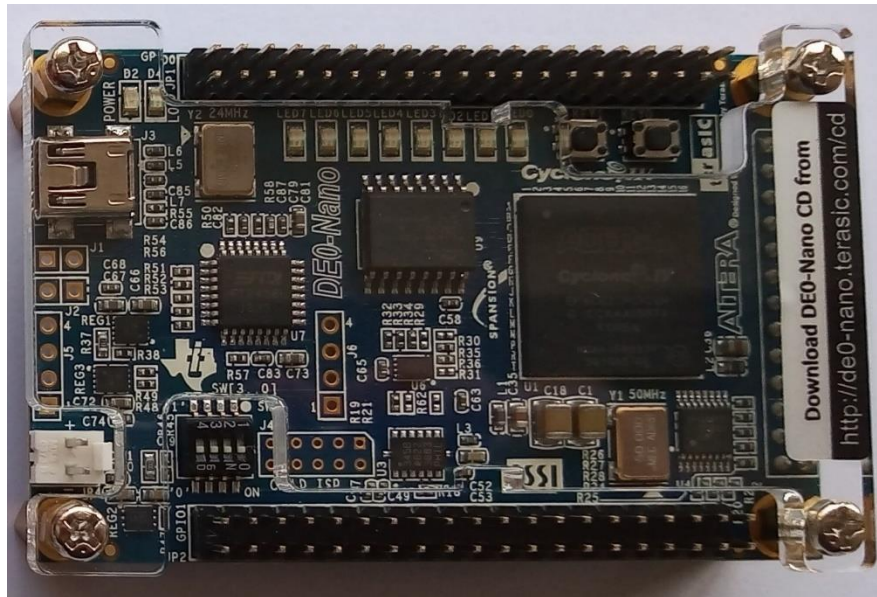


TRIGGER solution for LIME: status

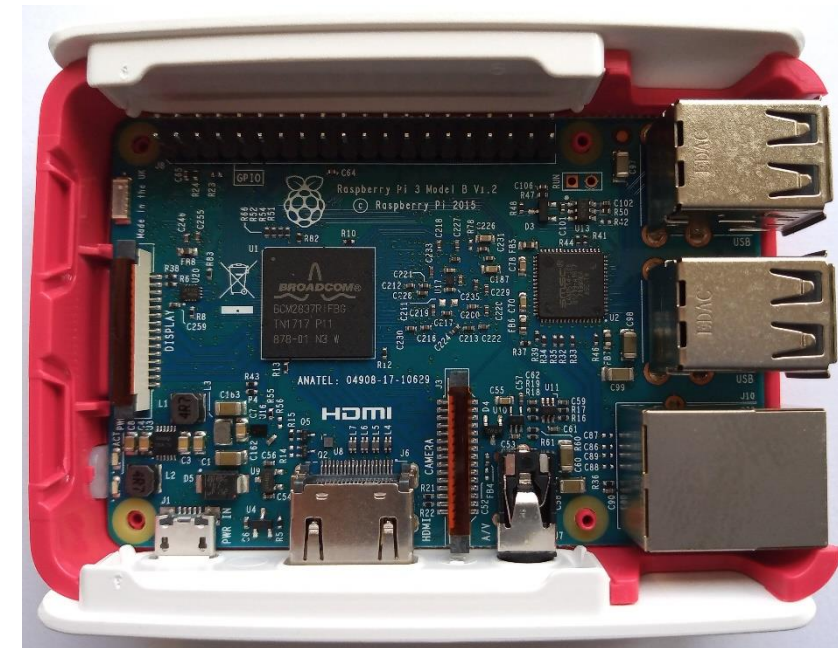
Herman Pessoa Lima Jr

14 Jul 2021

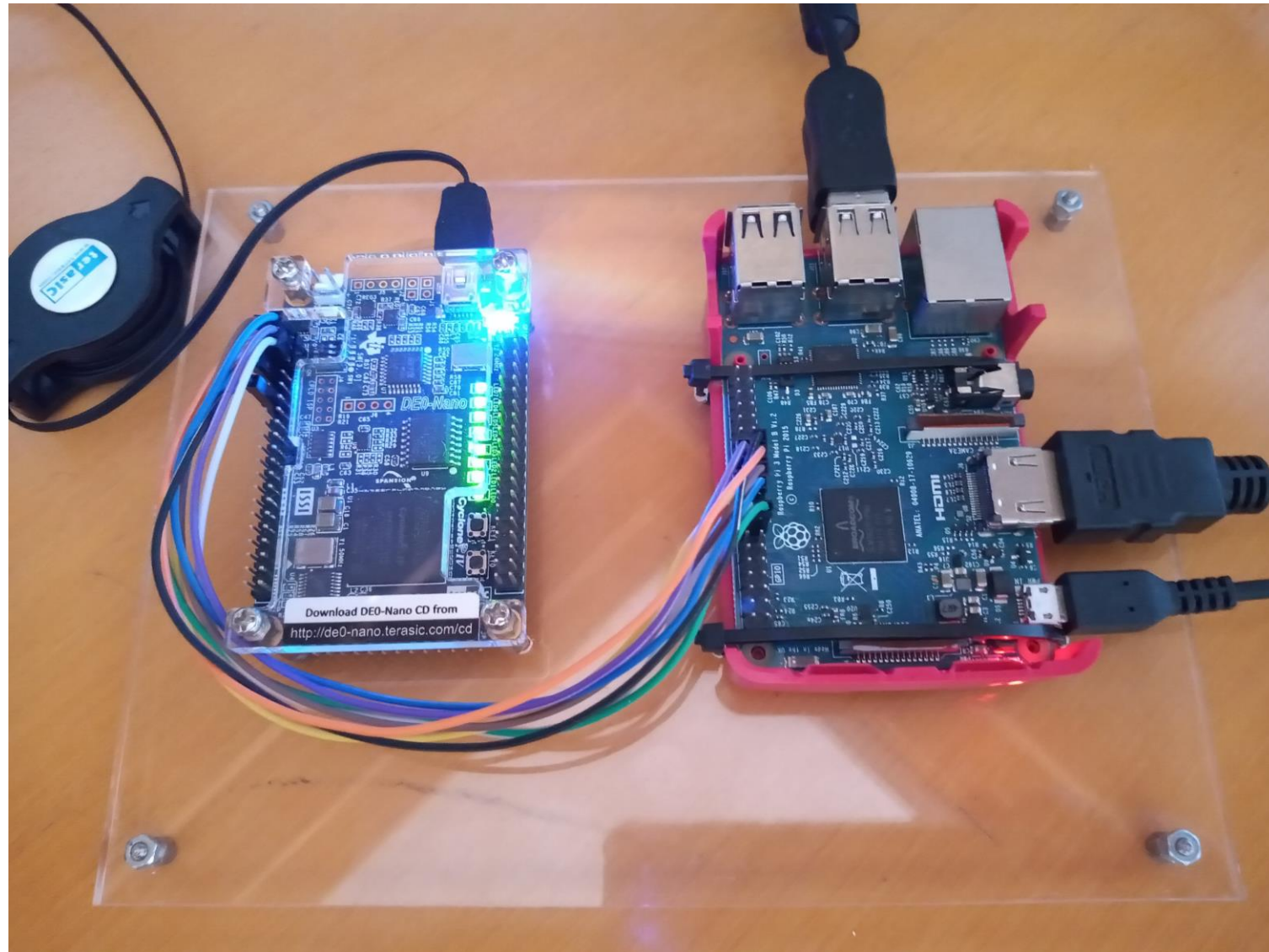
The boards in use for development and tests.



Terasic DE0 Nano FPGA Kit (US\$86)
Cyclone IV EP4CE22F17C6N

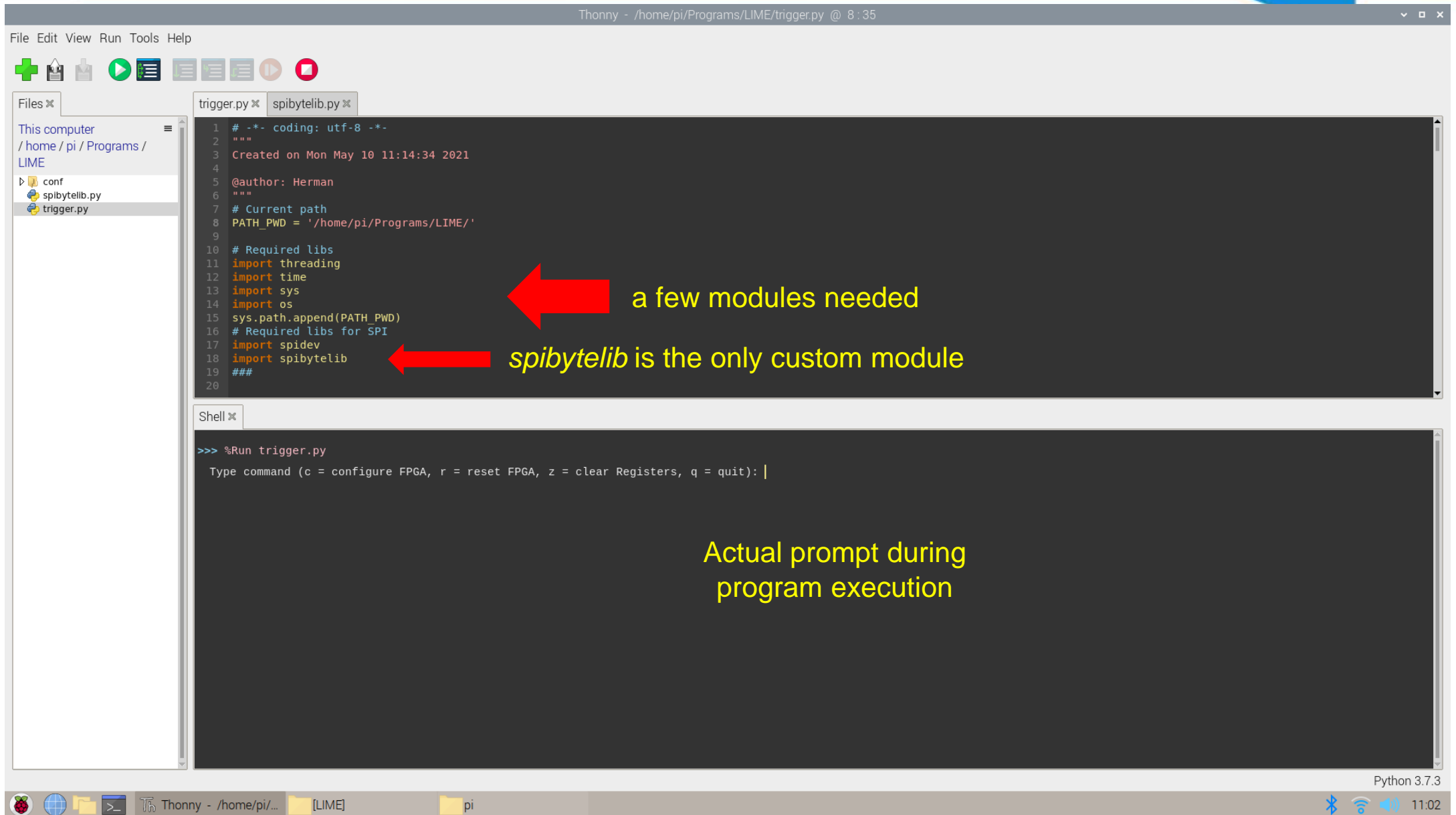


Raspberry Pi 3 Model B (US\$43)



Development setup

- ~~Pin assignment modifications for the new FPGA kit. **DONE**~~
- ~~Check software compatibility for the new RaspberryPI. **DONE**~~
- ~~Make the connections between the FGPA kit and the RaspPI. **DONE**~~
- ~~Check the SPI communication between the RaspPI and the FPGA kit. **DONE**~~
- No more Socket programming and State machines. As simple as possible code. **DONE**
- SPI Write operations to configure control registers in FPGA. **DONE**
- Design a simple Trigger Logic inside the FPGA (AND, OR gates). **ONGOING...**
- Test the system with external pulses (functionality). **15-16/07**
- Write a short manual to facilitate the duplication of such system in Rome. **19-23/07**
- Assemble the boards and connectors in a single mechanical frame (VME?, NIM?). (in Rome)



The screenshot shows the Thonny IDE interface. The top bar indicates the file path: Thonny - /home/pi/Programs/LIME/trigger.py @ 8:35. The menu bar includes File, Edit, View, Run, Tools, and Help. The toolbar contains icons for file operations and execution. The left sidebar shows the file explorer with the following structure:

- This computer
- /home/pi/Programs/LIME
- conf
- spibytelib.py
- trigger.py

The main editor displays the code for `trigger.py` and `spibytelib.py`. The code is as follows:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon May 10 11:14:34 2021
4
5 @author: Herman
6 """
7 # Current path
8 PATH_PWD = '/home/pi/Programs/LIME/'
9
10 # Required libs
11 import threading
12 import time
13 import sys
14 import os
15 sys.path.append(PATH_PWD)
16 # Required libs for SPI
17 import spidev
18 import spibytelib
19 ###
20
```

Two red arrows point to the `import` statements in the code. The first arrow points to lines 11-14, with the text "a few modules needed". The second arrow points to lines 17-18, with the text "spibytelib is the only custom module".

The Shell window shows the execution of the program:

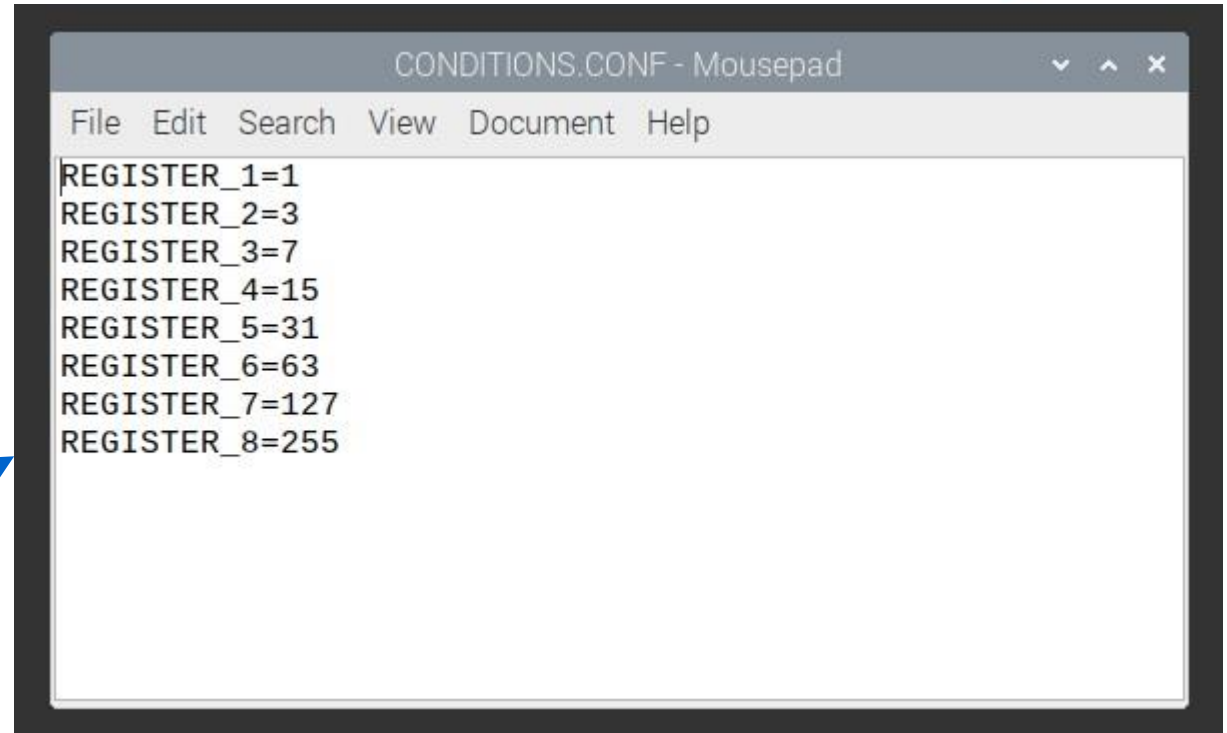
```
>>> %Run trigger.py
Type command (c = configure FPGA, r = reset FPGA, z = clear Registers, q = quit): |
```

The text "Actual prompt during program execution" is overlaid on the shell window.

The bottom status bar shows the system tray with icons for Raspberry Pi, network, and volume, along with the text "Thonny - /home/pi/... [LIME] pi" and "Python 3.7.3 11:02".

```
7 # Current path
8 PATH_PWD = '/home/pi/Programs/LIME/'
9
10 # Required libs
11 import threading
12 import time
13 import sys
14 import os
15 sys.path.append(PATH_PWD)
16 # Required libs for SPI
17 import spidev
18 import spibytelib
19 ###
20
21 # Files Control
22 #FILE_DATA_LOG_PATH = PATH_PWD + "log/DATA_LOG.LOG"
23 #FILE_STAT_LOG_PATH = PATH_PWD + "log/STATUS_LOG.LOG"
24
25 FILE_CONFIG_FILE_PATH = PATH_PWD + "conf/CONDITIONS.CONF"
26
```

```
98 # Main Process
99 while True:
100     #print("STATE is: ", STATE_CONTROL)
101     msg = input("Type command (c = configure FPGA, r = reset FPGA, z = clear
102     if msg == 'c':
103         #####
104         # Collect the CONFIG DATA and RESET the VALUE #
105         # Prepare the data to SEND #
106         #####
107         # 1. Load the Config File
108         conf_loaded = spibytelib.LOAD_CONF_FILE (FILE_CONFIG_FILE_PATH)
109         # 2. Mount the Packet Frame
110         frame_mounted = spibytelib.MOUNT_PKT_FROM_DECFILE (conf_loaded)
111         # 3. FULL Packet Mounted TX
112         #FULL_FRAME_PKT = spilibx.TX_FULL_PACKET_TO_FPGA (frame_mounted)
113         FULL_FRAME_PKT = frame_mounted
114         #
```



A screenshot of a terminal window titled "CONDITIONS.CONF - Mousepad". The window shows a configuration file with the following content:

```
REGISTER_1=1
REGISTER_2=3
REGISTER_3=7
REGISTER_4=15
REGISTER_5=31
REGISTER_6=63
REGISTER_7=127
REGISTER_8=255
```

A blue arrow points from the line `FILE_CONFIG_FILE_PATH = PATH_PWD + "conf/CONDITIONS.CONF"` in the first code block to this window.

Currently defined:

- 8 control registers (1 Byte each)
- All registers configured at once by “c” command
- Quantity of registers easy to increase (firmware and software modifications needed)