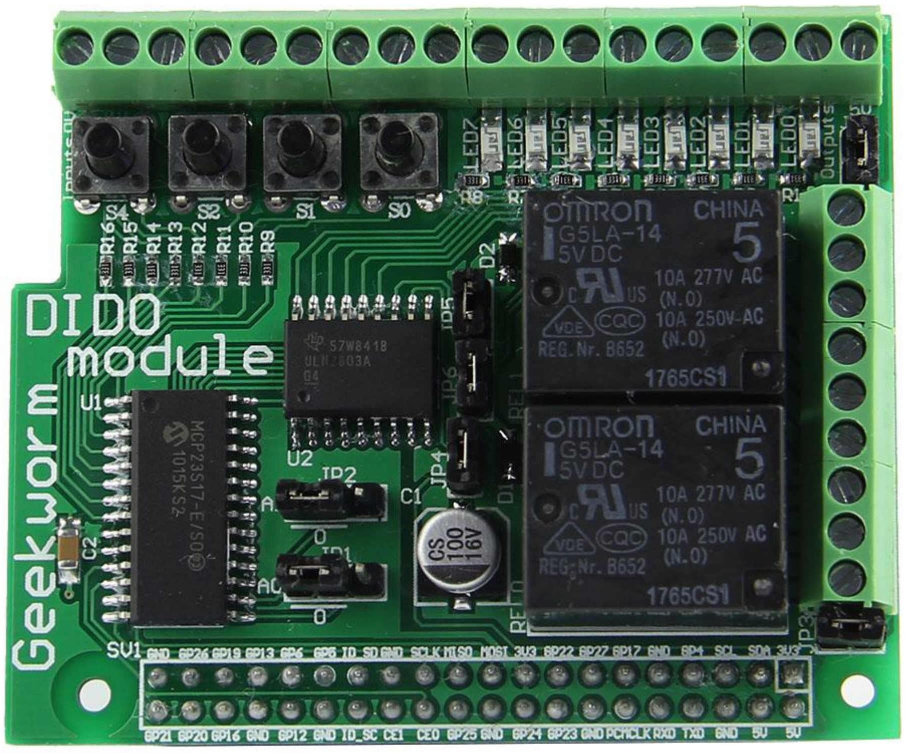
Install DIDO module Software Updating Raspbian



To ensure that you can easily install all the DIDO module software and make sure everything works, you should first update your Raspbian image. Don't Panic! This doesn't mean downloading a new image and rewriting your SD.

Log into your Pi and in the command line, type the following commands.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

When prompted enter `y`. These commands can take a while to run.

Enabling SPI

DIDO module products need the SPI pins on the Raspberry Pi to be enabled in order to function, this is simple enough to do. To enable the SPI, you must update to the latest version of the raspi-config tool, so open a Terminal and use the command

```
sudo apt-get install raspi-config
```

and enter `Y` when required.

Now start raspi-config by typing

```
sudo raspi-config
```

To view the list of advanced configuration options, select `Option 8 Advanced Options`.
Choose the `A5 SPI` option. Set this to `"Yes"`
Select `"Ok"` then `"Finish"`

Troubleshooting: If your raspi-config command did not work, or the SPI option was not listed, then make sure you have updated your copy of Raspbian With this.

Which Python is for me?

You may have heard people talking about different versions of Python and be a bit confused by it all, not to worry. We are here to set you straight and help you choose the right Python for use with your DIDO module.

Difference between Python3 and Python

There are two major versions of python that are in common use today, Python 3 and Python 2 (which is simply known as Python). Python 2 is the most widely used of these two though the Python community has been moving towards python 3 more and more. The issue is that their is limited forwards and backwards compatibility between these two versions of Python, so most Python 2 files have to be ran with the Python command and Python3 files with the Python3 command. This can be a problem if you are wanting to use multiple packages that only has releases for the different versions of Python.

Which should I use?

We have develop python 2 and 3 versions of the DIDO module software to facilitate a wider range of compatibility with other Python modules. If you are using other modules in your DIDO module programs, you should check which versions of Python your modules have releases for and choose the Python that is most compatible with your needs, for example, the Pygame module that is often used in RaspberryPi Python programs is currently only available as a Python 2 release. If you have no Python version specific modules and you are new to Python, then it is probably best to start with Python 2 as there are more guides and tutorials referring to it. Once you have picked your poison, make sure you are clear on the nuances of your version.

Example difference

One of the differences between the Pythons is the print statement:

Python 2 `print “Hello world!”`

Python 3 `print (“Hello world!”)`

There are many subtle differences between the Pythons and it is good to be aware of them.

Installing DIDO module modules

With the latest version of Raspbian installed you can easily install DIDO module software with apt-get.

To install the DIDO module software run the command:

```
sudo apt-get install python3-pifacedigital-emulator
```

Enter `Y` when prompted.

If you want to use Scratch with your DIDO module, you will need to install the DIDO module Scratch handler:

```
sudo apt-get install python3-pifacedigital-scratch-handler
```

Now reboot your Raspberry Pi

```
sudo reboot
```

Testing your DIDO module

To test all your DIDO module software has installed and your RaspberryPi is set up correctly, type one of the following commands into a Terminal:

```
python3 /usr/share/doc/python3-pifacedigitalio/examples/blink.py
```

Or for Python 2:

```
python /usr/share/doc/python-pifacedigitalio/examples/blink.py
```

Press `Ctrl c` to close the blink program.

DIDO module with python

Python code for DIDO module products can be written in two ways, using a python interpreter shell or by writing and saving your program with a text editor.

- **Python interpreter shell**

  Start a new python interpreter by typing `python` or `python3` (depending on whether you have installed DIDO module software for python or python3) into a Terminal. Type your python commands after you see the python prompt `>>>`. The commands will execute one by one as you enter them, unless they are contained inside control statements such as `while` and `if`.

  This is a great way to test bits of code and explore the different features of DIDO module products.

- **Saved Python program**

  To create a python program type

  ```
  touch yourProgramName.py
  ```

  Open this file in your favourite text editor, eg:

  ```
  leafpad yourProgramName.py
  ```

  Write your code in this file and Save it. Then run it with the command

  ```
  python yourProgramName.py
  ```

First steps with DIDO module and Python

To use DIDO module with Python, import the pifacedigital module:

```
import pifacedigitalio as p
```

Before use, the board must be initialised with a call to init.

```
p.init()
```

There are three main functions to control the interface

- `digital_read(pin_number)`
  - returns 1 or 0 depending on the state of the input numbered pin_number
- `digital_write(pin_number, state)`
  - sets the output numbered pin_number to state 0 or 1. State 1 turns the LED on and enables to open collector to sink current
- `digital_write_pullup(pin_number, state)`

- ○ sets a 10k pullup on input numbered pin_number to be state 0 or 1. State 1 is pullup enabled

Simple python examples

One of the quickest ways to get going is to look at examples of DIDO module in use. These can be ran by moving into the DIDO module examples directory

```
cd /usr/share/doc/python3-pifacedigitalio/examples
```

Type `ls` to view the contents of this directory and run the examples like so

```
python3 blink.py
```

Controlling an output (turn a relay on)

The first two outputs control the two on-board relays. To turn the first relay on, start a new python interpreter and type the following:

```
import pifacedigitalio as p

p.init()

p.digital_write(0,1)
```

Flashing an LED

```
from time import sleep

import pifacedigitalio as p

pinit()

while(True):

    p.digital_write(0,1) #turn on     sleep(1)

    p.digital_write(0,0) #turn off

    sleep(1)
```

Reading an input

To read the state of an input use the `p.digital_read(pin)` function as shown below. If a button is pressed the function returns a 1, otherwise it returns a 0.

```
import pifacedigitalio as p
```

```
p.init()
```

```
p.digital_read(1)
```

- Python prints *0*.
- Now hold down switch number 1 (marked S1) and type `p.digital_read(1)` again.
- Python prints *1*.