1 Introduction

Congratulations on having purchased a Teledyne SP Devices digitizer product. To get the most out of the digitizer we recommend that you read the documentation set carefully.

Disclaimers and Safety

⚠️ Caution!

Ground the antistatic package before removing the board from the package. Electrostatic discharge may damage the card. Be sure to ground yourself by touching the grounded frame and avoid touching any components on the card.

⚠️ Caution!

Before connecting any equipment to the digitizer, please check the absolute maximum ratings in the digitizer data sheet to assure that the connected equipment cannot damage the digitizer.
2 Installing the Software

Before connecting the digitizer to the host computer for the first time, the software development kit (SDK) must be installed. To install the SDK, enter the packages/ directory and find the subdirectory corresponding to the Linux distribution and processor architecture of the host computer. Instructions for each distribution follows below.

**Important**

Make sure the digitizer is connected to the host computer before you begin the installation process.

Ubuntu and Debian

Install the delivered SDK packages using `dpkg -i packagename`. Use the following order:

1. spd-adq-pci-dkms (needed for PCIe/PXIe/MTCA)
2. libadq0
3. adqtools

OpenSUSE and SUSE Linux Enterprise

Install the following packages using `zypper install packagename`:

- make
- kernel-devel
- kernel-source
- gcc

The version of kernel-devel and kernel-source must match your current kernel. Install the delivered SDK packages using `rpm -U packagename`. Use the following order:

1. dkms (needed for PCIe/PXIe/MTCA)
2. spd-adq-pci-dkms (needed for PCIe/PXIe/MTCA)
3. libadq0
4. adqtools

Fedora 19, 20 and 21

Install the following packages using `yum install packagename`:

- dkms (needed for PCIe/PXIe/MTCA)

Install the delivered SDK packages using `rpm -U packagename`. Use the following order:

1. spd-adq-pci-dkms (needed for PCIe/PXIe/MTCA)

---

1 These instructions are also available in the README included in the SDK archive.
2. libadq0
3. adqtools

**Fedora 22 and higher**

Install the following packages using `dnf install packagename`:

- `dkms` (needed for PCIe/PXIe/MTCA)

Install the delivered SDK packages using `dnf install packagename`. Use the following order:

1. `spd-adq-pci-dkms` (needed for PCIe/PXIe/MTCA)
2. `libadq0`
3. `adqtools`

**CentOS / Red Hat Enterprise Linux / Scientific Linux**

**Note**

For RHEL6, use CentOS6 packages.

Install the following packages using `yum install packagename`:

- `make`
- `kernel-devel`
- `gcc`

The version of `kernel-devel` must match your current kernel. Install the SDK packages using `rpm -U packagename`. Use the following order:

1. `dkms` (needed for PCIe/PXIe/MTCA)
2. `spd-adq-pci-dkms` (needed for PCIe/PXIe/MTCA)
3. `libadq0`
4. `adqtools`

After installing all necessary packages, reboot the system so that `udev` reads the updated configuration and the driver loads. In earlier versions there was a package called `adqupdate` which is now replaced by the `adqtools` package.
Device Access Rights

ADQ devices show up prefixed with /dev/adq_pcie_ and /dev/adq_usb_. The default udev setting is to add read/write access to the user group adq. The libadq0 package will create a user group called adq if that group does not already exist in the system. To grant a user access to the device, add the user to the adq group using with

```
$ usermod -a -G adq <username>
```

The user will have to logout and login again for the changes to take effect.

Compatibility

The PCIe kernel module supports kernel versions from 2.6.32 and forward, however kernel version 3.8.0 or newer is recommended. The PCIe kernel module is not signed and thus will not load if your kernel uses secure boot. If secure boot is enabled, it will need to be disabled before the kernel module can be loaded. Refer to your distribution documentation on how to do this.

ADQAPI

The ADQAPI is the application programming interface (API) used by the host computer to communicate with digitizers from the ADQ product range. The API functions are detailed in the ADQAPI reference guide [1] and general usage is documented in the ADQAPI user guide [2]. There are two different interfaces available: a C interface and a C++ interface. Most programming languages, e.g. Python, have a foreign function interface granting the ability to call functions from C dynamic link libraries directly, making this the more general interface of the two.
3 Connecting the Digitizer to the Host Computer

After the SDK has been installed, it is time to connect your digitizer to the host computer. The connection procedure depends on the form-factor.

**USB**

For USB units you need:

- A main power supply, 12 V (delivered together with the digitizer)
- A USB cable (delivered together with the digitizer)
- A host computer with a USB port (USB 2.0 or later)

Connect the digitizer to the power supply and to the host computer.

**PXie**

For PXie units you need a host computer with a free PXie or cPCIe slot. With the host computer powered off, plug in the digitizer. Power on the host computer.

**MTCA**

For a MTCA unit you need a host computer with an available MTCA slot. With the host computer powered off, plug in the digitizer. Power on the host computer.

**PCIe**

For PCIe units you need:

- A host computer with an available PCIe slot
- A PCIe 6-pin power connector like the one below. Please note that an adapter may be required for some systems. The adapter can be purchased from a computer accessory supplier.

![PCIe connector](image)

With the host computer powered off, plug in the digitizer and connect the power cable. Make sure that the digitizer is mechanically supported by screws through the front panel. Power on the host computer.
# PXIe Front Panel

<table>
<thead>
<tr>
<th>LED</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDY</td>
<td>Not used</td>
</tr>
<tr>
<td>STA/STAT</td>
<td>Blinking red light in combination with PWR LED off indicates that the device has overheated and partially powered down to prevent damage.</td>
</tr>
<tr>
<td>PWR</td>
<td>Solid green light indicates power and status OK.</td>
</tr>
<tr>
<td>USR/USER</td>
<td>Solid blue light when the digitizer is acquired by an application using the ADQAPI.</td>
</tr>
<tr>
<td>ATN</td>
<td>PXIe attention LED</td>
</tr>
</tbody>
</table>
### MTCA Front Panel

| MMC Error | Solid red light if the MMC detected an error. |
| MMC Operation | Solid green light while the device is running. |
| Hotswap | Linux only |

- **Solid**—OK to disconnect the device.
- **Flashing**—wait before disconnecting the device.
- **Off**—normal mode: pull hot swap pin gently to activate indicator.
4 Using the Digitizer

To introduce you to the interface for our digitizers: the ADQAPI, there are source code examples provided in the examples/ directory in the SDK archive. We recommended you to browse through the contents of this directory to get an overview of the available example code.

**Note**

Please note that there are different source code examples for different products and firmwares.

It is also helpful to familiarize yourself with the documentation for your specific product. Which documents to read depend on which firmware your digitizer is running. Refer to p. 9 for an overview of the available resources. Additional documentation is available on our web site\(^2\).

\(^2\)https://www.spdevices.com/documentation
## Common resources

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAN 16-1796</td>
<td>The ADQ7 product manual</td>
</tr>
<tr>
<td>UG 18-2059</td>
<td>Describes how to manage firmware files.</td>
</tr>
<tr>
<td>UG 08-0214</td>
<td>The ADQAPI user guide</td>
</tr>
<tr>
<td>RG 14-1351</td>
<td>The ADQAPI reference guide, documents the functions in the ADQAPI.</td>
</tr>
</tbody>
</table>

## Firmware-specific resources

<table>
<thead>
<tr>
<th>Resource</th>
<th>FWDAQ</th>
<th>FWSDR</th>
<th>FWPD</th>
<th>FWATD</th>
</tr>
</thead>
<tbody>
<tr>
<td>User guides</td>
<td>UG 17-2043</td>
<td>UG 18-2132</td>
<td>UG 17-1957</td>
<td></td>
</tr>
<tr>
<td>Source code examples in C</td>
<td>ADQAPI_example</td>
<td>ADQAPI_FWSDR_example</td>
<td>ADQAPI_FWPD_example</td>
<td>ADQAPI_FWATD_example</td>
</tr>
<tr>
<td></td>
<td>standard_example</td>
<td>fwsdr_example</td>
<td>fwpd_example</td>
<td>fwatd_example</td>
</tr>
<tr>
<td>GUls</td>
<td>ADCaptureLab (UG 18-2207)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application notes</td>
<td>AN 18-2118</td>
<td>AN 18-2104</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
References

