

[Configuring LabJack T Series Digital Inputs and Outputs \(DIO\) using MatDeck](#)

In this document, we will illustrate two alternative ways for configuring the DIO channels for LabJack T7 and T4 devices.

- MatDeck provides LabJack functions, and configuration can be done directly in MatDeck's C++ style Script . Additionally, you can configure devices in Python.
- However, the most effective and intuitive way for configuring the devices is by using MatDeck GUI configuration.

This document is "alive" and simultaneously executes commands, which is one of MatDeck's unique advantages.

Configuring PWM out at Selected Channel in MatDeck script

The pulse width modulation (PWM) is used to produce a signal for various applications. PWM out signal is generated using the DIO extended features. In this example, we configure the LabJack T7 device to produce a PWM output at DIO0(FIO0). These extended features are explained in detail in the following links:

<https://labjack.com/support/datasheets/t7/digital-io/extended-features>

<https://labjack.com/support/datasheets/t7/digital-io/extended-features/pwm-out>

MatDeck supports LabJack functions which can be used directly inside MatDeck's script to configure LabJack devices. In order to configure and use the device, the LabJack T7 device should be opened in the document. The function `ljdevice_open()` returns the handle of the opened device:

```
1 dev := ljdevice_open("any", "any", "any")
```

PWM Out at FIO0(DIO0) requires the clock source, thus the clock is first configured. There are three parameters to select for the configuration: clock source, clock divisor and the roll value for the given clock. There are three different clocks supported by T7, the most common is clock0 whose frequency is 80MHz. The clock divisor can be any power of two from 1, 2, up to 256. In this example, we use a divisor value equal to one. The roll value is determined according to the desired frequency of the PWM Out signal. For example, if the desired frequency is 1kHz, the roll value is $80\text{Hz}/\text{Divisor}/1\text{kHz}=80000$. Before all values are written to the registers, the clock0 should be disabled.

```
2 //Disable clock0
3 ljdevice_write(dev, "DIO_EF_CLOCK0_ENABLE", 0)
4 //Setup Clock
5 ljdevice_write(dev, "DIO_EF_CLOCK0_DIVISOR", 1)//DIO_EF_CLOCK#_DIVISOR are
6 1,2,4,8,16,32,64,or 256 pre-counter to devide 80MHz
ljdevice_write(dev, "DIO_EF_CLOCK0_ROLL_VALUE", 80000)//Devide 80MHz by
80000 = 1000Hz
```

PWM out at FIO0 (DIO0) is configured by setting 0 at DIO0_EF_INDEX. A duty cycle is set by writing the appropriate value at DIO0_EF_CONFIG_A. If the desired value of the duty cycle is 50%, then DIO0_EF_CONFIG_A will be equal to half of the roll value, which is 40000.

```
7 //PWM Out at DIO0
8 ljdevice_write(dev, "DIO0_EF_ENABLE", 0)
9 //DIO Index valu 0 - PWM set
10 ljdevice_write(dev, "DIO0_EF_INDEX", 0)
11 //PWM duty cycle
12 ljdevice_write(dev, "DIO0_EF_CONFIG_A", 40000)
```

Congured Extended Features should be enabled, this is done by writing one to the ENABLE register:

```
13 ljdevice_write(dev, "DIO_EF_CLOCK0_ENABLE", 1)
14 ljdevice_write(dev, "DIO0_EF_ENABLE", 1)
15 a := ljdevice_last_error("s") //Check for errors in configuration
```

```
a = "LJ_SUCCESS"
```

At the end, the device should be closed.

```
16 ljdevice_close(dev)
```

Configuring PWM Out by using MatDeck GUI Form

All of the above parameters can be set by using `ljdioT7_config_form` in a very intuitive manner. By using this way, all lines of the script code are substituted by GUI settings. The same form can setup all device pins.

```
form := ljdioT7_config_form(o, "Form_DIO_101")
ljdioT7_config_form_configure(form)
```

LabJack - DIO Configuration Form

Device Type: T7 | Connection Type: ANY | Device ID: ANY

Tabs: DIO(0:7) FIO(0:7) | DIO(8:15) EIO(0:7) | DIO(16:19) CIO(0:3) | DIO(20:22) MIO(0:2) | EF Clock Source

DIO0: PWM Out | Desired Freq. (Hz) 1000 | Duty Cicle (%) 50

DIO1: [Dropdown]

DIO2: [Dropdown]

DIO3: [Dropdown]

DIO4: [Dropdown]

DIO5: [Dropdown]

DIO6: [Dropdown]

DIO7: [Dropdown]

Configure

The tab ,EF Clock Source, is used to set the clock source, clock divisor and the roll value for the selected clock. DIO(0:7)-FIO(0:7) is used to set up the PWM output at the selected DIO 0 channel.

Adding Graph to Document

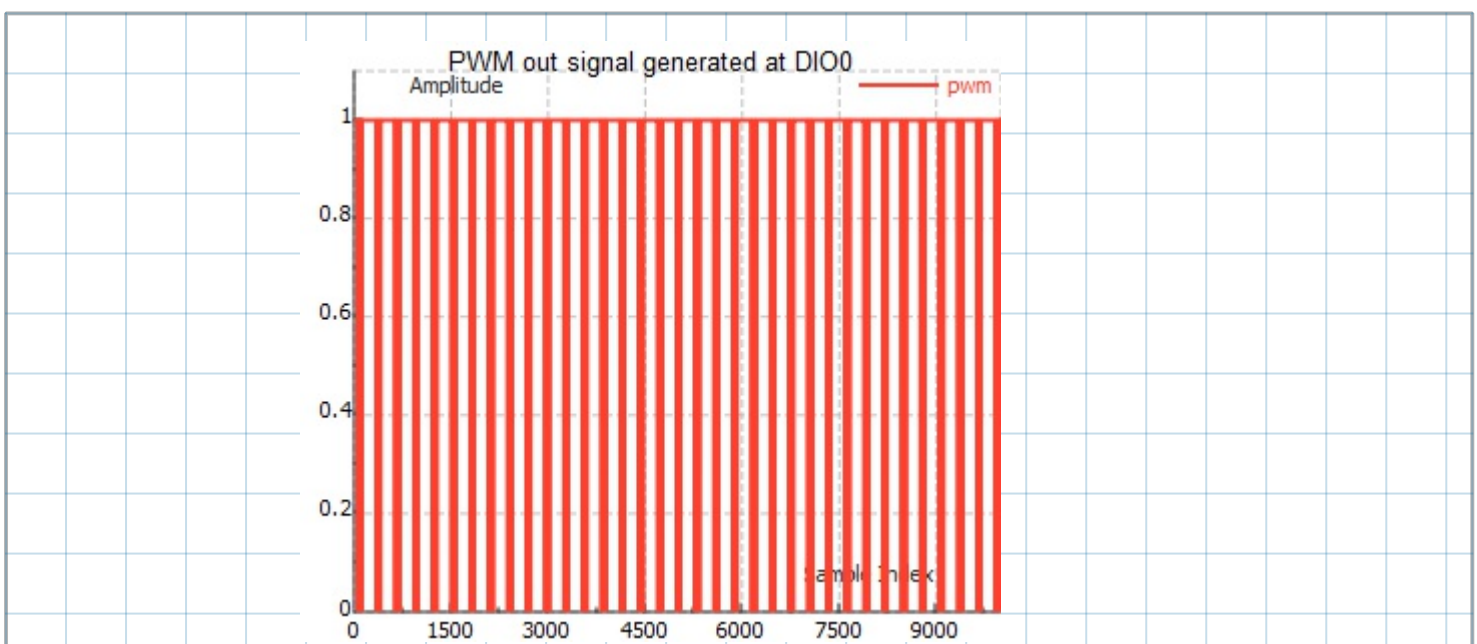
MatDeck provides 2D and 3D graphs which can be used to visualize data. The 2D graph is added from the insert tab. However, before the graph is added, variables for visualization should be calculated. As shown in the next example.

The PWM out at the DIO0 is read in the stream mode. The stream mode is set by writing in the appropriate registers. The function `ljdevice_stream_start()` is used to prepare the device for reading, the user has to specify scans per read, and scan rate. The function `ljdevice_stream_read()` is used to obtain data. If multiple channels are streamed, all the data is stored in a single vector, thus reshape is used to separate the channels.

```
17 // Configure Stream Mode
18 dev := ljdevice_open("any", "any", "any")
19 ljdevice_write(dev, "STREAM_TRIGGER_INDEX", 0)
20 ljdevice_write(dev, "STREAM_CLOCK_SOURCE", 0)
21 ljdevice_stream_start(dev, 10000, ["DIO0"; "STREAM_OUT0"], 10000)
22 pwm := ljdevice_stream_read(dev, 10000)
23 ljdevice_stream_stop(dev)
```

Adding 2D graphs is very simple:

- First check if the canvas is placed in the document where you want to show graph. If it is not there, go to the Insert ribbon tab, click on the Canvas icon and move the cursor/mouse onto the document where you want your canvas to be.
- Then go to the Insert ribbon tab again, click on the Graph icon and place the cursor/mouse onto the canvas where you want your graph to be. A graph without curve will appear.
- Select the graph, right click on the mouse and choose Add Variable, choose X, Y axis variables for the graph. Then on the selected graph, right click and select Properties. In the Graph properties form, choose which variable is the x-axis and which one is the y-axis, fill in other properties and you will a curve will appear on the graph. For more details see the MatDeck user manual, the page under the title Insert Tab and subtitle Graph.



The device should be opened at the beginning of operation, and closed at the end. The device should be reset to default settings to prevent unexpected outputs and inputs.

```
25 ljdevice_write(dev, "IO_CONFIG_SET_CURRENT_TO_DEFAULT", 1)
26 ljdevice_close(dev)
```