Xylo™IMU
Dev Kit
Datasheet
Sep 2023

The information contained herein is for informational purposes only, and is subject to change without notice.

Intellectual Property Rights

SynSense owns the copyrights, trademarks and other intellectual property rights and interests in this document. The fact that SynSense provides this document to you does not affect the rights and interests of SynSense as described above.
Brand and product names are trademarks or registered trademarks of their respective owners.
No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document.

No Warranty

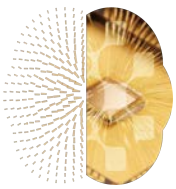
While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and SynSense is under no obligation to update or otherwise correct this information. SynSense makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of SynSense hardware, software or other products described herein.

Disclaimer

To the extent permitted by applicable law, SenSense shall not be liable for any direct, indirect, incidental, special, incidental or other damages, costs, liabilities or claims of any kind arising out of or in connection with the use of this document,with respect to the operation or use of SynSense hardware, software or other products described herein.

Applicable Terms and Conditions for Products

Terms and limitations applicable to the purchase or use of SynSense's products are as set forth in a signed agreement between you and SynSense or in SynSense's Standard Terms and Conditions.



Content

1. Introduction 1
    1.1. Xylo™IMU 1
    1.2. Xylo™IMU Dev Kit 2
2. Features 3
    2.1. Xylo™IMU chip 3
    2.2. Hardware Development Kit (HDK) 3
    2.3. What is on Xylo™IMU 4
        2.3.1 IMU input interface 4
        2.3.2 Xylo™ Spiking Neural Network (SNN) inference core 5
    2.4. About the Xylo™ family 5
3. Electrical characteristics 6
4. Mechanical specification 7
    4.1. Mother board 7
    4.2. Daughter board 10
5. Getting started 12
    5.1. Install Rockpool 12
    5.2. Rockpool documentation for Xylo™IMU 12
    5.3. USB configuration for Linux 12
    5.4. Connecting to the Xylo™IMU from Python using Rockpool 12
6. Change log 14

1. Introduction

1.1. Xylo™IMU

Xylo™IMU is a fully digital AI chip which consists of a dedicated IMU sensor data acquisition and conversion block (IMU interface, supporting the MC3632 IMU sensor) and a low-power inference core (Xylo™ Core), using spiking neural networks (SNNs) for real-time sensory inference.

The IMU interface will read from an external IMU sensor via an SPI master interface, convert IMU sensory data to events, and send the events to Xylo™ Core for inferences. Xylo™ Core is a highly configurable SNN processing core, supporting a wide range of flexible network architectures (feed-forward, recurrent, residual, etc.).

Xylo™IMU supports up to 16 input channels; input projection weight dimensions of up to 16x128; 496 configurable hidden neurons; output projection weight dimensions of up to 128x16; and up to 16 readout channels for classification purposes. Xylo™IMU provides computationally powerful current-based Leaky Integrate and Fire (CUBA-LIF) spiking neurons, with fully-configurable per-synapse leak, bias and threshold parameters.

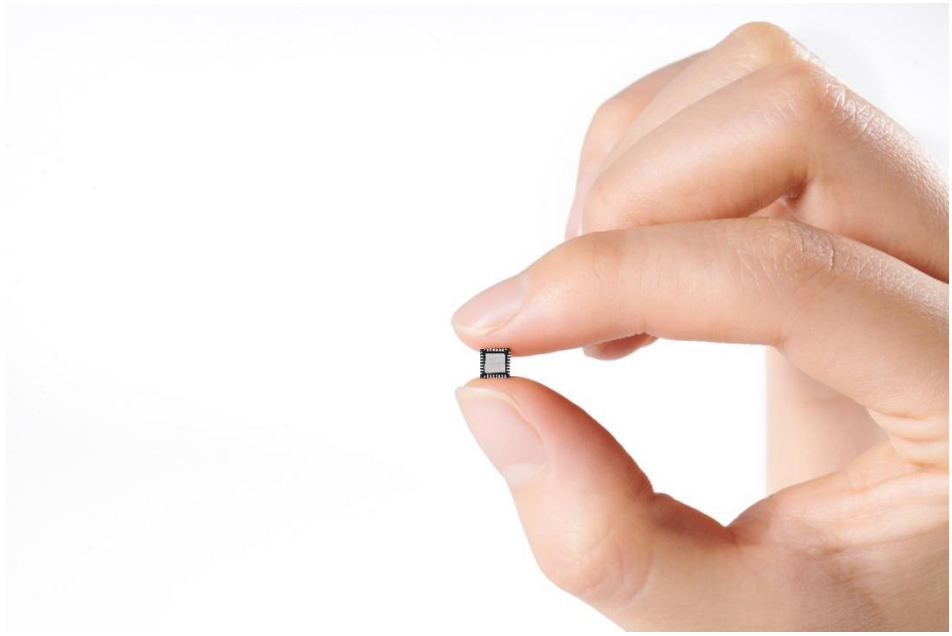


Figure 1 . Xylo™IMU

1.2. Xylo™IMU Dev Kit

The Xylo™IMU Dev Kit is powered by the SynSense Xylo™IMU chip, which brings the flexibility of highly configurable SNN processing to microwatt energy budgets. With appropriate network design and training, Xylo™IMU supports a range of IMU-based battery-powered applications, for example:

- Human behavior detection
- Fall detection
- Livestock monitoring
- etc.

Development of these application prototypes is straightforward, using the SynSense python-based toolchain Rockpool.

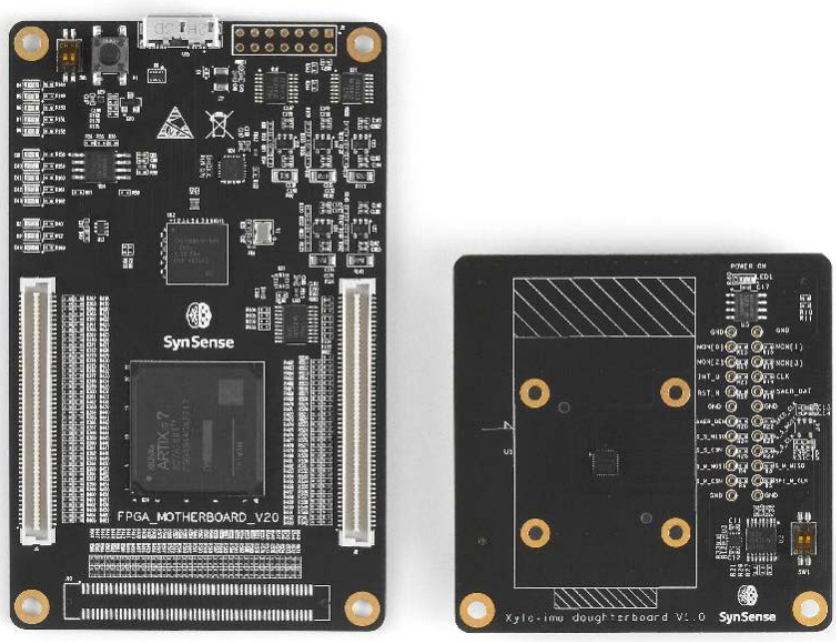


Figure 2 . Xylo™IMU Dev Kit

2. Features

2.1. Xylo™IMU chip

- 4-wire SPI slave interface for register and RAM config, support both single and burst
- Mode access
- 4-wire SPI master interface for external IMU access
- Support up to ~500 reservoir neurons and up to 16 classifications
- 1-bit configurable interrupt indicating classification / detection done
- 4-bits monitor pins for debug purpose, internal debug registers available
- Up to 50 MHz internal operating frequency
- Extremely low memory footprint (~75 KB), memory power control granularity down to 2 KB
- Ultra-low average working power consumption (<300 uW)
- 4 mm x 4 mm QFN-32 package

2.2. Hardware Development Kit (HDK)

- Contains one Xylo™IMU inference processor
- 3-axis Accelerometer IMU on board(MC3632), ODR of 400Hz
- USB bus powered, USB 3.1 for interfacing with a host PC
- High-level Python API (Rockpool) for application development, HDK interfacing and deployment
- On board power monitoring of VDD_IO, VDD_CORE of Xylo™IMU
- Selectable IMU data source: on board IMU sensor or streaming data from a host PC

2.3. What is on Xylo™IMU

Xylo™IMU consists of an efficient IMU sensor data encoding interface, coupled with the Xylo™ SNN core for inference.

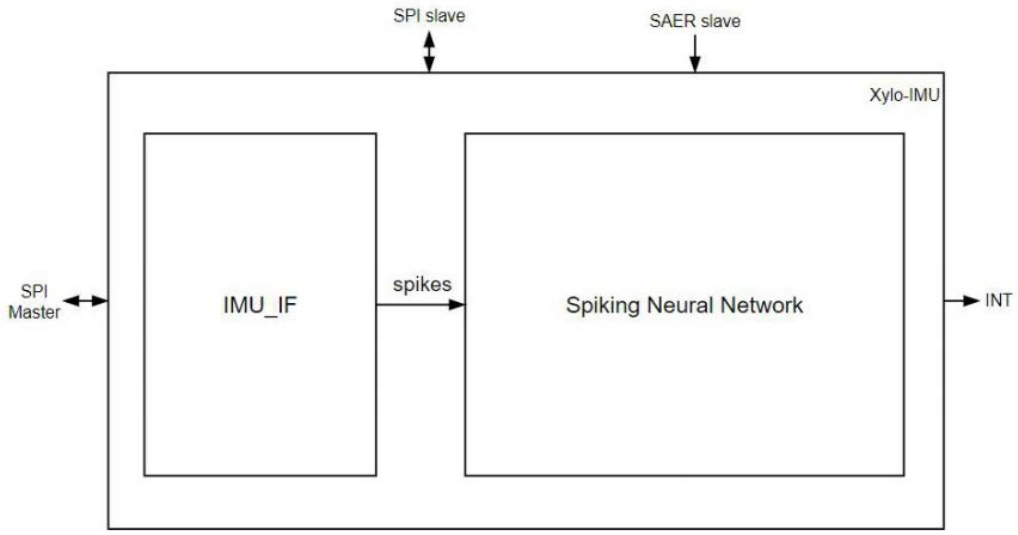


Figure 3 . Xylo™IMU block diagram

2.3.1 IMU input interface

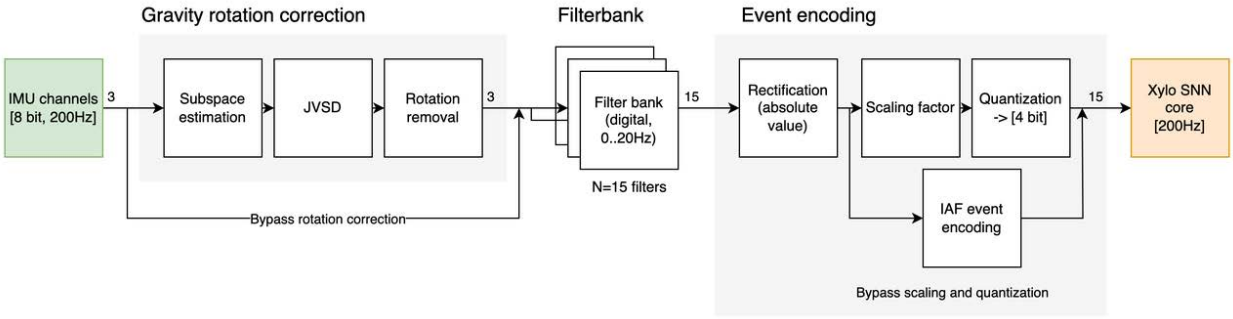


Figure 4 . IMU input interface

Xylo™IMU contains an encoding interface for connecting directly with a MEMS 3-channel force sensor, and converting the acceleration signals into event streams for inference by the Xylo™ SNN core. The IMU encoding interface provides several selectable and configurable preprocessing steps:

- Orientation estimation and correction with respect to gravity [optional]
- Configurable filter bank: Up to 5 bands per IMU channel, 0–100Hz
- Selectable quantisation or IAF encoding pathways
    - Scaling and quantization up to 15 events per time-step per channel
    - High-rate integrate-and-fire (IAF) direct event encoding

2.3.2 Xylo™ Spiking Neural Network (SNN) inference core

The Xylo™ SNN core performs inference on a stream of sensory data. It provides up to 16 channels of quantised event-encoded input, and 32 channels of binary event-encoded output.

- Convenient training, quantisation and configuration with rockpool.ai
- 512 leaky integrate-and-fire (LIF) model spiking neurons
  - 480 LIF hidden neurons
  - 32 LIF output neurons
- 15 events per time-step for input channels and hidden neurons
- 1 event per time-step for output neurons / output channels
- Maximum fan-in of 64 per hidden neuron
- Mean fan-in of 32 per hidden neuron
- Maximum input projection to 128 hidden neurons
- Maximum output projection from 128 hidden neurons

2.4. About the Xylo™ family

Xylo™ is a family of ultra-low-power sensory processing and classification devices, providing efficient sensor interfaces for a range of modalities, and relying on low-bit-depth, sparse neural networks for efficient inference.

Xylo™ devices comprise an efficient sensory encoding front-end, a Xylo™ SNN inference core with on-board memory, and control logic.

Xylo™ is currently available in several versions with varying HW support and front-ends.
The various devices in the family interface directly with analog and digital audio sensors (Xylo™Audio); IMU motion devices (Xylo™IMU); and other sensory input modalities.

3. Electrical characteristics

Feature	Spec.	Units
Core Supply Voltage	1.1	V
IO Supply Voltage	2.5 / 3.3	V
Operating Temperature	-40 ~ 125	℃
Digital Input / Output High Level	2.5 / 3.3	V
Digital Input / Output Low Level	0	V
Clock Frequency	50	MHz
Power Consumption	< 300	μW

4. Mechanical specification

4.1. Mother board

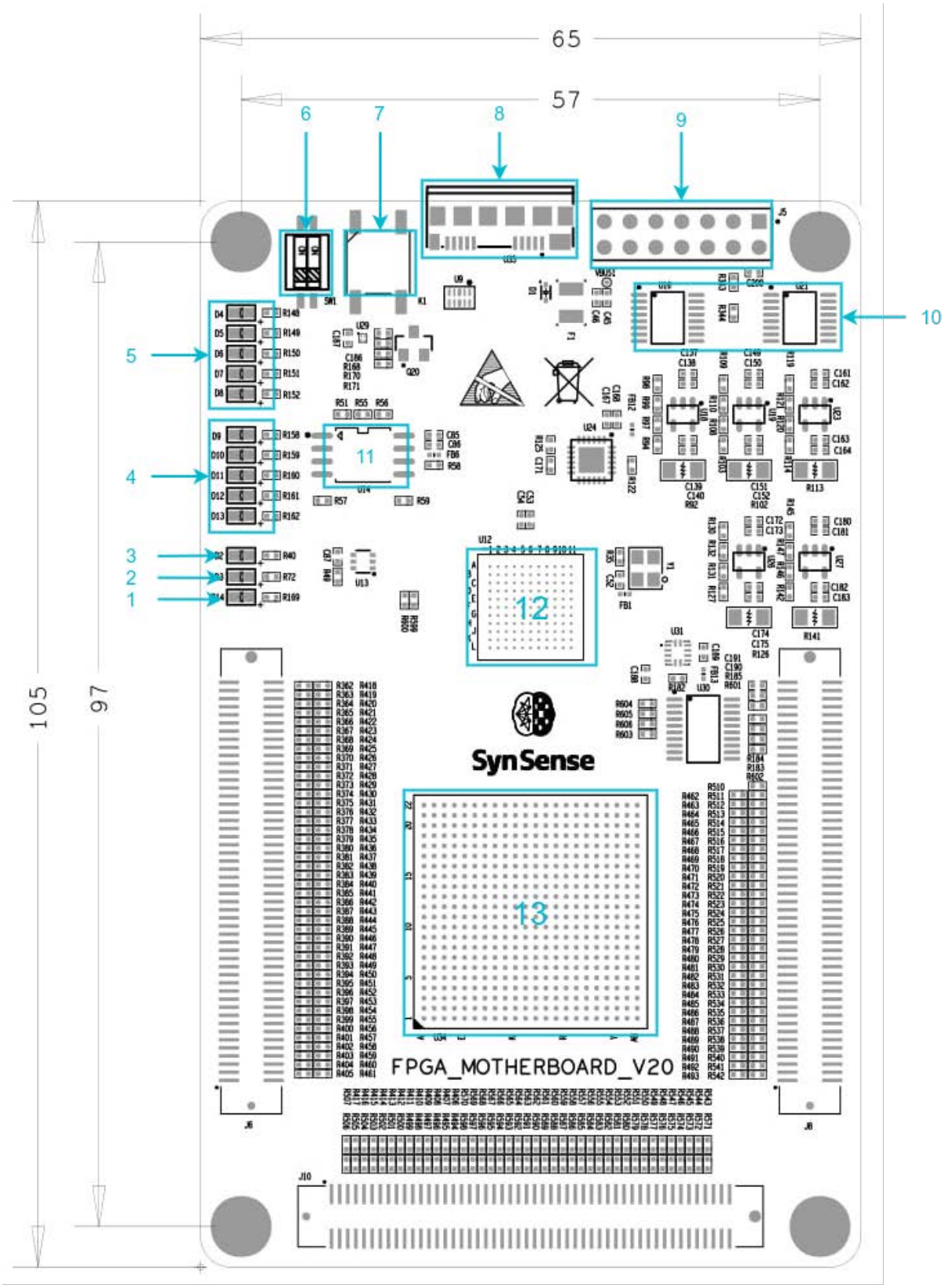


Figure 5 . Front view of the FPGA mother board (mm)

1. System Power LED
2. FPGA CFG Done Indicator
3. USB3.0 Controller State Indicator
4. SoC Power Traces State LEDs
5. Debug State Indicator
6. USB3.0 Controller CFG Switch (RSV)
7. System Reset Key
8. USB3.0 Micro-B Port
9. FPGA JTAG (RSV)
10. High Precision Power Monitor
11. Flash
12. USB3.0 Controller
13. FPGA

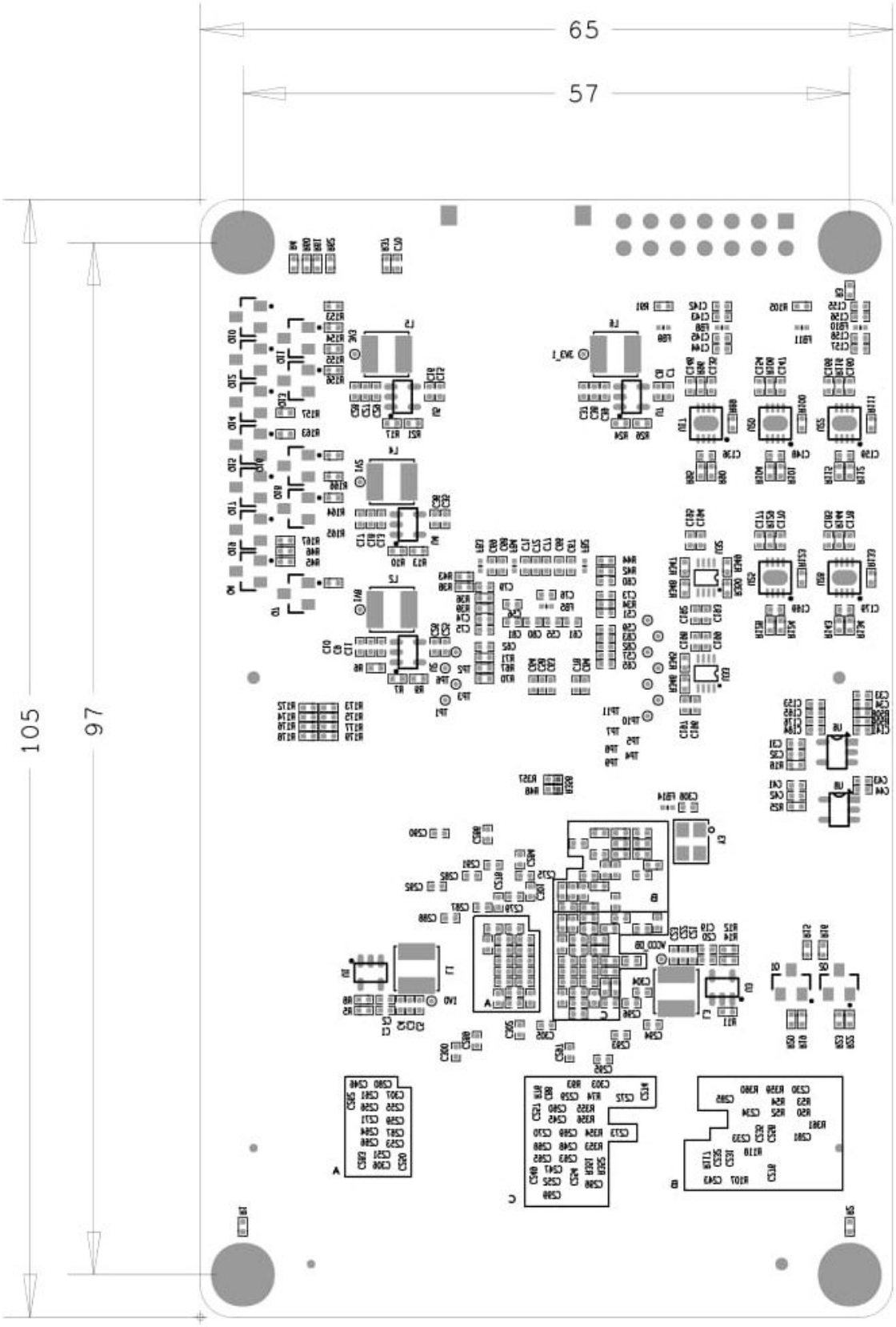


Figure 6 . Bottom view of the FPGA mother board (mm)

4.2. Daughter board

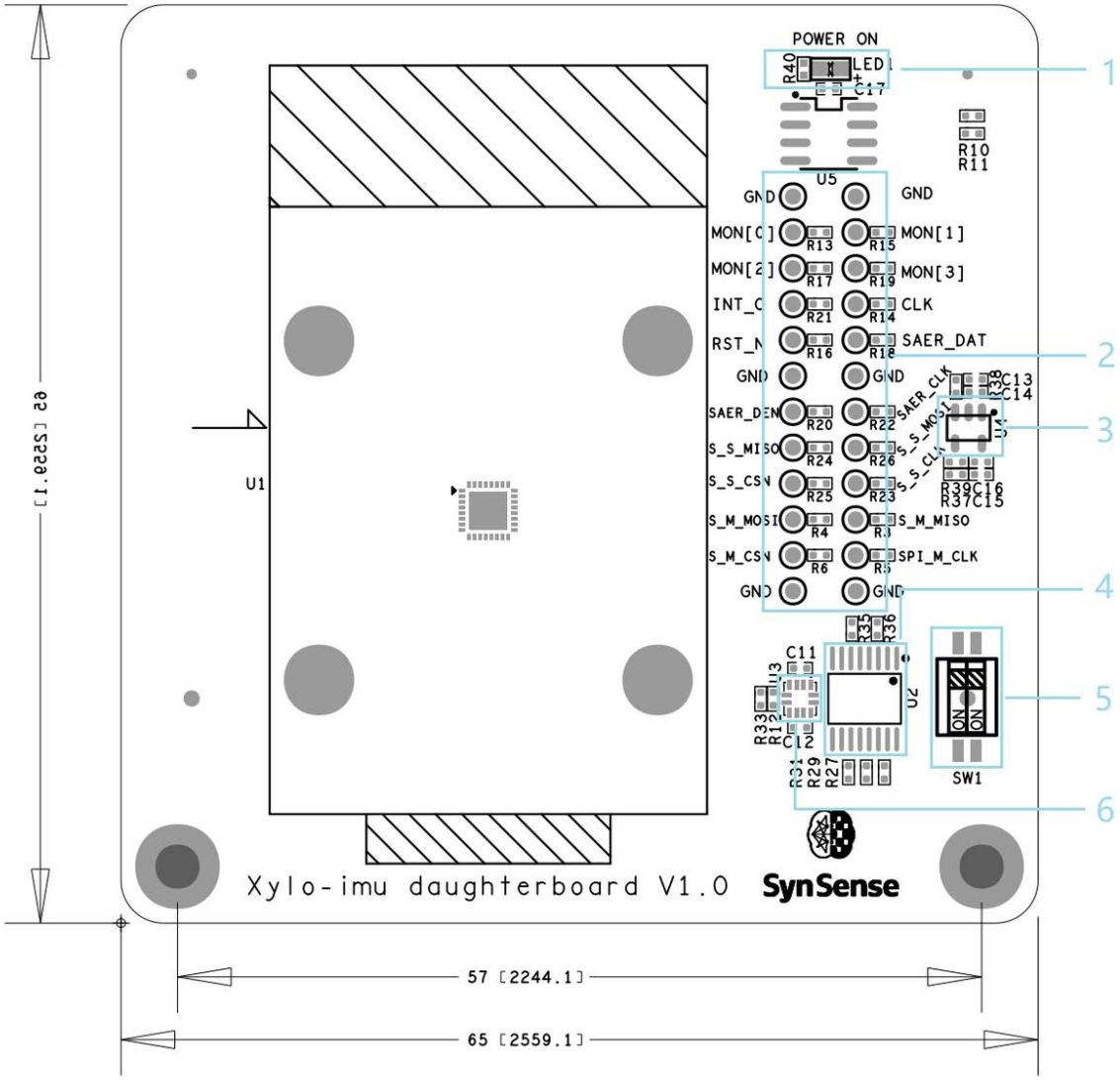


Figure 7 . Front view of the Xylo™IMU daughter board (mm(mil))

1. Power ON LED
2. Test Holes for chip
3. IO Power 3.3V(Optional)
4. Chip IF IO switch
5. IO power selector
6. 3-axis Accelerometer

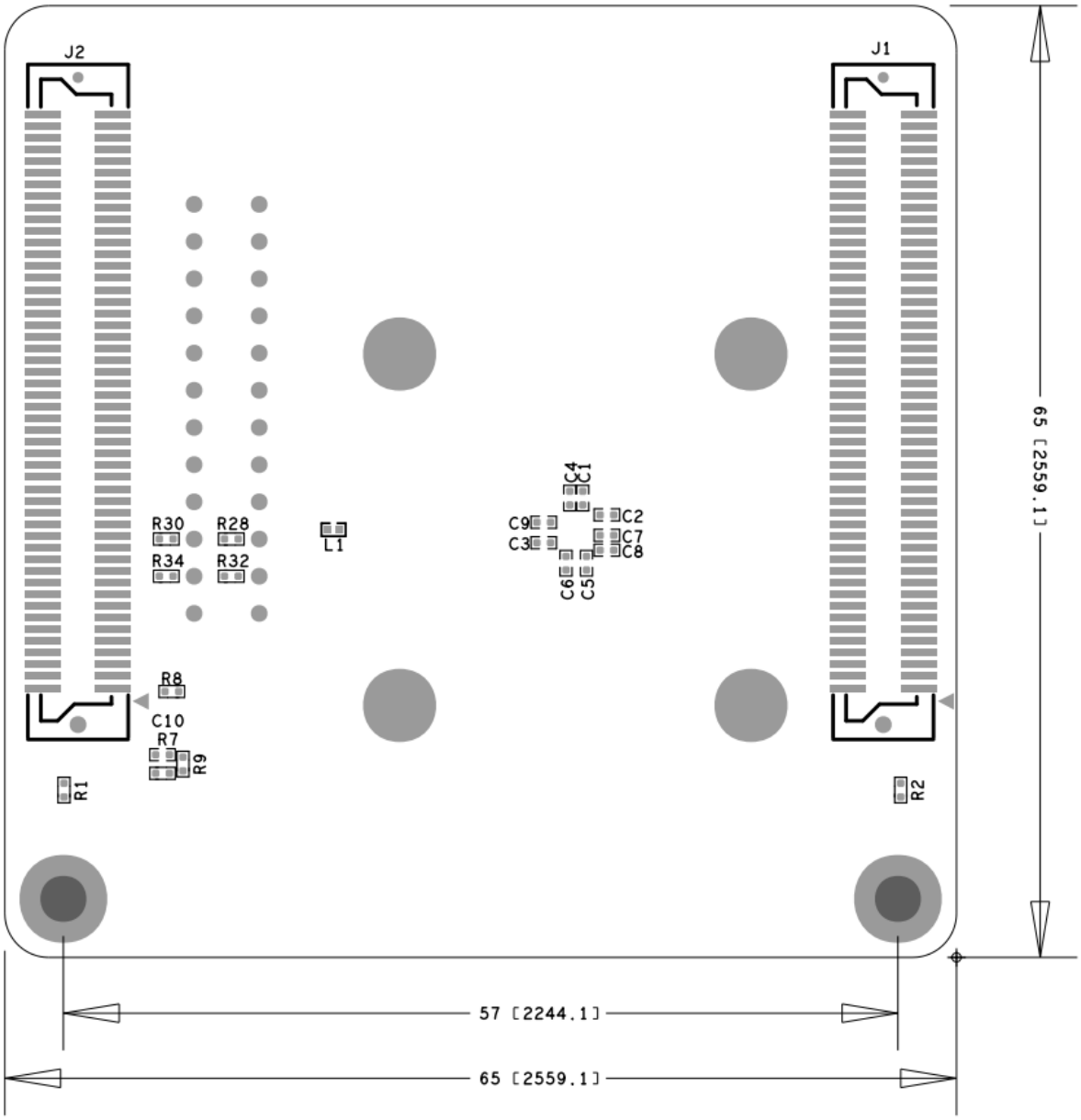


Figure 8 . Bottom view of the Xylo™IMU daughter board (mm(mil))

5. Getting started

5.1. Install Rockpool

```
pip install rockpool[xylo]
```

5.2. Rockpool documentation for Xylo™IMU

https://Rockpool.ai

5.3. USB configuration for Linux

Please refer to Samna online documentation: https://synsense-sys-int.gitlab.io/samna/install.html#udev-rules-on-linux-systems

5.4. Connecting to the Xylo™IMU from Python using Rockpool

```
from rockpool.devices.xylo import find_xylo_hdks
import numpy as np

# - Enumerate the connected Xylo devices
hdks, modules, device_vers = find_xylo_hdks()
```

```
The connected Xylo HDK contains a Xylo IMU. Importing
`rockpool.devices.xylo.imu`
```

```
xhdk = hdks[0]
xylo = modules[0]
```

```
# - Access the Xylo SNN core on the HDK
mod = xylo.XyloSamna(hdk, config)
output, _, _ = mod(input_data)
```

```
# - Record data from the on-board IMU
mod = xylo.IMUData(hdk)
data, _, _ = mod(np.zeros((0, 200, 0)))
```

```
# - Deploy an application to Xylo IMU with live sensor input
mod = xylo.XyloIMUMonitor(hdk, config)
output, _, _ = mod(np.zeros((0, 200, 0)))
```

Quick-start documentation: https://rockpool.ai/devices/xylo-imu/xylo-imu-intro.html

6. Change log

No.	Ver	Date	Editor	Changes
1	V0.1	2023.09	AL	Initial version
2	V1.0	2023.09	Dylan Muir	Text edits in Product Definition and Features

Make Intelligence Smarter