FlyDVS: An Event-Driven Wireless Ultra-Low Power Visual Sensor Node

Alfio Di Mauro*, Moritz Scherer*, Jordi Fornt Mas*, Basile Bougenot*, Michele Magno*, Luca Benini*
*Integrated Systems Laboratory, ETH Zurich, Switzerland
Emails: {adimauro,scheremo,jormas,basileb,michele.magno,lbenini}@ethz.ch

***Abstract*—Event-based cameras, also called dynamic vision sensors (DVS), inspired by the human vision system, are gaining popularity due to their potential energy-saving since they generate asynchronous events only from the pixels changes in the field of view. Unfortunately, in most current uses, data acquisition, processing, and streaming of data from event-based cameras are performed by power-hungry hardware, mainly high-power FPGAs. For this reason, the overall power consumption of an event-based system that includes digital capture and streaming of events, is in the order of hundreds of milliwatts or even watts, reducing significantly usability in real-life low-power applications such as wearable devices. This work presents FlyDVS, the first event-driven wireless ultra-low-power visual sensor node that includes a low-power Lattice FPGA and, a Bluetooth wireless system-on-chip, and hosts a commercial ultra-low-power DVS camera module. Experimental results show that the low-power FPGA can reach up to 874 efps (event-frames per second) with only 17.6mW of power, and the sensor node consumes an overall power of 35.5 mW (including wireless streaming) at 200 efps. We demonstrate FlyDVS in a real-life scenario, namely, to acquire event frames of a gesture recognition data set.**

***Index Terms*—Brain-Inspired Sensor, Event-based camera, Bluetooth Low Energy, Low power Design, FPGA, ULP, edge device.**

I. Introduction

Event cameras such as Dynamic Vision Sensor (DVS), also known as neuromorphic cameras, are biologically-inspired vision sensors that capture light intensity changes [1]. Brightness changes are sensed asynchronously and independently for every pixel and are manifested as output spikes, or events, which can have positive or negative polarity (transition from high intensity to low intensity and vice versa).

Event-based cameras present numerous advantages compared to traditional cameras [2]. Most importantly, the event-based representation of images eliminates redundancy, and makes the transmission of video frames much more effective; a conventional camera outputs every pixel on the frame, resulting in massive data transmissions, as well as high bandwidths. DVS cameras are data-driven: they only transmit pixels whose intensity change has exceeded a certain threshold. Due to their frame-less nature, event cameras have much faster response times and lower latency (both in the order of µs), very high dynamic range (140 dB vs 40 dB of standard cameras), and lower power consumption [2]. These unique features make DVS cameras particularly interesting for a variety of applications, such as wearable image acquisition, hand-gesture recognition, robotics, healthcare-oriented, and human-computer interaction (HCI) where the latency and energy efficiency are dominant constraints [1], [3], [4]. The most stringent constraint characterizing these application scenarios is the limited energy budget available to acquire the data from the vision system. Indeed, modern wearable or mobile smart sensor nodes are typically supplied by small batteries. Additionally, when we scale such ideas to the extreme low-energy budget application context, [5], effective use of the energy drained by the battery becomes the priority.

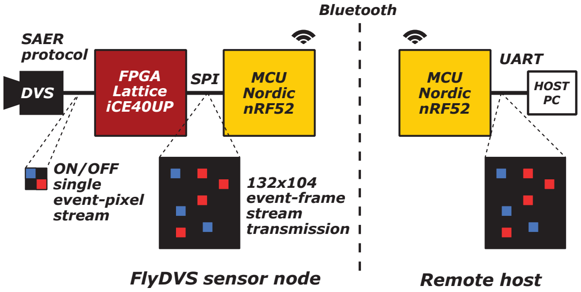


Fig. 1: FlyDVS acquisition system block diagram. The diagram is split into a remote (sensor node) part, which includes the camera, the FPGA and a microcontroller to transmit the events, and a central (host) part, which receives the events sent by the remote part

Over the last ten years, there was a drastic increase in the number of smart sensors deployed in our daily environment. Key phenomena enabling such a pervasive sensor-node diffusion have been the technological scaling and the consequent power consumption reduction of sensing devices, as well as the extremely low energy per operation consumption achieved by edge computing devices [6]–[8], which allowed to bring more cognitive capabilities onto the sensor nodes. Such sensor nodes are often connected to a remote host through a low-energy wireless channel.

As edge-devices become more computationally capable [9], eliminating the bottlenecks that prevent the deployment of event-based sensors in-the-field becomes a key enabler for the development of smarter sensor nodes. In this direction, a promising approach is to exploit the energy-to-information proportionality of those event sensors. Indeed, as opposed to constant-rate sensors (e.g., frame-based cameras), event-based sensors typically produce a variable amount of data, which is related to the input activity of the sensor [10]. Edge processing and event cameras are also enabling intelligent and low latency devices for biomedical applications.

On the other hand, today's event cameras are characterized by the low power consumption of the sensor, but a high power consumption due to the interface, as the sensor, often expose non-standard communication protocol. Therefore, data acquisition is mainly done with a high-power Field Programmable Gate Array (FPGA) and a USB interface [11]. This limits significantly the use of these promising technologies on real application scenarios of wearable and low-power devices for

978-3-9819263-5-4/DATE21/©2021 EDAA

mobile applications and healthcare. [12].

This paper presents an end-to-end data acquisition node for event-based sensors that exploits a low-power FPGA to interface with the sensor and low-energy Bluetooth for transmitting the event-frames (eframes) wirelessly. The main contribution of this paper can be summarized as follows:

- The design and implementation of a data acquisition system for event-based vision sensors capable of acquiring 874 event-frames per second (efps) with a $17.6\,\mathrm{mW}$ power budget, which is around 40x lower power than the current commercial USB version of the event-based camera [13]
- The transmission over a Bluetooth channel of $200\,\mathrm{efps}$ in a $35.5\,\mathrm{mW}$ power budget
- An an end-to-end low-power sample application for our sensor node where the DVS camera is used to collect an event-based gesture data set for a gesture recognition task.

II. System Architecture

Fig.1 shows a block diagram of the FlyDVS system architecture, which can be divided in three main blocks: *i)* the event-based sensor, namely a DVS camera, *ii)* the FPGA subsystem for interfacing with the camera, and *iii)* the Bluetooth low-energy microcontroller subsystem for data transfer. In Fig.1, we can observe an example of how the system transmits information to a host system.

A. The DVS Camera

The event camera selected for the proposed design is the DVS132S sensor described in [14]. This sensor has been selected for its trade-off between resolution, power consumption, and the event-rate. In particular, the event camera sensor features a maximum resolution of 132x104 pixels, each with a size of 10µm x 10µm. The maximum output event-rate is $180\,\mathrm{Meps}$, the minimum reported power consumption is $250\,\mu\mathrm{W}$, measured at $100\,\mathrm{keps}$. When the luminosity of a pixel has increased/decreased by a certain threshold, an ON/OFF event is generated. When neither of these conditions is met, no event is triggered, therefore no energy is spent on the communication interface. Data transmissions from the sensor consist of the events generated for each pixel. As a result, each frame request triggers the transmission of a maximum of 13,728 (132x104) events and a minimum of no events. On the power consumption side, although the event camera consumes only a few mW [14], the interface can reach a few hundred of mW [13].

B. The FPGA

To minimize the power consumption of the interface, we selected a low-power FPGA, namely, a Lattice Semiconductor iCE40UP5K[1]. This choice is a trade-off among the required FPGA resources, power consumption, and the maximum operating frequency of the device. The FPGA is supplied at its nominal voltages, i.e., $1.2\,\mathrm{V}$ for the fabric (core), and $3.3\,\mathrm{V}$ for the FPGA IO banks. In the following, we describe the architecture of the circuit implemented on the FPGA. Fig.2 reports a block diagram of the DVS interface circuit. This subsystem can be divided into three subsystems.

1) DVS driver: that is the module that physically interacts with the DVS camera. To stream the event-pixels, the DVS uses a digital synchronous protocol, namely the synchronous address-event representation Synchronous Address-Event Representation (SAER). With 8 bit-parallel data lines transmitting first the address of the `row` that has at least one active pixel, and then all the active `column` addresses of each event-pixel in sequence.

2) The event-frame buffer: this module stores the active pixel addresses acquired by the DVS driver module in a given time interval, which can be configured. A module called *address converter* receives first a row (y) address, therefore, it starts the row update process. In this phase, a memory controller updates all the pixels of the selected row, according to the received column (x) addresses. The row update stops as soon as a special column termination address is received. Then, the row update can start again, and the frame-buffer is ready to accept a new row address.

3) The SPI subsystem: to transfer the data to the microcontroller, we implemented a master Serial Peripheral Interface (SPI) capable to fetch the data from the event-frame buffer. The data transmission is managed an SPI controller module. To optimize the transmission, and to avoid starving the SPI, e.g., if not enough data have been received from the DVS driver, the SPI controller waits until the SPI payload of 80 Event-Words (240 Bytes) is ready at the SPI input First-In First-Out Queue (FIFO). The SPI payload size has been chosen as the highest number of event-pixels that fit into a single SPI transaction, which in our system is limited to 256 Bytes.

C. The microcontroller

As peripheral and central Microcontroller (MCU) units, we chose two nRF52 family system on chip (SoC) from Nordic Semiconductor. The SoC is built around a 64 MHz ARM Cortex-M4F microcontroller and hosts a 2.4 GHz Bluetooth transceiver, which is specifically targeted for low-power Bluetooth applications.

The peripheral MCU (connected to the FPGA) wakes up as soon as an interrupt generated by the SPI module is received, indicating that an SPI transaction has been completed, and new data are available. Then, it forwards the received SPI packet to the Bluetooth Low-Energy (BLE) driver, which initiates a Bluetooth transmission of the data. The central MCU then receives the incoming Bluetooth packet and forwards it to its Universal Asynchronous Receiver-Transmitter (UART) port, connected to the host PC, which eventually collects and displays the event-frames.

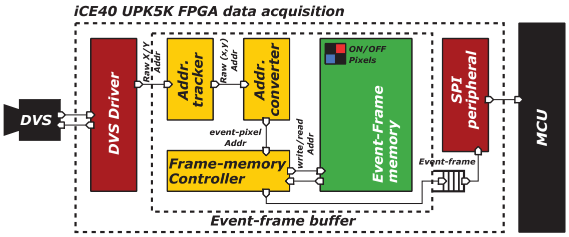


Fig. 2: FPGA DVS event-frame acquisition system architecture

[1] http://www.latticesemi.com/view_document?document_id=51968

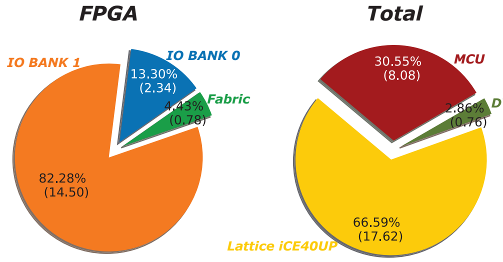


Fig. 3: System power consumption breakdown. On the left, the power consumption of the Lattice iCE40UP. On the right, the FlyDVS sensor node total power consumption; including wireless transmission at 200 efps.

III. Experimental Results

The whole system has been designed and implemented to evaluate the power consumption for each building block of our system and the functionality of the proposed solution.

Table I reports the FPGA resources utilization. From the table it is evident that our design occupies only 21% of the FPGA Lookup Tables (LUTs) and 73% of the available memory, leaving a significant amount of free computational resources to implement additional data pre-processing algorithms.

From the static timing analysis (Static Timing Analysis (STA)), The critical path of the digital circuit implemented on the FPGA is 36 ns. Therefore, the FPGA clock could run at a maximum frequency of 27 MHz, reaching a theoretical maximum event-frame of 2360 efps. However, as the FPGA does not represent a bottleneck for the event-pixel acquisition, we were able to lower the FPGA operating frequency to 6 MHz. We measured the power consumption of the FPGA at 6 MHz, reporting 17.62 mW in presence of high activity of the sensor, and when the SPI is transmitting 200 efps. The power consumption of the FPGA decreases to 14.5 mW when no SPI data transmission is happening, i.e., we are not transmitting event-frames over the SPI, but we are collecting event-pixels from the camera. Note that this power is also including level-shifters to convert the signals from 3.3 V to 1.2 V, to be compatible with the DVS camera IO bank voltage. The total power consumption of the system is reported in Fig.3.

The maximum end-to-end worst-case event-frame rate guaranteed by the system is 5.2 efps (including the streaming over Bluetooth), i.e. when the input bandwidth of the FPGA is saturated by the DVS camera. Figure 4 reports an illustration of the worst-case communication bandwidth between each sub-module of the system. Note that each block, i.e., each group of 4 pixels, can transmit up to 8 events, however, this represents a worst case, as all the event-pixel would be active at the same time. From the figure, it is evident that the main theoretical bandwidth limitation is introduced by the Bluetooth communication channel.

The FPGA can sustain a worst-case 874 efps from the camera, which is more than one order of magnitude higher than what conventional cameras can provide in a comparable power budget, enabling on-FPGA event processing when high responsiveness to the scene change is needed by the application. If we consider the bandwidth limitation introduced by the SPI, the worst-case efps decreases to 72.8 efps, which is comparable with what is provided by commercial frame-based cameras. The bottleneck of our system is the Bluetooth communication channel, which limits the end-to-end event-frame transmission rate. In the worst-case scenario, the system can still sustain 5.2 efps end-to-end transmission.

Element	Total	Utilized	Util. %
LUT	5280	1154	21
Flip-Flop	5280	441	8
DSP	8	0	0
IOs	39	38	97
EBR RAM	30	22	73
SPRAM	4	0	0

TABLE I: FPGA Resource utilization.

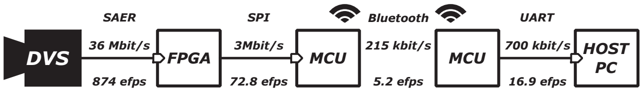


Fig. 4: Minimum sustained communication bandwidth among system modules in worst-case operating scenario.

To evaluate the designed and implemented system in terms of eframe acquisition, a hand-gesture recognition data set, useful for many healthcare applications, has been acquired, and it is further illustrated in Section IV. We recorded the data set by setting 5 ms as the time of an event-frame, meaning that the events are collected from the camera in a 5 ms time, and then transmitted through the full communication pipeline to the host PC. Fig. 5 reports the distribution of the events in the reference period during the acquisition of the data set. We can observe that the average number of events per event-frame never exceeds 300, and it is distributed around 50. In this operating mode, our system could sustain and end-to-end 240 efps rate without event-pixel loss. However, by accepting a negligible amount of event-pixel being lost, i.e. less than 5%, the system can sustain an event-frame rate of approximately 720 efps. From this experiment we also validated that in a real use case, the required bandwidth is determined by the sensor activity. Specifically, in the context of a gesture recognition task, our system could achieve an event-frame rate significantly higher than the theoretical estimated worst-case frame rate

IV. Dataset

The custom dataset acquired from the DVS acquisition system is based on the DVS128 gesture dataset from IBM Research [1]. This dataset captured human hand gestures using a DVS128 camera, with a total of 11 gesture classes. The dataset recorded 29 subjects under different illuminations for several seconds, with a sampling time of 1 ms (1000 efps). There are a total of 122 recordings per class. The DVS128 dataset is licensed under Creative Commons.

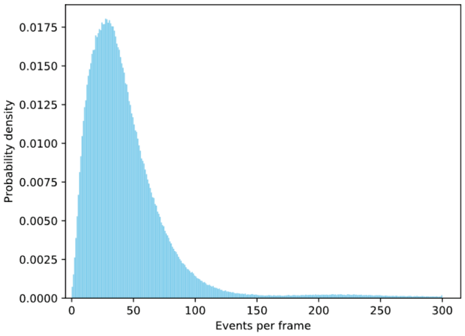
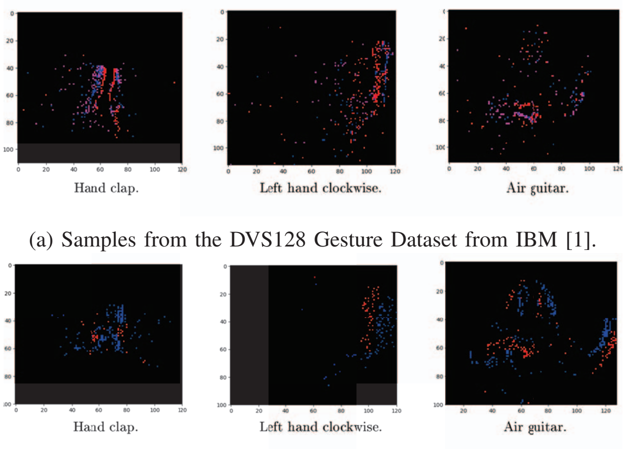


Fig. 5: Probability density function describing the likelihood of the event number per event-frame in the hand-gesture data set.



(a) Samples from the DVS128 Gesture Dataset from IBM [1].

(b) Samples from the dataset acquired with our system using Bluetooth low energy wireless communication.

Fig. 6: Data set comparison: example snapshots for three equivalent gestures.

To have a future comparison in terms of quality, the data set acquired in this work has the same 11 classes, that have been acquired with the low-power DVS132S camera and the wireless acquisition system developed. For a proof-of-concept demonstration, the new recordings captured 5 subjects under the same illumination conditions. The subjects performed 14 repetitions for each of the 11 gestures. Hence, 70 samples per class have been generated, with a total of 770 recordings.

Figures 6a and 6b show recordings at one particular time-step for both datasets. It can be noticed that the acquired recordings look less sharp than the counterparts of IBM. This is mainly due to the following factors:

- The lens used with the DVS132S camera is a prototyped 3D printed piece, which lacks the precision of an industrial manufacturing process (unlike the DVS128 camera used by IBM).
- The acquisition rate is lower (200 efps for the DVS vs 1 kefps of IBM).
- The cameras used for the data set acquisition is different, specifically, the DVS132S targets low-power operation, which might come at a cost in terms of frame sharpness.

On the other hand, this sharpness drop may not be detrimental for an artificial neural network (ANN) model accuracy which processes the data. In particular, for Spiking Neural Network (SNN) the first layer of the deep Spiking Neural Networks (SNN) architecture usually applies a pooling layer to downsample the image, thus losing all high-frequency details [15]. This aspect will be further investigated in future works.

V. Conclusion

This work presented the design and the implementation of FlyDVS, the first event camera-based sensor node designed for wireless and low-power data acquisition and processing. The proposed architecture exploits a 132x104 pixel low power event camera, and it includes a low power Lattice FPGA and a Bluetooth system on Chip. FlyDVS is ready to acquire a data set for wearable and healthcare application scenarios in a wireless and low power fashion. Experimental results acquiring a real hand-gesture data set have shown the capability of the proposed solution to acquire up to $874\,\mathrm{efps}$ (event-frames per second) from the DVS camera consuming only $17.62\,\mathrm{mW}$ of power. The whole system power consumption is $35.5\,\mathrm{mW}$, including the wireless event-frame streaming at $200\,\mathrm{efps}$. Future works will focus on more data set acquisition and the capability to run spiking neural networks or sparse convolution on the transmitted data or close to the sensor on the available low power resources. As the FPGA power is dominant, future work will investigate the ASIC implementation of the interface to reduce both internal power and IO power of the interface.

VI. Acknowledgements

The authors would like to thank *Armasuisse Science and Technology* for funding this research, and *IniVation* for kindly lending us a DVS camera.

References

[1] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza *et al.*, "A low power, fully event-based gesture recognition system," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7243–7252.

[2] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis *et al.*, "Event-based vision: A survey," *arXiv preprint arXiv:1904.08405*, 2019.

[3] W. Song, Q. Han, Z. Lin, N. Yan, D. Luo, Y. Liao, M. Zhang, Z. Wang, X. Xie, A. Wang *et al.*, "Design of a flexible wearable smart semg recorder integrated gradient boosting decision tree based hand gesture recognition," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 6, pp. 1563–1574, 2019.

[4] M. Vandersteegen, W. Reusen, K. Van Beeck, and T. Goedemé, "Low-latency hand gesture recognition with a low-resolution thermal imager," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 98–99.

[5] V. K. Sarker, M. Jiang, T. N. Gia, A. Anzanpour, A. M. Rahmani, and P. Liljeberg, "Portable multipurpose bio-signal acquisition and wireless streaming device for wearables," in *2017 IEEE Sensors Applications Symposium (SAS)*, 2017, pp. 1–6.

[6] M. Eggimann, S. Mach, M. Magno, and L. Benini, "A risc-v based open hardware platform for always-on wearable smart sensing," in *2019 IEEE 8th International Workshop on Advances in Sensors and Interfaces (IWASI)*. IEEE, 2019, pp. 169–174.

[7] M. Gautschi, P. D. Schiavone, A. Traber, I. Loi, A. Pullini, D. Rossi, E. Flamand, F. K. Gurkaynak, and L. Benini, "Near-Threshold RISC-V Core With DSP Extensions for Scalable IoT Endpoint Devices," *IEEE TVLSI*, vol. 25, no. 10, pp. 2700–2713, 2017.

[8] A. D. Mauro, F. Conti, P. D. Schiavone, D. Rossi, and L. Benini, "Always-on 674uw@4gop/s error resilient binary neural networks with aggressive sram voltage scaling on a 22-nm iot end-node," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–14, 2020.

[9] A. Pullini, D. Rossi, I. Loi, A. Di Mauro, and L. Benini, "Mr. wolf: A 1 gflop/s energy-proportional parallel ultra low power soc for iot edge processing," in *ESSCIRC 2018 - IEEE 44th European Solid State Circuits Conference (ESSCIRC)*, 2018, pp. 274–277.

[10] A. Di Mauro, F. Conti, and L. Benini, "An ultra-low power address-event sensor interface for energy-proportional time-to-information extraction," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2017, pp. 1–6.

[11] G. García, C. Jara, J. Pomares, A. Alabdo, L. Poggi, and F. Torres, "A Survey on FPGA-Based Sensor Systems: Towards Intelligent and Reconfigurable Low-Power Sensors for Computer Vision, Control and Signal Processing," *Sensors*, vol. 14, no. 4, p. 6247–6278, Mar 2014. [Online]. Available: http://dx.doi.org/10.3390/s140406247

[12] A. Banerjee, C. Chakraborty, A. Kumar, and D. Biswas, "Emerging trends in iot and big data analytics for biomedical and health care technologies," in *Handbook of data science approaches for biomedical engineering*. Elsevier, 2020, pp. 121–152.

[13] IniVation, *IniVation Specifications – Current models*, 2020 (accessed Sept, 2020). [Online]. Available: https://inivation.com/wp-content/uploads/2020/09/2020-09-16-DVS-Specifications.pdf

[14] C. Li, L. Longinotti, F. Corradi, and T. Delbruck, "A 132 by 104 10um-pixel 250uw 1kefps dynamic vision sensor with pixel-parallel noise and spatial redundancy suppression," in *2019 Symposium on VLSI Circuits*, 2019, pp. C216–C217.

[15] L. Cheng, Y. Liu, Z.-G. Hou, M. Tan, D. Du, and M. Fei, "A rapid spiking neural network approach with an application on hand gesture recognition," *IEEE Transactions on Cognitive and Developmental Systems*, 2019.