



Introduction to Pygmy



G. V. V. Sharma *

Contents

1	Software	1
2	Setup	1
3	Frequency	2

Abstract—This document provides a simple introduction to software and hardware using the Pygmy FPGA/microcontroller board. The exercises provided here are suitable for students from primary school till college.

1 Software

Download codes from the following link

<https://github.com/gadepall/vaman/tree/master/fpga/setup/codes/>

2 Setup

- 2.1. Connect the Pygmy to the Raspberry Pi through USB.
- 2.2. There is a button and an LED to the left of the USB port on the Pygmy. There is another button to the right of the LED.

*The author is with the Department of Electrical Engineering, IIT Hyderabad, 502285, India. email: gadepall@ee.iith.ac.in. All material in this document is released under GNU GPL. Free to use for anything.

- 2.3. Press the right button first and immediately press the left button. The LED will be blinking green. The Pygmy is now in bootloader mode.
- 2.4. Login to termux-ubuntu on the android device and execute the following commands

```
cd /storage/emulated/0/Download
svn co https://github.com/gadepall/vaman/trunk/fpga/setup/codes/blink
ql_symbiflow -compile -src /storage/emulated/0/Download/blink -d ql-eos-s3 -P PU64 -v helloworldfpga.v -t \helloworldfpga -p quickfeather.pcf -dump binary
scp /storage/emulated/0/Download/blink/helloworldfpga.bin pi@192.168.0.114:
```

Make sure that the appropriate IP address for the raspberry pi is given in the above command.

- 2.5. Now execute the following commands on the raspberry pi.

```
python3 /home/pi/TinyFPGA-Programmer-Application/tinyfpga-programmer-gui.py --port /dev/ttyACM0 --
```

```
appfpga /home/pi/
helloworldfpga.bin --mode
fpga
```

- 2.6. Make sure that the correct USB port address is given in the above command. Then press the button to the right of the USB port. After some time, the LED will start blinking red.

3 Frequency

- 3.1. In the following verilog program,

```
codes/blink/helloworldfpga.v
```

pay attention to the following lines

```
delay = delay+1;
if(delay > 20000000)
begin
delay=27'b0;
led=!led;
end
```

It may be deduced from the above that the blink frequency is 20 MHz.

- 3.2. In instruction 3.1, replace

```
if(delay > 20000000)
```

with

```
if(delay==27'
b1001100010010110100000000)
```

and execute the verilog code.

- 3.3. Since the delay is 20 MHz, the blink period is 1 second. Modify the verilog code so that the blink period becomes 0.5s.
- 3.4. Find the bit length of 20 MHz.
Solution:

$$\log_2(20000000) \approx 27 \quad (3.4.1)$$

- 3.5. Obtain the above answer using a Python code.

Solution: Execute the following code and compare with instruction 3.2.

```
codes/blink/freq_count.py
```

- 3.6. Replace the following line in the code in instruction 3.1

```
assign redled = led; //If you
want to change led colour to
red,
```

Type	Pygmy Pin	Connection
Input	IO_28	GND
Output	IO_11	LED

TABLE 3.8.1: Pygmy Input/Output.

with

```
assign blueled = led;
```

and execute the code.

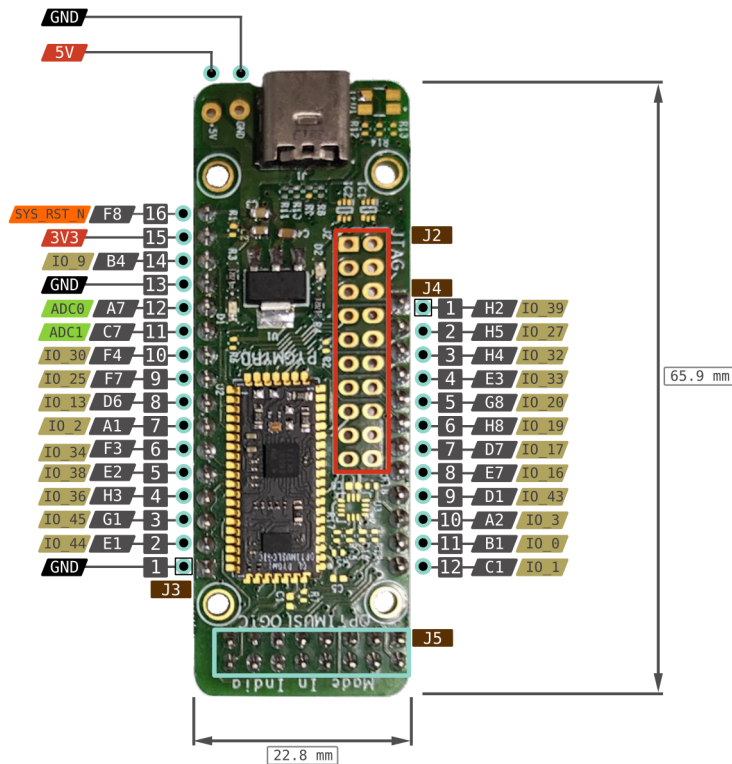
- 3.7. Ensure that the LED stays on in green colour. Solution: Execute the following code

```
codes/blink/onoff.v
```

- 3.8. Using Table 3.8.1 and Fig. 3.8.1, control the onboard LED through an external input. Connect an external LED and control it using an output pin as well.
Solution: Execute the following code and take out the input pin connect to GND. Plug it again. Do this repeatedly.

```
codes/input/blink_ip.v
codes/input/pygmy.pcf
```

PYGMY BB v1 PINOUT



- Reset
- Power
- GND
- I/O/Pad Number
- Physical Pin/Ball
- Analog Input
- Pin Function(s)
- Component Pin
- Board Header Pin

On-Board Components

SPI FLASH Memory [on Pygmy Stamp]

SS	IO_39	H2	SPI_MASTER SSn1
SCLK	IO_34	F3	SPI_MASTER_CLK
SI	IO_38	E2	SPI_MASTER_MOSI
SO	IO_36	H3	SPI_MASTER_MISO

Buttons

USR	IO_6	B3	GPIO[0]
-----	------	----	---------

RGB LED

RED	IO_22	G7	GPIO[6]
GREEN	IO_21	H7	GPIO[5]
BLUE	IO_18	E8	GPIO[4]

BMI160 ACCEL + GYRO

SCx	IO_0	B1	SCL_0
SDx	IO_1	C1	SDA_0

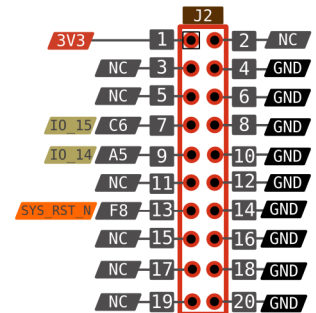
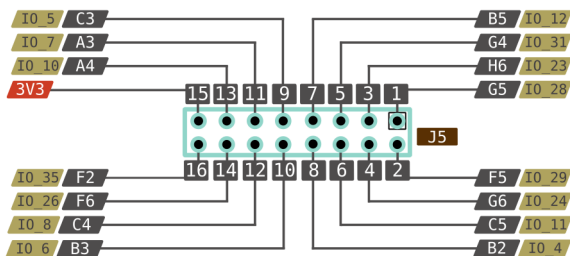


Fig. 3.8.1: