This version (26 Feb 2021 [2] [2]) was approved by Andrew, John De Crescio.
This Revision, approved version (17 Feb 2021 [1]) was created by John De Crescio.

ADXRS290 Gyroscope PMOD Demo

The EVAL-ADXRS290-PMOZ is a simple Pmod form factor evaluation board for the ADXRS290, a high performance MEMS yaw rate and roll axis angular rate sensor. Gyroscopic designs for use in stabilization applications.

The solder sensors and digitizes the **X axis** and **Y axis** yaw rate **and roll pitch** angular rate, producing a positive reading for clockwise rotation about the x and y axis.

The ADXRS290 provides an output full-scale range of ±300°/s with a sensitivity of 206 LSB/°/s. The ADXRS290 also provides a temperature output that is proportional to absolute temperature. The rate of the package around an x-axis and, regular rate data is formatted as a 16-bit twos complement and is accessible through an SPI digital interface. The ADXRS290 contains a low noise filter of 0.004°/s/√ Hz and a bandwidth of 40 Hz, which is ideal for most applications.

The ADXRS290 operates from a single 5V and requires a complete power-down, with a maximum case frequency of 10kHz.

The Digital communication is [2] EVAL-ADXRS290-PMOZ is accomplished using a standard-compatible [2] PMOD port.

Connector P1	
Description	**Pmod**
NC	1
MOSI	2
MISO	3
SCLK	4
GND	5, 11
VDD	6, 12
RDY	7
RST	8
CS	9
NC	10

Features

- Yaw rate and roll angular rate
- Ultralow noise: 0.004°/s/√Hz
- High vibration rejection over a wide frequency range
- Power saving Standby mode
- Fast start-up with programmable bandwidth
- Fast startup time from standby mode, 1 ms
- Low delay of ≤ 0.5 ms for a 40 Hz input at the extend bandwidth setting
- Programmable high-pass filter cut-off frequency
- Programmable high-pass filter cut-off frequency
- SPI interface
- 2.7 V to 5.2 V operation
- -25°C to +85°C operation
- PMOD form factor

Device Driver Support

The software driver stack (software) are provided for controlling the **EVAL-ADXRS290-PMOZ PMOD** using the [2] device driver on the **EVAL-ADICUP3029** platform and Linux device driver on the **Raspberry Pi** platform.

The Linux driver uses the standard Industrial I/O framework, giving the advantage of application code on the Linux platform Linux IIO framework. The Linux IIO framework provides a standard sysfs interface for application code to talk directly to the platform hardware, communicating with the device over the local bus(serial), or from a remote host over the network or [2] backchannel.

The ADICUP3029 embedded application uses the ADXRS290 [2] [2] driver, and emulates the Linux IIO Framework through the Iceboard daemon library. The application communicates with the host computer via the serial backchannel, over a [2] USB2 physical connection. This facilitates rapid application development using a host computer. Implementation then embedded code is straightforward.

Software, sysfs software (bin_IIO) IIO Oscilloscope, PyADI-IIO, etc.) can be used to talk to the EVAL-ADXRS290-PMOZ on Raspberry Pi and on ADICUP3029.

General Setup Using ADICUP3029

The EVAL-ADXRS290-PMOZ can be used with ADICUP3029.

Demo Requirements

The following is a list of items needed in order to replicate this demo: Hardware:

- EVAL-ADICUP3029
- EVAL-ADXRS290-PMOZ
- Micro USB cable (USB 2)
- PC or Laptop with a USB port

Software:

- **Iceboard** (pre-built HEX file)
- [2] ADXRS290_IIO_GIL_BRING_UP
  - Bring-up/porting project for ADICUP3029 platform

To build the project from source, frame the instructions in the [2] tutorial wiki.

Setting up the Hardware

1. Connect **EVAL-ADXRS290-PMOZ** board at connector **P1** of the **EVAL-ADICUP3029**
2. Connect a micro USB cable to the PC1 connector of the EVAL-ADICUP3029 and connect it to a computer. The final setup should look similar to the picture below.

Flashing the Firmware/Program

1. Make sure the following switches are as shown from the table below:
2. Connect the ADICUP3029 to the PC host via micro USB cable as shown below.
3. Power up PC, open My Computer and look for a USB drive. Wait for a few seconds for the drivers are complete and correct.

4. Simply extract the zip file from drag and drop the provided .bin file to the CMSIS-DAP drive, and your ADICUP3029 board will be programmed. The CS0 and LED will blink rapidly.
5. The CS0 will stop blinking and will stay ON once the programming is done.
6. Disconnect then reconnect the EVAL-ADICUP3029 to the host computer.

> **IIO Oscilloscope Software Installation Command Line Interface** [2] program is also available for the ADXRS290 Pmod on the EVAL-ADXRS290-PMOZ. ADXRS290 Gyroscope PMOD Demo.

General Setup Using Raspberry Pi

The EVAL-ADXRS290-PMOZ can be used with a Raspberry Pi.

Demo Requirements

The following is a list of items needed in order to replicate this demo:

- **Hardware**
  - EVAL-ADXRS290-PMOZ Circuit Evaluation Board
  - Raspberry Pi 3 Model B, Pi 4 Model B
  - 16GB or larger Class 10 or faster micro [2] card
  - 2Gb, 2.5A power supply with micro [2] connector [2] [2] power supply for Raspberry Pi 4)
  - Micro USB cable (USB 2) for serial backchannel
    - For Serial console to access program run, go to Putty from SchmartBoard
    - [2] Micro USB cable (USB 2) for serial backchannel
- Linux Kernel Source
  - 4.19.94 +kernel, kernel-devel, kernel-headers
  - Host Workstation (laptop/desktop computer) or same software as Raspberry Pi
- **Software**
  - Analog Devices Kuiper Linux

Loading Image on SD Card

Analog Devices Kuiper Linux image is used to flash the SD card. It also used to install [2] Kuiper Linux on [2] card. Computer instructions, including varies to download the [2] image, how to write it to the [2] card, and how to configure the system are provided at **Analog Devices Kuiper Linux**.

[2] Kuiper image and follow the system configuration procedure.

Configuring the SD Card

Follow the Kuiper Linux installation guide **Preparing the Image: Raspberry Pi** to the Analog Devices Kuiper Linux page, substituting the following items in **config.txt**:

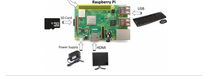
```
dtoverlay=pi3-adxrs290
```

Setting up the Hardware

To set up the circuit for evaluation, consider the following steps:

> Do note that the I/O pins of the Raspberry Pi board are only 3.3V tolerant. Any peripheral to be attached should be powered by a source not exceeding 3.3V.

1. Connect EVAL-ADXRS290-PMOZ board at chalk header connector of the Raspberry Pi.
2. Connect the micro USB cable to the micro USB connector on the Raspberry Pi board and the designated slot on the VPX.
3. Connect the system to a monitor using an HDMI cable through the micro HDMI connector on the VPX.
4. Connect a [2] keyboard and mouse to the VPX through the [2] ports.
5. Finish on the VPX board by plugging in its 5V power supply with a micro USB connector. The final setup should look similar to the picture below.



Software (both ADICUP3029 and Raspberry Pi)

Connection

To be able to control your device, the software must be able to connect to it. The context created in the software depends on the backend used to talk to the device. For the ADXRS290 Pmod, there are two options: USB (using the [2] Iceboard) or Linux (using the Raspberry Pi running the ADXRS290 Daemon). For using the [2] Iceboard, Linux and the ADICUP3029 running the ADICUP3029 IIO demo project. The user needs to supply a [2] which will be used in the context creation.

For Linux, the context can be created using the package that exports all IIO attributes. Make sure that you have the libiio package installed on your machine. Upon installation, always verify the command on the terminal that is accessed. 2

For SPI Direct Local Access:

```
iio_info
```

For Windows machine connected to Raspberry Pi:

```
iio_info -u ip:ip address of your pi
```

Example:

```
# plus Raspberry Pi has IP address 192.168.1.7, you have to use iio_info -u ip:192.168.1.7 or just iio_info
```

> Do note that the Windows machine and the SPI board should be connected to the same network in order for the machine to detect the device.

For Windows machine connected to ADICUP3029:

```
iio_info -u serial:/dev/ttyX,
```

Example:

```
# on a Windows machine, you can check the port of your ADICUP3029 via Device Manager in the Ports (COM & LPT) section. If your device is at COM4, then type in: iio_info -u serial:COM4 or iio_info
# In a Unix-based machine, you will see it under the /dev directory as the tty/serial *tty/dev/*, where x is a number depending on how many serial [2] devices attached. If you see that your device is ttyUSB0, you have to type: iio_info -u serial:/dev/ttyUSB0
```

IIO Commands

There are different commands that can be used to manage the device being used. The **iio_attr** command reads and writes IIO attributes.

```
echo(@adxrs290-0 iio_attr [OPTIONS]...
```

Example:

```
# To look at the current attributes, enter this code on the terminal:
echo(@adxrs290-0 iio_attr -a -c
```

The **iio_mon** command reads a series [2] or [2] response in an IIO device. This is generally not needed for end applications, but can be useful in debugging. The **iio_reg** command is a debug utility that is persistent using the **-p** option and can be used directly to accessing the device via SPI or Linux (as are using the ADICUP3029 platform).

```
echo(@adxrs290-0 iio_reg -p <content> <device> <register> [<value>]
```

Example:

```
# To read the device ID (register = 0x02) of an ADXRS290 interfaced via SPI from a Windows machine, enter the following code on the terminal:
iio_reg -u ip:ip address> adxrs290 0x02
```

IIO Oscilloscope

> **Make sure to downloadupdate to the latest version** of IIO Oscilloscope found on the GitHub https://github.com/analogdevicesinc/osc/releases/latest.

1. Once done with the installation of an update of the latest IIO Oscilloscope, open the application. The user needs to supply a [2] which will be same the context created at the IIO Oscilloscope. Select the ADXRS290 and click on the device from the previous section.
2. Press refresh to display available IIO Devices, once selected the appropriate Linux context.

Debug Panel

Below is the Debug panel of ADXRS290 IIO Oscilloscope that directly accesses the attributes of the device.

DMM Panel

Access the DMM panel to see the instantaneous reading of the X and Y IIO angular velocities.

Waveform Panel

The Waveform panel, also known as the Capture window, displays the real-time waveform of ADXRS290's responses.

> The readings from the DMM panel will freeze upon activation of the waveform pane in the waveform panel [2] the two panels are designed to be used separately and not simultaneously.

PyADI-IIO

PyADI-IIO is a python abstraction module for [2] hardware with IIO drivers to make them easier to use. This module provides device-specific APIs for communicating with the IIO Oscilloscope. These interfaces try to match the driver naming as much as possible without the need to understand the composition of IIOC and IIO.

Install PyADI-IIO using one of the methods in PyADI-IIO.

The ADXRS290 example requires a number of packages to be installed before it can be used. To ensure that all required packages are present, run the following command in the terminal. Some python projects and tool be affected by this added configuration. Make sure that [2] virtualenv has been installed before proceeding.

Creating a Virtual Environment

1. Open a command prompt and navigate to the pyadi-iio directory
2. Create a virtual environment by entering the following command:

```
$ pyadi-iio\python -m venv venv\adxrs290
```

Note that the last argument adxrs290 is just the name of the virtual environment to be created. It can be replaced by any other name or identifier that you prefer. In case the code does not work, try:

```
$ pyadi-iio\venv\adxrs290\Scripts\activate
```

3. Install the following command to activate the virtual environment:

```
$ pyadi-iio\venv\adxrs290\Scripts\activate
```

Installing the Packages

Install activation of the virtual environment, enter the following commands:

```
(venv\adxrs290) $ pyadi-iio\venv\adxrs290\Scripts\activate
(venv\adxrs290) $ (venv\adxrs290) $ (venv\adxrs290) $ (venv\adxrs290) $ pyadi-iio\ ./ ./ ./
```