

```
sudo apt-get install libps5000a
```

Questo comando installerà le librerie necessarie per poter utilizzare il codice d'esempio del Picoscope. Adesso, andate nella cartella:

```
cd /opt/picoscope/share/doc/libps5000a
```

Collegate il Picoscope attraverso la porta USB al vostro computer ed eseguite il comando:

```
./usbtest
```

Se il Picoscope è alimentato, collegato al vostro pc, e avete installato con successo la libreria necessaria, dovrete ottenere un output simile a questo: Adesso che ci siamo assicurati che la libreria è installata

```
(base) paolo@paolo-Nitro5:/opt/picoscope/share/doc/libps5000a$ ./usbtest
*****
***** Pico Technology Linux USB Diagnostics *****
*****
This script looks for Pico USB devices connected to your computer
and checks whether you are able to access them. If your user account
does not have permission to access the devices, the script will try
to suggest how to correct this for your particular Linux installation.
Press enter to continue...

Getting user and group information...
- You are running this script as paolo.
- You are in the following groups:
  paolo adm cdrom sudo dip video plugdev lpadmin sambashare

Please connect your Pico USB device and press enter to continue...

Pico USB device found: /dev/bus/usb/001/005
- It belongs to root (which is not you) who has permissions rw-
- The members of group root (which you are not in) have permissions rw-
- Everyone else (this is you) has permissions rw-
- You can write to this device and so will be able to use it.
(base) paolo@paolo-Nitro5:/opt/picoscope/share/doc/libps5000a$
```

correttamente, scarichiamo in una cartella apposita tutti i file d'esempio disponibili all'indirizzo github:

```
git clone https://github.com/picotech/picosdk-c-examples
```

Una volta scaricato, andate nella cartella:

```
cd ps5000a/linux-build-files/
```

Dove "ps5000a" è la libreria che utilizzeremo per la nostra versione del Picoscope (5444D). Una volta in questa cartella copiate il file che si trova nella cartella "ps5000aCon":

```
cp ../ps5000aCon/ps5000aCon.c .
```

e usate il seguente comando:

```
./autogen.sh
```

Adesso è possibile incorrere in una serie di errori, di cui elenco le soluzioni:

1. Can't exec "aclocal": No such file or directory at /usr/share/autotconf/AutoM4te/FileUtils.pm line 326.
autoreconf: failed to run aclocal: No such file or directory

Soluzione:

```
sudo apt-get install automake
```

2. `configure: error: libps5000a-1.1/ps5000aApi.h missing!`

Soluzione: Bisogna rimuovere dai seguenti file presenti nella attuale cartella: `configure.ac` e `ps5000aCon.c` tutti i riferimenti alle librerie `libps5000a-1.1` in questo modo:
`libps5000a-1.1`→`libps5000a` oppure `...-1.1`→`...`

Una volta risolti gli eventuali errori eseguire il seguente comando:

```
make
```

Se appare questo output sul terminale:

```
make all-am
make[1]: Entering directory '/home/md13/Desktop/Fermilab/Magnetometer/INO/picosdk_C/ps5000a/linux-build-files'
CC ps5000aCon.o
CCLD ps5000aCon
make[1]: Leaving directory '/home/md13/Desktop/Fermilab/Magnetometer/INO/picosdk_C/ps5000a/linux-build-files'
```

Allora tutto ok! Dovrebbe essere presente adesso nella cartella un eseguibile che è possibile lanciare con il comando:

```
./ps5000aCon
```

2 Utilizzare il codice