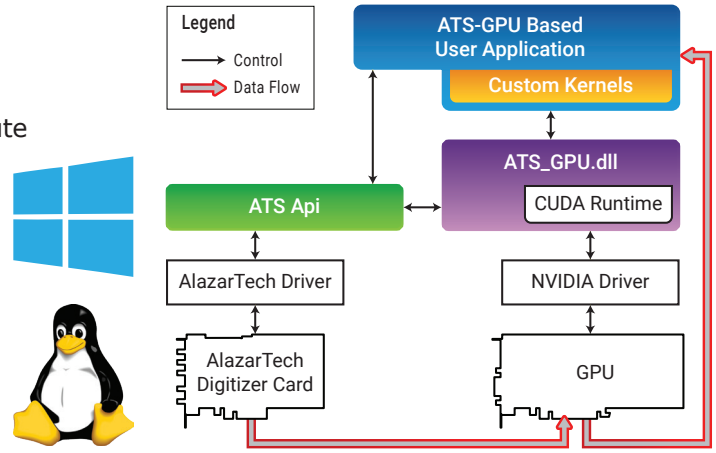
ATS-GPU-BASE

Real Time Signal Processing Software

- Transfer A/D data to GPU at high speed
- Up to 22 GB/s transfer rate for PCIe Gen 4 digitizer boards
- Supports CUDA®-enabled GPUs with compute capability 3.0 to 8.6‡
- Designed to work with AlazarTech® PCI Express waveform digitizers
- Availability of an optional OCT Signal Processing Library that includes:
  - ◇ Very high-speed floating-point FFT
  - ◇ Dispersion compensation and windowing functions
- Compatible with Windows® & Linux®



Product	GPU Compatibility	Operating System	Kernel Programming Language	Throughput to GPU
ATS-GPU-BASE version 24.2	CUDA compute capability 3.0 to 8.6‡	64-bit Windows & 64-bit Linux	CUDA	Up to 22 GB/s

Overview

ATS-GPU-BASE is a software library developed by AlazarTech that transfers data acquired by its family of PCI Express waveform digitizers to a CUDA-enabled Graphical Processing unit (GPU) at sustained transfer rates as high as 22 GB/s.

Interfacing waveform digitizers to GPUs involves creating a software mechanism to move data from one to the other and back to user buffers. The standard techniques used most often can get the job done, but feature very low data throughput due to software overheads.

AlazarTech designed ATS-GPU to eliminate this software bottleneck so that data can be moved from AlazarTech digitizers to GPUs and from GPUs to user buffers at full PCIe bus speeds.

Once the data is available in GPU memory, many types of digital signal processing (DSP) can be done on this data at near-hardware speeds.

Data transferred by ATS-GPU-BASE is presented in GPU memory as a buffer queue. Expert-level GPU programmers can create very high-performance custom kernels to manipulate this data using an easy-to-use application programming interface (API).

ATS-GPU-BASE includes an example program that demonstrates how to use the ATS-GPU library to transfer data from a waveform digitizer to a GPU. The example also shows how to do simple data processing on the GPU using CUDA kernels, and how to transfer the processed data back to host memory (RAM). Users can use this example program as a starting point to create their own custom kernels to do GPU-based DSP.

An example of a high-performance kernel is the optional ATS-GPU-OCT library (sold separately), which acquires data using ATS-GPU-BASE and then implements full OCT signal processing algorithm using CUDA kernels. ATS-GPU-OCT was benchmarked at up to 950,000 4K FFTs per second, demonstrating the power and efficiency of the ATS-GPU-BASE platform. Kernel code running on the GPU can apply a windowing function, do a floating point FFT, calculate the amplitude, and convert the result to a log scale. Please refer to the ATS-GPU-OCT datasheet for more information.

ATS-GPU-BASE Limitations:

Technical Support: Users should note that technical support for ATS-GPU-BASE is limited to the published user manual; no other technical support will be provided.

Programming language: Note that ATS-GPU-BASE development can only be done using C/C++; it is not possible to program in Python, MATLAB, or LabVIEW.

Required programming knowledge: Users must have expert programming knowledge of CUDA development in order to customize ATS-GPU kernels.

Use-case approval requirement: In order to avoid possible disappointment in product functionality and to ensure that customers are selecting the correct solution, AlazarTech requires that customers complete a questionnaire prior to purchasing ATS-GPU-BASE as a standalone library.

GPU-Based Signal Processing

Graphical Processing Units (GPUs) were originally designed for rendering high-quality video for gaming applications, which required being able to perform massive amount of real-time calculations. The highly parallel architecture of modern GPUs also makes them an ideal platform for digital signal processing (DSP) and high performance computing (HPC) systems.

In the past, complex real-time signal processing, such as FFT, correlation, FIR filtering etc., could only be achieved using dedicated DSP processors or by implementing the algorithms inside an FPGA or an ASIC. All these methods are non-trivial, expensive, time consuming and require highly specialized engineering skills.

Using GPUs, users can implement any algorithm that can be parallelized in a GPU using well known software techniques and gain a better than 10-fold improvement over CPU based signal processing. The reason why GPUs perform so well for DSP applications is that they contain hundreds of processing cores (kernels) running in parallel, while sharing a very high speed graphical memory bank.

Latency

ATS-GPU-BASE uses multiple CUDA streams to move data between the digitizer and GPU. This means there is a latency between data being acquired by the digitizer board and GPU receiving this data. The exact latency is determined by the buffer size used as well as the transfer rate of the PCIe link, but typical values are in in the range of several milliseconds.

Data Throughput Benchmarks

Benchmarks were done on AlazarTech ATS®9373 in an Intel i9-7900X 10-Core @ 3.3 GHz system with an ASUS® x299 motherboard, 32 GB DDR4, and NVIDIA® Quadro® P5000 GPU. The following throughput was measured in continuous streaming mode:

Buffer Size	Throughput*
1 MB	6.8 GB/s
2 MB	6.8 GB/s
4 MB	6.9 GB/s
8 MB	6.9 GB/s

* Based on benchmarks done in November 2018

Tests in an Intel i7 5930k 6-core @ 3.5 GHz system with an ASUS x99 Deluxe motherboard, 64 GB DDR4, and the same NVIDIA Quadro P5000 GPU produced very similar results.

A Typical ATS-GPU-BASE Application

A typical user application that uses ATS-GPU-BASE consists of the following minimum sections:

1) User application sets up waveform digitizer hardware (sample rate, input range, trigger parameters etc.).
2) User application allocates buffers and sets up the GPU.
3) User application starts data capture.

   ATS-GPU-BASE starts streaming data to GPU, one buffer at a time.
4) User-written GPU kernels do the following:
   - Process a buffer;
   - Copy result buffer to user memory;
   - Get next buffer, and repeat.
5) User application running on CPU consumes result buffer.

   For highest performance, make sure data consumption is faster than the rate at which result buffers are supplied by GPU kernels.
6) This continues until the application has to be closed.

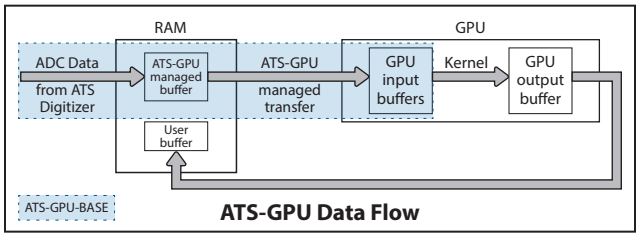
ATS-GPU-BASE and CUDA Runtime Library

ATS-GPU-BASE is shipped with a specific version of CUDA runtime library and links statically to it.

Programmers are allowed to use a different version of CUDA runtime library for their custom kernel code. NVIDIA guarantees that the two versions of CUDA runtime libraries will be interoperable.

Note: ATS-GPU only supports Windows versions and Linux distributions that are supported by NVIDIA's CUDA Toolkit. 32-bit operating system support is also similarly limited by NVIDIA. In particular, the ATS-GPU-OCT Signal Processing library cannot be built as a 32-bit library. We currently use CUDA toolkit 10.2, older versions are untested.

ATS-GPU Data Flow



ATS-GPU Data Flow

ATS-GPU-BASE is supplied with an example user application in source code. The application includes GPU kernels that use ATS-GPU-BASE to receive data, do very simple signal processing (data inversion), and copy the processed (inverted) data back to a user buffer. All this is done at the highest possible data transfer rate.

Programmers can replace the data inversion code with application-specific signal processing kernels to develop custom applications.

Boxcar Averaging

Version 23.1.0 and higher of ATS-GPU-BASE includes C, MATLAB, and Python Boxcar Averaging example programs that provide the ability to perform real-time boxcar averaging on signals acquired by AlazarTech waveform digitizers. It uses optimized GPU routines that allow raw data acquisition rates up to 6.9 GB/s.

This signal processing module can lead to a major improvement of signal-to-noise ratio without using CPU resources and without doing FPGA programming.

Buffer Averaging

Version 24.1.0 and higher of ATS-GPU-BASE includes MATLAB and Python Buffer Averaging example programs that configure an ATS digitizer to make an AutoDMA acquisition, and performs real-time on-GPU averaging of the acquired data buffers.

Record Averaging

Version 24.1.0 and higher of ATS-GPU-BASE includes a MATLAB Record Averaging example program that configures an ATS digitizer to make an AutoDMA acquisition, and performs real-time on-GPU record averaging of the acquired data. The records are averaged over a number of records specified by the user.

Successive Acquisitions

Version 24.2.1 and higher of ATS-GPU-BASE includes a C sample program that demonstrates how to configure an ATS digitizer to make successive AutoDMA acquisitions and perform on-GPU processing without reallocating buffers between acquisitions.

Performance Dependencies

Since the host CPU is involved in moving data to and from the GPU and in scheduling GPU kernels, CPU speed and motherboard's memory bandwidth can have a significant impact on the overall performance.

The optional ATS-GPU-OCT Signal Processing Library was used to benchmark performance.

On an ASUS X299 motherboard that uses an Intel i9-7900X 3.3 GHz CPU and DDR4 memory (32 GB RAM), a combination of the ATS9373 and NVIDIA Quadro P5000 (Pascal) GPU was able to do a 2048 point FFT at a rate of 1900 kHz.

An older DDR3-based machine performed significantly slower.

Complexity of the kernel code running on the GPU can have a significant impact on the overall performance. Users should optimize their code to take advantage of the GPU's high speed memory.

Computer Power Supply

GPUs are power hungry. Even consumer-grade models such as ASUS GTX980 require a power supply that can provide at least 500 Watts of power. As such, users must make sure their computer's power supply has sufficient capacity.

Compatible GPUs

ATS-GPU is designed to be compatible with CUDA-enabled GPUs with compute capability 3.0 to 8.6‡. Testing was done using NVIDIA Quadro P5000 and NVIDIA GeForce® RTX 2080 Ti.

It should be noted that ATS-GPU supports only one GPU at a time. If you have multiple GPUs installed in your computer, ATS-GPU will let you select one of them for use.

Compatible Waveform Digitizers

All AlazarTech PCI Express and Thunderbolt 3 waveform digitizers are compatible with ATS-GPU. Only single-board configurations are supported at this time.

AlazarTech's PCI bus waveform digitizers are not supported, as the host CPU is more than capable of handling data rates generated by PCI bus boards.

ATS-GPU cannot directly be interfaced with non-AlazarTech waveform digitizers.

Data Throughput to GPU

The data transfer rate to GPU is dependent on the generation of PCI Express digitizer board used or Thunderbolt 3 system configuration:

PCIe Link Speed	Transfer Rate
Gen 4 x16: ATS9376, ATS9470, ATS9473	Up to 22 GB/s
Gen 3 x8: ATS9373, ATS9371, ATS9637, ATS9437	Up to 6.9 GB/s
Gen 2 x8: ATS9360, ATS9416	Up to 3.5 GB/s
Gen 3 x4: ATS9364	Up to 3 GB/s
Gen 2 x4: ATS9872, ATS9352, ATS9353, ATS9628, ATS9428 Gen 1 x8: ATS9870, ATS9350, ATS9351, ATS9625, ATS9626, ATS9440	Up to 1.6 GB/s
Gen 1 x4: ATS9462	Up to 720 MB/s
Gen 1 x1: ATS9146, ATS9182, ATS9130, ATS9120	Up to 200 MB/s
Thunderbolt 3: ATST364, ATST146, ATST352 *dependent on system configuration	Up to *2.6 GB/s

Electronic Delivery

As of June 2020, AlazarTech software products are only available as a digital download. Customers who purchase ATS-GPU-BASE must provide a valid email address to receive their serial number, download link, and required license key.

Software Licensing Policy

Users are allowed to freely distribute the ATS-GPU-BASE library as long as they have purchased one ATS-GPU-BASE license and there is an AlazarTech PCI Express waveform digitizer present in the same computer.

Users must purchase a separate license for each computer on which ATS-GPU is installed.

In no case is the user allowed to distribute or share the source code of ATS-GPU with other users.

Annual Subscriptions

The purchase of an ATS-GPU-BASE license includes a subscription that allows users to download ATS-GPU-BASE updates from the AlazarTech website for a period of 12 months from the date of purchase.

Customers who want to download new releases beyond this 12 month period must purchase extended maintenance.

Additional add-on libraries for ATS-GPU-BASE, such as the *ATS-GPU-OCT Signal Processing Library* are not covered by the annual subscription, i.e. holders of an annual subscription will have to purchase subscriptions for additional libraries separately.

Writing Custom GPU Kernels

ATS-GPU-BASE includes an example program in C/C++ source code, which implements very simple GPU kernels that invert data and write it back to a buffer in computer memory.

Expert-level GPU programmers who need to write their own kernels should start with the included source code, add CUDA code in the appropriate place, and compile their libraries.

The example program is provided with a Visual Studio project and a CMake build file. We use more recent C++ features, and Visual Studio 2015 and later is required. On Linux, a C++11 compiler is required and can be accessed on older distributions via a devtoolset (RHEL and CentOS 6 for example).

Writing, testing, and debugging modified kernels will be the sole responsibility of the user and AlazarTech will not be responsible for assisting the user with such custom modifications.

Users must have expert programming knowledge of CUDA development in order to customize ATS-GPU kernels.

Note that technical support for ATS-GPU-BASE is limited to the published user manual; no other technical support will be provided.

Extended Maintenance

Customers can extend their ATS-GPU-BASE subscription by ordering the 1 year extended maintenance for ATS-GPU-BASE (order number ATSGPU-002).

This must be purchased before expiration of the standard subscription (or before expiration of an extended subscription).

If the subscription lapses, renewal at a later date will be subject to a reinstatement fee to cover the administrative costs of reinstatement. Furthermore, subscription extensions must be purchased to cover the lapsed period.

Get your subscription end date by registering your product at: www.alazartech.com/en/my-account/my-products/. You will need the product serial number, which can be found in the email you received with your download link and password. In the case of older purchases, the serial number can be found on the CD envelope.

Subscription extensions will not be offered for discontinued products.

‡ Version 4.1 or higher of ATS-GPU-BASE is required for support of GPUs with CUDA-compute capability 3.0 to 8.6.
Version 4.0.1 provides support for compute capability 3.0 to 7.5.

ORDERING INFORMATION

ATS-GPU-BASE: GPU Streaming Library License + 1 Year Subscription	ATSGPU-001
ATS-GPU-BASE-1YR: 1 year extended maintenance for ATS-GPU-BASE	ATSGPU-002
ATS-GPU-OCT: Signal Processing Library License + 1 Year Subscription (requires ATSGPU-001; also requires ATS-SDK for use with Python, MATLAB, & LabVIEW)	ATSGPU-101
ATS-GPU-OCT-1YR: 1 year extended maintenance for ATS-GPU-OCT	ATSGPU-102
ATS-GPU-NUFFT: ATS-GPU-OCT Extension for fixed-frequency sampled data License + 1 Year Subscription (requires ATSGPU-001 and ATSGPU-101)	ATSGPU-201
ATS-GPU-NUFFT-1YR: 1 year extended maintenance for ATS-GPU-NUFFT	ATSGPU-202
ATS-SDK purchased with a digitizer board or ATS-GPU: License + 1 Year Subscription (Supports C/C++, Python, MATLAB, and LabVIEW)	ATS-SDK
ATS-SDK purchased separately: License + 1 Year Subscription + 5 hours of technical support (Supports C/C++, Python, MATLAB, and LabVIEW)	ATS-SDK-WOD

ATS-GPU main API functions

```
ATS_GPU_AbortCapture
ATS_GPU_AllocBuffer
ATS_GPU_FreeBuffer
ATS_GPU_GenerateCPUBoxcarFunction
ATS_GPU_GetBuffer
ATS_GPU_GetVersion
ATS_GPU_PostBuffer
ATS_GPU_QueryCUDADeviceCount
ATS_GPU_QueryCUDADeviceName
ATS_GPU_SetCUDAComputeDevice
ATS_GPU_Setup
ATS_GPU_StartCapture
```


Manufactured By:

Alazar Technologies, Inc.

6600 TRANS-CANADA HIGHWAY, SUITE 310
POINTE-CLAIRE, QC, CANADA H9R 4S2

TOLL FREE: 1-877-7-ALAZAR OR 1-877-725-2927
TEL: (514) 426-4899 FAX: (514) 426-2723

E-MAIL: sales@alazartech.com

DATASHEET REVISION HISTORY

Changes from version 23.1 (Dec 2023) to version 24.2	Section, Page
Updated maximum data throughput to 22 GB/s (with PCIe Gen 4 x16)	Global change
Specified available Boxcar Averaging sample programs (C, MATLAB, Python)	Boxcar Averaging, pg. 2
Added section on Buffer Averaging	Buffer Averaging, pg. 2
Added section on Record Averaging	Record Averaging, pg. 2
Added section on Successive Acquisitions	Successive Acquisitions, pg. 2
Added Gen 4 x16	Data Throughput to GPU, pg. 3
Modified reinstatement fee details	Extended Maintenance, pg. 3
Added `ATS_GPU_GenerateCPUBoxcarFunction` and `ATS_GPU_GetVersion`	ATS-GPU main API functions, pg. 4

Changes from version 4.2.2 (Nov 2022) to version 23.1	Section, Page
Added section on Boxcar Averaging	Boxcar Averaging, pg. 2
Added new products to table	Data Throughput to GPU, pg. 3
Replaced password with license key	Electronic Delivery, pg. 3
Modified to include new subscription reinstatement policy	Extended Maintenance, pg. 3

Changes from version 4.2 (Aug 2022) to version 4.2.2	Section, Page
Updated CUDA Compute Capability: ATS-GPU-BASE now supports compute capability 3.0 to 8.6	Global change
Added ATS9364, ATS9182, and ATST364	Data Throughput to GPU, pg. 3
Updated ATS-GPU licensing policy: a separate license is required for each computer	Software Licensing Policy, pg. 3

Changes from version 4.1a (Nov 2021) to version 4.2	Section, Page
Updated ATS-GPU-BASE version number	Feature Table, pg. 1
Added section: *ATS-GPU-BASE Limitations*	ATS-GPU-BASE Limitations, pg. 1
Added Thunderbolt 3 digitizers	Compatible Waveform Digitizers, pg. 2
Added ATS9872, ATS9353, and Thunderbolt 3 digitizers	Data Throughput to GPU, pg. 3
Updated section to remove technical support from list of subscription benefits	Annual Subscriptions, pg. 3
Added note about technical support being limited to the published user manual	Writing Custom GPU Kernels, pg. 3
Updated section to remove technical support from maintenance extension	Extended Maintenance, pg. 3
Updated descriptions for maintenance items *ATSGPU-002, ATSGPU-102 & ATSGPU-202:* Removed technical support from maintenance extension	Ordering Information, pg. 4
Added products ATS-SDK and ATS-SDK-WOD	Ordering Information, pg. 4

Changes from version 4.1 (June 2020) to version 4.1a	Section, Page
Updated product registration URL	Extended Support & Maintenance, pg. 3

Changes from version 4.0b (Jan 2020) to version 4.1	Section, Page
Updated CUDA Compute Capability: ATS-GPU-BASE 4.1 supports compute capability 3.0 or higher	Global change
Updated introductory text Updated benchmark data from 900,000 4K FFTs per second to 950,000 4K FFTs per second Removed zero-padding from signal processing examples because it is the responsibility of the user	Overview, pg. 1
Added section on Latency	Latency, pg. 1
Updated CUDA toolkit in use to version 10.2	ATS-GPU-BASE and CUDA Runtime Library, pg. 2
Added NVIDIA GeForce RTX 2080 Ti to GPUs used for testing	Compatible GPUs, pg. 2
Divided table using the number of PCIe lanes	Data Throughput to GPU, pg. 2
Added section: *Electronic Delivery*	Electronic Delivery, pg. 2
Updated location of serial number	Extended Support and Maintenance, pg. 3
Added ATS-GPU-NUFFT library extension order information	Ordering Information, pg. 3

DATASHEET REVISION HISTORY

Changes from version 4.0a (May 2019) to version 4.0b	Section, Page
Updated CUDA Compute Capability: ATS-GPU-BASE now supports compute capability 3.0 to 7.5	Global change
Added new products (ATS9352, ATS9146) to data transfer rate table	Data Throughput to GPU, pg. 2

Changes from version 4.0 (Jan 2019) to version 4.0a	Section, Page
Added paragraph about support and updates beyond the included 12 months	Annual Subscriptions, pg. 3
Added section: *Extended Support & Maintenance*	Extended Support & Maintenance, pg. 3