```
//This is example how Python can be used within MatDeck
//MatDeck serves as Python editor

//Here begins Python block
#py

"""
Enables a 10 kHz PWM output on FIO0 for the T7 or FIO6 for the T4, enables a
high-speed counter on CIO2 (DIO18), waits 1 second and reads the counter. Jumper
FIO0/FIO6 to CIO2 and the read value. Value should be close to 10000.

DIO extended features, PWM output and high-speed counter documented here:

https://labjack.com/support/datasheets/t-series/digital-io/extended-features
https://labjack.com/support/datasheets/t-series/digital-io/extended-features/pwm-out
https://labjack.com/support/datasheets/t-series/digital-io/extended-features/high-speed-counter

"""
import time

from labjack import ljm


# Open first found LabJack
handle = ljm.openS("ANY", "ANY", "ANY")  # Any device, Any connection, Any identifier
#handle = ljm.openS("T7", "ANY", "ANY")  # T7 device, Any connection, Any identifier
#handle = ljm.openS("T4", "ANY", "ANY")  # T4 device, Any connection, Any identifier
#handle = ljm.open(ljm.constants.dtANY, ljm.constants.ctANY, "ANY")  # Any device, Any connection, Any
identifier

info = ljm.getHandleInfo(handle)
print("Opened a LabJack with Device type: %i, Connection type: %i,\n"
      "Serial number: %i, IP address: %s, Port: %i,\nMax bytes per MB: %i" %
      (info[0], info[1], info[2], ljm.numberToIP(info[3]), info[4], info[5]))

deviceType = info[0]

# Configure the PWM output and counter.
if deviceType == ljm.constants.dtT4:
    # For the T4, use FIO6 (DIO6) for the PWM output
    pwmDIO = 6

    # Set FIO and EIO lines to digital I/O.
    ljm.eWriteNames(handle, 2,
                    ["DIO_INHIBIT", "DIO_ANALOG_ENABLE"],
                    [0xFBF, 0x000])
else:
    # For the T7 and other devices, use FIO0 (DIO0) for the PWM output
    pwmDIO = 0
aNames = ["DIO_EF_CLOCK0_DIVISOR", "DIO_EF_CLOCK0_ROLL_VALUE",
          "DIO_EF_CLOCK0_ENABLE", "DIO%i_EF_ENABLE" % pwmDIO,
          "DIO%i_EF_INDEX" % pwmDIO, "DIO%i_EF_CONFIG_A" % pwmDIO,
          "DIO%i_EF_ENABLE" % pwmDIO, "DIO18_EF_ENABLE",
          "DIO18_EF_INDEX", "DIO18_EF_ENABLE"]
aValues = [1, 8000,
           1, 0,
           0, 2000,
           1, 0,
           7, 1]
numFrames = len(aNames)
results = ljm.eWriteNames(handle, numFrames, aNames, aValues)

# Wait 1 second.
time.sleep(1.0)

# Read from the counter.
value = ljm.eReadName(handle, "DIO18_EF_READ_A")

print("\nCounter = %f" % (value))

# Turn off PWM output and counter
aNames = ["DIO_EF_CLOCK0_ENABLE", "DIO%i_EF_ENABLE" % pwmDIO]
aValues = [0, 0]
numFrames = len(aNames)
results = ljm.eWriteNames(handle, numFrames, aNames, aValues)

# Close handle
ljm.close(handle)

###
// Here is the end of Python block
```