

¹ Open-source active cooling system for GPU and ² FPGA accelerator cards

Rodolfo Manin* and Alberto Saa†

*Instituto de Matemática, Estatística e Computação Científica,
Universidade de Campinas, Campinas, SP, Brazil*

August 8, 2022

Abstract

Typically, high-performance Graphics Processing Units (GPU) and Field Programmable Gate Arrays (FPGA) accelerators are high-power demanding elements, designed to be mounted in specialized racks with intense forced airflow systems to assure their functioning within proper thermal limits. We present here an open-source, Arduino-based, programmable active cooling system for GPU and FPGA cards that allow their installation in common computer cases. Only standard components and simple 3D-printed pieces are employed in our design. We built and have been continuously using this cooling system for a GPU NVIDIA Tesla V100 and an FPGA Xilinx Alveo U280. The setup is inexpensive and has proved to be flexible and extremely reliable. Furthermore, it can be easily adapted for any other card or element requiring similar cooling.

Metadata Overview

Main design files:

<https://github.com/RodLophus/ArduinoFanController>

Target group: scientists and engineers demanding high performance computing, datacenter managers.

Skills required: Arduino programming – easy, 3D printing – easy, electronics and mechanical assembly – easy.

Replication: The authors built and have continuously used the setup for a GPU NVIDIA Tesla V100 and an FPGA Xilinx Alveo U280.

See section “Build Details” for more detail.

Keywords

High Performance Computing (HPC), GPU, FPGA, active/pассивное охлаждение.

*ORCID: 0000-0002-6154-340X, email: rodolfo@ime.unicamp.br

[†]ORCID: 0000-0003-1520-4076, email: asaa@ime.unicamp.br

²⁰ **(1) Overview**

²¹ **Introduction**

²² High-performance Graphics Processing Units (GPU) and Field Programmable
²³ Gate Arrays (FPGA) accelerators are widely employed nowadays in intensive
²⁴ scientific computing and machine learning applications. The cutting-edge mod-
²⁵ els are typically cooled by a passive heat sink system and designed to be installed
²⁶ in some specific, high-cost, PCIe/NVLink racks [1] with a strong front-to-back
²⁷ forced airflow system to assure their operation within proper thermal limits.
²⁸ Fig. 1 illustrates a common passive cooling airflow for a NVIDIA GPU card.
Under intense use, the power consumption of a typical high-end accelerator card

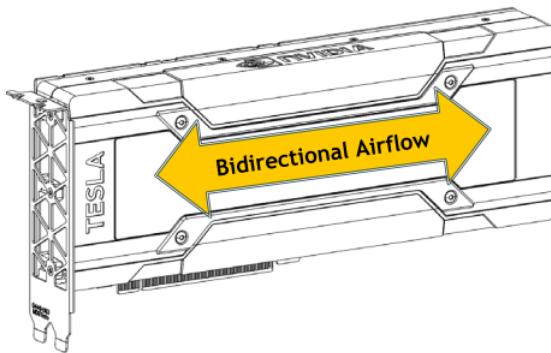


Figure 1: Schematic representation of the passive cooling airflow for a GPU NVIDIA Tesla P100, see [2].

²⁹ can easily surpass 300 W [3], requiring a minimal fully ducted airflow through
³⁰ the card's heat sink of approximately 35 cubic feet per minute (CFM) [2]. Such
³¹ operation features clearly discourage any alternative installation of these high-
³² power demanding accelerator cards in nonspecific racks.

³⁴ However, it is indeed not uncommon to have some of these cards without the
³⁵ proper racks to install them. Having this situation in mind and our own recent
³⁶ necessities, we have designed and built an inexpensive open-source, Arduino-
³⁷ based, programmable active cooling system for a GPU NVIDIA Tesla V100 and
³⁸ for an FPGA Xilinx Alveo U280 mounted in a common PC motherboard in
³⁹ a standard rack-mounted computer case. Only off-the-shelf components and
⁴⁰ simple 3D-printed pieces are employed in our design. For each card, we used
⁴¹ a customized 3D-printed air duct and a dedicated cooler fan with pulse width
⁴² modulation (PWM), activated by our Arduino-based controller circuit, with
⁴³ a temperature sensor placed at the air exit port at the back of the card. The
⁴⁴ controller circuit is configured to adjust linearly the power fan for a temperature
⁴⁵ range $[T_{\min}, T_{\max}]$ that can be set accordingly to the particular card and typical
⁴⁶ operation. However, since the Arduino-based circuit is highly flexible, virtually
⁴⁷ any other controlling strategy and monitoring of the power fan can be easily
⁴⁸ implemented at the Arduino code.

⁴⁹ We have extensively tested our cooling system and it has proved to be very

50 flexible and reliable. Both accelerator cards are now being continuously used
51 in a high performance computing environment. Finally, our setup can be easily
52 adapted for any other card or element with similar heat sink cooling necessities.

53 Overall Implementation and design

54 Fig. 2 depicts our complete setup for a GPU NVIDIA Tesla V100 and for an
55 FPGA Xilinx Alveo U280 mounted in a common PCIe motherboard (ASUS
PRIME Intel H510M-E) in a standard rack-mounted computer case. The 3D-



56 Figure 2: The complete active cooling setup for a GPU NVIDIA Tesla V100
57 (top) and for an FPGA Xilinx Alveo U280 (bottom).

58 printed ducts in high temperature Polylactic Acid (PLA), the fans, and the
59 controller circuit boards are clearly visible. See Fig. 3 for some details of the
60 duct design for the GPU NVIDIA Tesla V100. For further information, build
61 details, and pertinent source files, see the full build instructions at [4]. The
62 airflow is provided by a dedicated $60 \times 60 \times 38$ mm 4-pin PWM cooler fan with
63 a maximum airflow of 53 CFM. To guarantee mechanical stability and avoid
64 vibrations, the fan must be fixed at the computer case, and not at the card. To
65 this end, it is important to employ a computer case with a transverse support
for fans as those ones depicted in Fig. 2.

66 The Arduino-based fan controller circuit is depicted in Fig. 4. All pertinent
67 details can be found in the full build instructions at [4]. The controller circuit

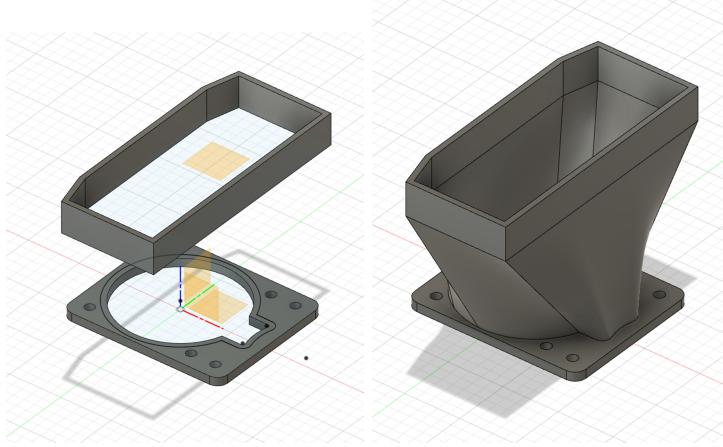


Figure 3: The design of the 3D-printed duct for the GPU NVIDIA Tesla V100. Left: 3D model of the customized end connections for the card and for the fan. Right: A standard 3D modeling software tool is used to generate the smooth structure connecting the end connections, giving origin to the duct. All design details, instructions, and specifications are available at [4].

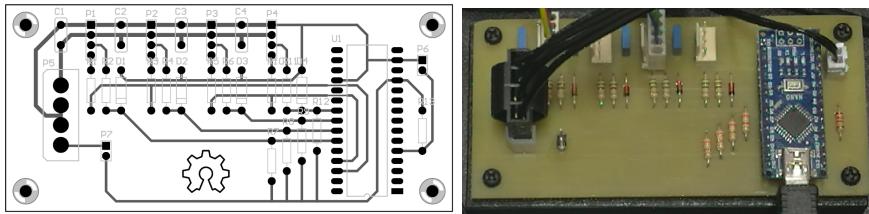


Figure 4: The fan controller based in an Arduino Nano microcontroller. Left: the circuit board. Right: The assembled controller circuit, which can control up to two pairs of 4-pin PWM fans (P_1, P_2 and P_3, P_4 connectors). The NTC thermistor is connected to P_6 . The 12 V ATX high power supply is connected to P_5 . In our case, the Arduino 5 V supply is being provided by the USB connection, but it can also be supplied by the ATX power on P_5 if jumper P_7 is closed. For further details about the electronics, see the full build instructions at [4].

68 can activate up to two pairs of 4-pin PWM fans with independent programs.
 69 We are limited to only two independent programs since the Arduino nano has
 70 only two PWM channels with 16 bits timer, the remaining four 8 bits channels
 71 are inadequate to control the fans. In our case, a second common $120 \times 120 \times$
 72 38 cooler fan, pairwise activated with the main cooler, is placed at the front
 73 part of the computer case to provide enough airflow into the computer case.
 74 The temperature sensor, a Negative Temperature Coefficient (NTC) thermistor,
 75 is placed at the air exit port in the back of the card, see Fig. 5. We have
 76 programmed the Arduino to adjust linearly the power fan for the thermistor
 77 temperature in the range $[T_{\min}, T_{\max}]$. In our case, for continuous use in a
 78 high-performance computing environment, we have set the fan power at 40%



Figure 5: Location of the temperature sensor (NTC thermistor), at the air exit port in the back of the card.

and at 100%, respectively, for temperatures below $T_{\min} = 40^{\circ}\text{C}$ and above $T_{\max} = 55^{\circ}\text{C}$. For the intermediate temperatures, the fan power is linearly increased. Notice that one could use some commercially available alternative fan controller circuit as, for instance, the ZFC39 PWM PC CPU Fan Temperature Controller [5], for implementing simple controlling strategies as the linearly increasing one. However, one of the greatest advantages of our Arduino-based circuit is namely its high flexibility, allowing virtually any other programmable controlling strategy and monitoring for the power fan. Our setups are installed in an air-conditioned datacenter, and the cards can work at full power with internal temperatures comfortably below their working limit of 70°C .

Universality of the design

Our design can be easily applied for any other card or element with similar heat sink cooling necessities. The Arduino can be programmed to implement virtually any controlling strategy and monitoring of the airflow.

(2) Quality control

Safety

There are very few security hazards in the assembling and operation of the setup. The computer power source must remain unplugged during the assembling. The GPU and FPGA cards must never work without the proper cooling system. For instance, our GPU NVIDIA Tesla V100 and FPGA Xilinx Alveo U280, without the proper airflow, reach the shutdown temperature (around 90°C) in some few minutes of operation. Running these cards for long times in temperatures above their proper limits (around 70°C) can permanently damage them.

Calibration

Since we use thermal components, some calibrations are indeed required. The NTC thermistor is incorporated in our circuit by means of a simple voltage divider, see Fig. 6, whose output voltage V_{out} is read by one of the Arduino analog input pins. From V_{out} , the thermistor resistance $R(T)$ is determined. The function $R(T)$ for an NTC thermistor is given by the well-known Steinhardt-Hart

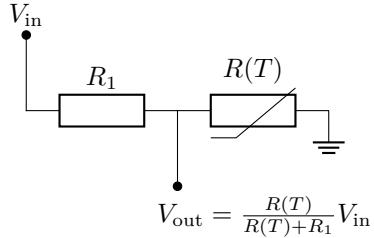


Figure 6: Simple passive voltage divider with an NTC thermistor $R(T)$ and a resistor R_1 . The output voltage V_{out} is read by one of the Arduino analog entries. Since V_{in} and R_1 are known, $R(T)$ can be determined from the value of V_{out} . See [4] for further details.

108 equation [6]

$$109 \quad T^{-1} = a + b \log R + c \log^3 R, \quad (1)$$

110 where the absolute scale is used for the temperature T . Hence, to convert
 111 between T and R , we need the three parameter a , b , and c , which can be
 112 be determined, for instance, by a least square fitting from a set of measurements
 113 (T_i, R_i) , $i = 1, \dots, n$, with $n > 2$. However, for narrow temperature ranges, as in
 114 the present case, the last term in the Steinhart-Hart equation can be neglected,
 115 and we have the more familiar relation

$$116 \quad \beta (T^{-1} - T_*^{-1}) = \log \frac{R}{R_*}, \quad (2)$$

117 where we have set $a = T_*^{-1} - b \log R_*$ and $b = \beta^{-1}$. We use (2) in our Arduino
 118 code [4]. The simplest of-the-shell NTC thermistors are characterized by their
 119 nominal resistance R_* for $T_* = 25^\circ\text{C}$ and the parameter β . In our case, we
 120 employed thermistors with $R_* = 50\text{k}\Omega$ and $\beta = 3950\text{K}$. These parameter,
 121 which are used in the Arduino code, can be easily double-checked with some
 122 simple temperature and resistance measurements.

123 However, the fundamental point of our calibration is not the thermistor charac-
 124 terization, but the management of the difference between the card functioning
 125 temperature, measured internally at its circuit components, and the tempera-
 126 ture measured by the thermistor in the air exit port of the card. It is natural
 127 to expect that the thermistor temperature be smaller than the card internal
 128 temperature, but the difference can depend on many variables, from component
 129 details to environment conditions. The temperature range $[T_{\min}, T_{\max}]$ for the
 130 control of the fan power must be determined on a case-by-case basis. We suggest
 131 to set initially our configuration ($T_{\min} = 40^\circ\text{C}$ with power fan $P = 40\%$ and
 132 $T_{\max} = 55^\circ\text{C}$), and adjust it in order to have the fan running with its power
 133 around 70% with the cards in intense use. It is not advisable to run the fans at
 134 full power in normal conditions since some temperature overshooting can and
 135 typically occur, requiring a quick momentary increase of the power fan.

136 General testing

137 The testing of the cooling system is rather simple. We used some burn-in test
 138 programs to stress the high performance cards and monitored their temperature.

¹³⁹ See [4] for further details

¹⁴⁰ (3) Application

¹⁴¹ Use case

¹⁴² We built and have been continuously using this cooling system for a GPU
¹⁴³ NVIDIA Tesla V100 and an FPGA Xilinx Alveo U280. Both cards are mounted
¹⁴⁴ in a common PCIe motherboard ASUS PRIME Intel H510M-E, installed in a
¹⁴⁵ standard rack-mounted computer case, running in an air-conditioned datacen-
¹⁴⁶ ter. The setup has proved to be flexible and extremely reliable. All design and
¹⁴⁷ built details are available at [4].

¹⁴⁸ Reuse potential and adaptability

¹⁴⁹ The setup can be easily adapted for any other card or element requiring similar
¹⁵⁰ cooling. The 3D-printed PLA air duct must be customized for each card and
¹⁵¹ computer case, but the overall building, calibration, and testing are essentially
¹⁵² the same. For a suggested customization procedure for the air ducts, see the
¹⁵³ full build instructions at [4].

¹⁵⁴ (4) Build Details

¹⁵⁵ Availability of materials and methods

¹⁵⁶ Only standard, easily available, components and simple 3D-printed pieces are
¹⁵⁷ employed in our design. We summarize below the main employed components
¹⁵⁸ and materials. See [4] for the full list.

- ¹⁵⁹ 1. Arduino nano microcontroller.
- ¹⁶⁰ 2. High-performance $60 \times 60 \times 38$ mm 4-pin PWM cooler fan as, for instance,
 - ¹⁶¹ (a) Sanyo Denki model 9GA0612P1K611 [7] or
 - ¹⁶² (b) Delta Electronics model PFC0612DE. [8].
- ¹⁶³ 3. Negative Temperature Coefficient (NTC) thermistor with $R_{25} = 50\text{ k}\Omega$.
- ¹⁶⁴ 4. High temperature Polylactic Acid (PLA) filament (also called “Premium
¹⁶⁵ PLA”). Most regular PLA must not be used with temperatures above
¹⁶⁶ 50°C).
- ¹⁶⁷ 5. Adhesive EPDM rubber foam tape, usual electronics components, usual
¹⁶⁸ bolts, etc.

¹⁶⁹ Ease of build

¹⁷⁰ No advanced skills are required. The most critical step in the building is the
¹⁷¹ customization of the air-ducts, which involves some careful measurements and
¹⁷² a 3D modeling. See the full build instructions at [4] for the details of the two
¹⁷³ already built cases.

¹⁷⁴ **Operating software and peripherals**

¹⁷⁵ We use an Arduino nano microcontroller and, hence, we need the Arduino soft-
¹⁷⁶ ware, which is free and widely available.

¹⁷⁷ **Dependencies**

¹⁷⁸ There is no relevant dependencies for the design.

¹⁷⁹ **Hardware documentation and files location:**

¹⁸⁰ Name: GitHub

¹⁸¹ Persistent identifier: <https://github.com/RodLophus/ArduinoFanController>

¹⁸² Publisher: Rodolfo Manin

¹⁸³ Date published: 2022-08-XX

¹⁸⁴ Hardware License: CERN-OHL-S-2.0

¹⁸⁵ Software and Documentation License: MIT License

¹⁸⁶ **(5) Discussion**

¹⁸⁷ **Conclusions**

¹⁸⁸ We have presented and built an open-source, Arduino-based, programmable
¹⁸⁹ active cooling system for a GPU NVIDIA Tesla V100 and an FPGA Xilinx
¹⁹⁰ Alveo U280 cards that allowed their installation and use in common computer
¹⁹¹ cases. Our inexpensive setup, which has been continuously used it in a high-
¹⁹² performance computing environment, has proved to be flexible and extremely
¹⁹³ reliable. Furthermore, it can be easily adapted for any other card or element
¹⁹⁴ requiring similar cooling.

¹⁹⁵ **Paper author contributions**

¹⁹⁶ Rodolfo Manin: 3D printing of the prototype pieces; mechanical and electronic
¹⁹⁷ assembling, Arduino programming.

¹⁹⁸ Alberto Saa: acquiring resources and funding.

¹⁹⁹ Both authors contributed equally to the conceptualization, design, testing, and
²⁰⁰ validation of the system, and to the paper and documentation writing.

²⁰¹ **Acknowledgements**

²⁰² AS thanks Vitor Cardoso and José S. Lemos for the warm hospitality at the
²⁰³ Center for Astrophysics and Gravitation of the University of Lisbon, where this
²⁰⁴ work was concluded.

205 **Funding statement**

206 AS acknowledges the financial support of CNPq (grant 302674/2018-7) and
207 FAPESP (grant 21/09293-7).

208 **Competing interests**

209 The authors declare that they have no competing interests.

210 **References**

- 211 [1] See, for instance, the SuperMicro GPU SuperServer SYS-420GP-TNR at
212 <https://www.supermicro.com/en/products/system/GPU/4U/SYS-420GP-TNR>
- 213 [2] See, for instance, the specifications of the NVidia Tesla P100 PCIe GPU
214 accelerator at
215 <https://images.nvidia.com/content/pdf/tesla/NV-tesla-p100-pcie-PB-08248-001-v01.pdf>
- 216 [3] M. Spetko, O. Vysocky, B. Jansik B, L. Riha, *DGX-A100 Face to Face*
217 *DGX-2 – Performance, Power and Thermal Behavior Evaluation*, Energies
218 **14**, 376 (2021). <https://doi.org/10.3390/en14020376>
- 219 [4] <https://github.com/RodLophus/ArduinoFanController>
- 220 [5] <https://www.amazon.com/Temperature-Control-Speed-Controller-Module/dp/B019P0FLHW>
- 221 [6] D. Ibrahim, *Microcontroller-Based Temperature Monitoring and Control*,
222 Newnes - Elsevier, 2002.
- 223 [7] <https://products.sanyodenki.com/en/sanace/dc/dc-fan/9GA0612P1K60>
- 224 [8] <https://www.delta-fan.com/Download/Spec/PFC0612DE-F00.pdf>