Xylo-Audio Development Kit Datasheet

Jul 2022

Contents

1. Introduction

1.1 Xylo-Audio

Xylo is a family of digital SNN inference ASICs, operating in real time at low power (<10mW). These devices are designed for low-power continuous temporal signal processing and pattern detection tasks, such as keyword spotting, ambient scene detection, industrial and livestock monitoring, bio-signal monitoring, etc.
Xylo-Audio (SYNS61201) contains an analog audio front-end, designed to flexibly convert single-channel audio signals into asynchronous events for processing by the Xylo SNN core.

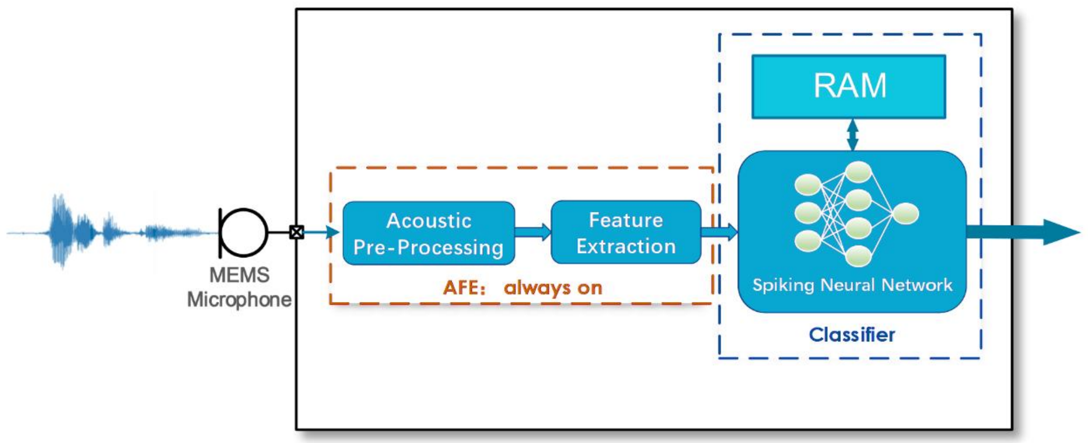

Figure 1: Xylo-Audio block diagram

1.2 Xylo-Audio development kit

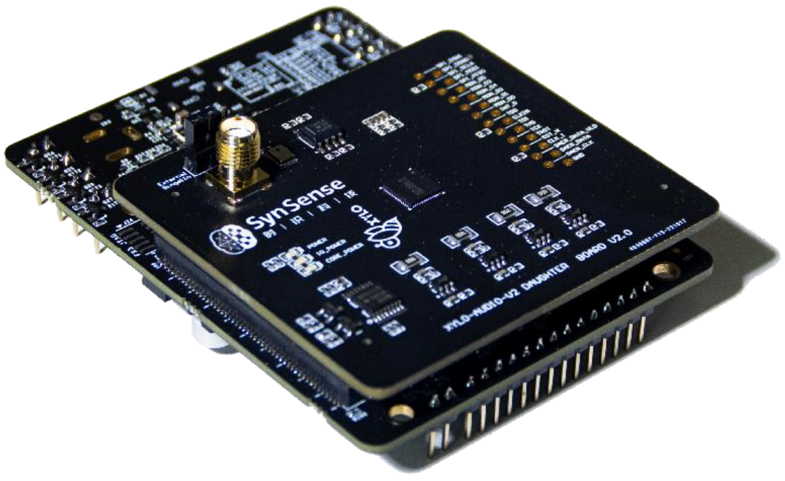

Figure 2: Xylo-Audio development kit

Xylo-Audio development kit is powered by SynSense Xylo-Audio chip, which brings the flexibility of convolutional audio processing to microwatt energy budgets. It provides the capabilities for real-time audio detection. A high performance MEMS mic, power consumption measuring unit, external audio input port and a USB-C port are also provided by Xylo-Audio development kit. Xylo-Audio development kit is not only limited to keywords detection, but also capable of detecting almost any audio feature. Development of network is made easy with our open-source Python library Rock-pool and SynSense device toolchain Samna.

2. Features

2.1 Xylo-Audio

2.1.1 Key features

- AFE integrated, works directly with kinds of MEMS microphones. Supports both single end and differential mode
- Event-based analog feature extraction with neural technology for power saving
- With programmable AFE gain, the amount of information extracted from the incoming signal can scale in complexity, which supports large range of input sound intensity
- 4-wire SPI slave interface for register and RAM configuration, support both single and burst mode access
- Support up to 1000 reservoir neurons and up to 8 classification
- 1-bit configurable interrupt indicating classification / detection done
- 4-bit monitor pins for debug purpose, internal debug registers available
- Up to 100 MHz internal operating frequency
- Extremely low memory footprint (~150 KB), memory power control granularity down to 2 KB
- Ultra-low average working power consumption (~550 uW)

2.1.2 Electrical characteristics

Feature	Specification	Units
VDD Core Supply voltage	1.1	V
IOVDD Supply Voltage	2.5	V
Digital Input/Output High Level	2.5	V
Digital Input/Output Low Level	0	V
MIC Input Common Mode Voltage	0 - 1.1	V
Input Signal Bandwidth	100 - 20K	Hz
Max Input Amplitude (RMS)	100	mV
Max Clock Frequency (AFE)	12.5	Mhz
Max Clock Frequency (Xylo-core)	100	Mhz
Startup Time	0.5	sec
Typical average Power	<350	uW

Table 1: Electrical characteristics

2.1.3 The Xylo SNN core

The Xylo SNN core implements a real-time or accelerated-time simulation of a population of LIF spiking neurons, for use in inference tasks. It is a reconfigurable device, able to implement networks with up to 1000 hidden neurons and 8 output neurons (Figure 3). Each neuron is a digital binary LIF neuron with up to two 16-bit synaptic states and a 16-bit membrane state, with integer logic (Figure 4).

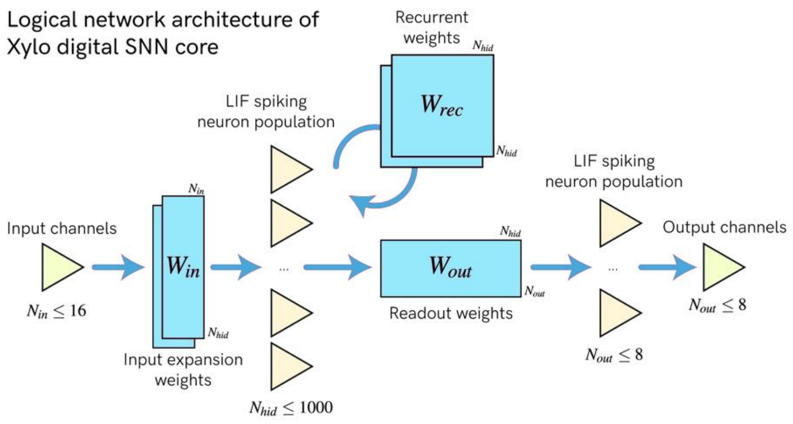

Figure 3: The logical architecture of the Xylo SNN core

Xylo supports up to 16 asynchronous event input channels and up to 8 binary event output channels. Xylo provides configuration for up to 1000 recurrent LIF spiking neurons, to flexibly implement feed-forward and recurrent network architectures.

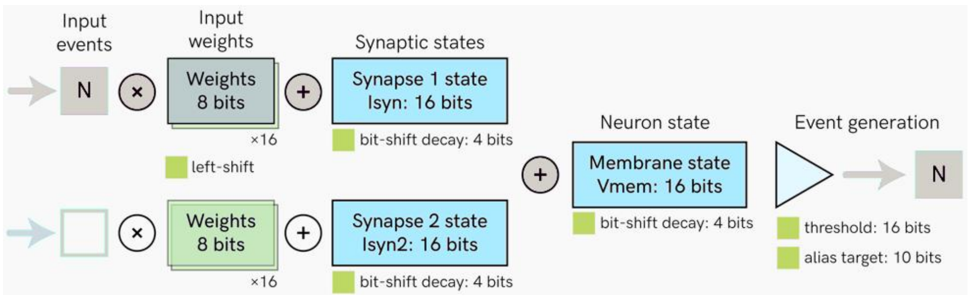

Figure 4: The digital binary LIF spiking neuron model implemented by Xylo

(Up to two synaptic states and one membrane state are supported per neuron)

Per-neuron synaptic and membrane time constants are configurable, as well as individual spiking thresholds per neuron.

2.1.4 The Xylo audio front-end module

The audio front-end module (AFE) is an analog core that converts single-channel analog signals into 16 event-coded channels. The logical design of the component is shown in Figure 5.

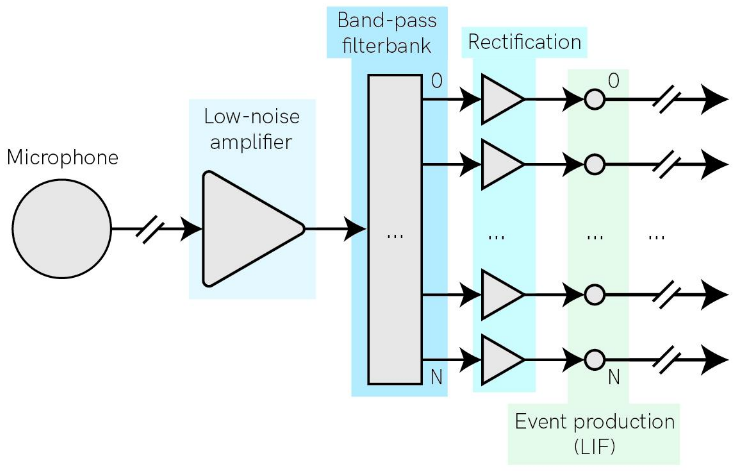

Figure 5: The logical architecture of the Xylo analog audio front-end

The module operates by splitting a single-channel audio signal into 16 frequency bands, using a configurable band-pass filter bank. The module then analyses the instantaneous power in each frequency band, encoding the instantaneous power as a series of asynchronous events. The module therefore outputs 16 channels of asynchronous events, which are passed to the SNN core on Xylo for processing. A configurable band-pass filter bank, followed by rectification and event production, converts single-channel analog signals in 16 asynchronous event channels. Each output channel encodes the instantaneous power within each frequency band of the filter bank.

2.2 Development kit

- MEMS MIC on board, supports both single end and differential mode
- External audio SMA input port, supports both single end and differential mode
- Power consumption measuring unit on board
- 1x USB 3.0 TYPE-C port
- Single end to difference module, for external audio signal only
- Switchable power supply, user can choose on-chip LDO

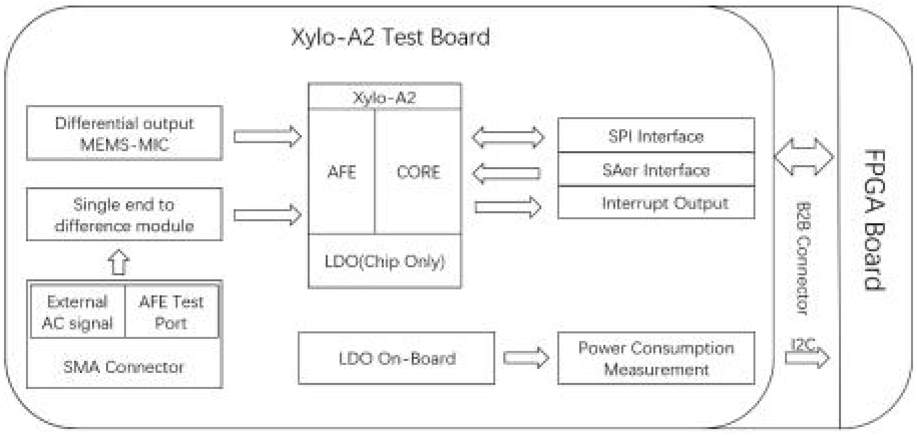

Figure 6: High-level diagram of main components of the development kit

3. Mechanical specification

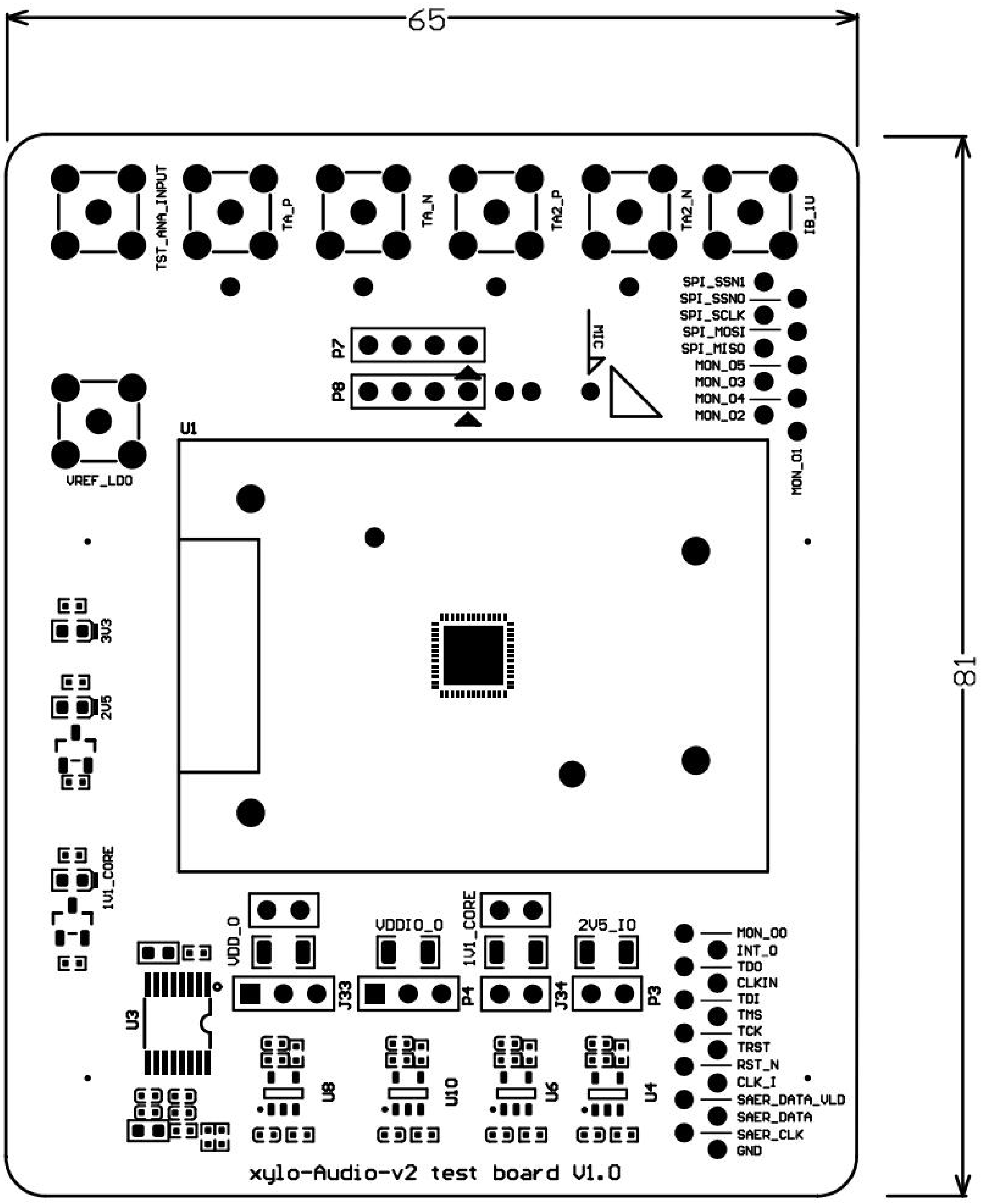

Figure 7: Top mechanical dimensions (mm)

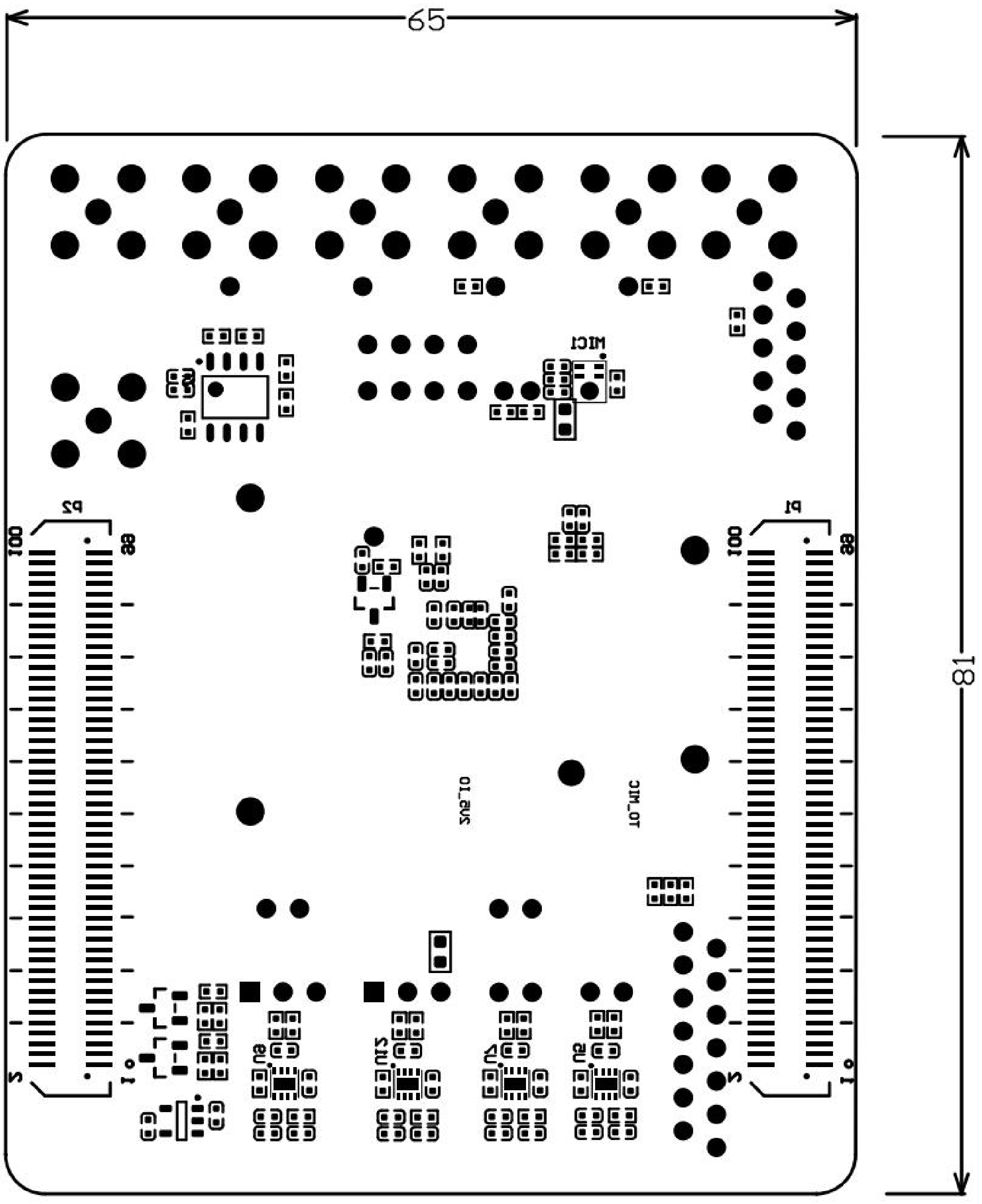

Figure 8: Bottom mechanical dimensions (mm)

4. Getting started

SynSense provides Rockpooland Samnato help develop on the Xylo-Audio Development Kit.

Rockpool is an open source Python package for developing signal processing applications with spiking neural networks. Rockpool allows you to build networks, simulate, train and test them, deploy them either in simulation or on event-driven neuromorphic compute hardware. Rockpool provides layers with a number of simulation backends, including Brian2, NEST, Torch, JAX, Numba and raw numpy. Rockpool is designed to make machine learning based on SNNs easier. It is not designed for detailed simulation of biological networks.

Samna is the developer interface to the SynSense toolchain and run-time envi- ronment for interacting with all SynSense devices. Develped towards efficiency and user friendly, a set of Python API is availabe with the core running in C++, it is possible to work with neuromorphic devices in a professional and elegant manner. Samna also features an event based stream filter system allows real-time, multi-branch processing of the event based stream coming in or out from the device. With an integration of a just-in-time compiler in Samna, the fexlibility of this filter system has been taken to an even higher dimension, which supports adding users defined filter functions at run-time to meet requirements of any different scenarios.

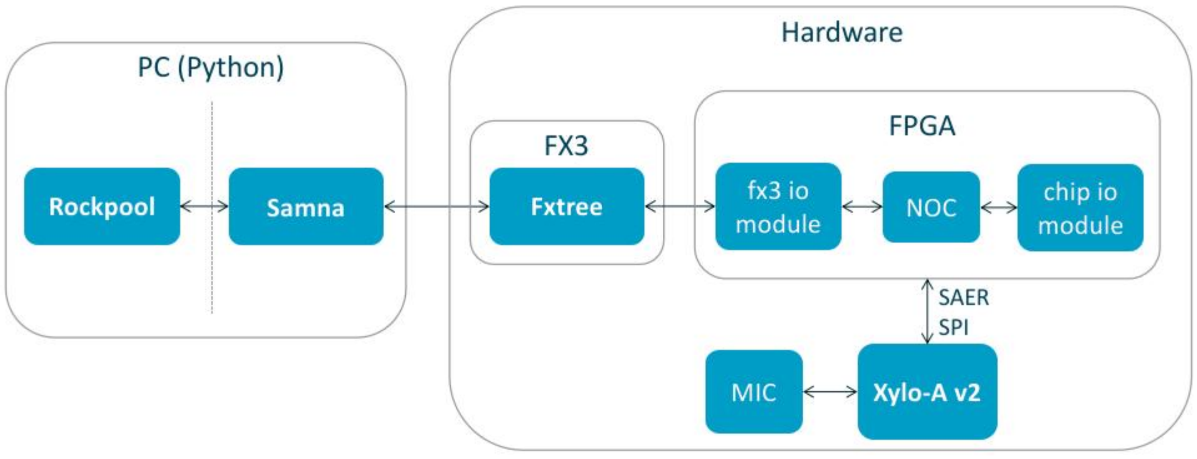

Figure 9: System diagram for application

5. Changelog

5.1 1.0 - 2022.06.30

- Initial publish