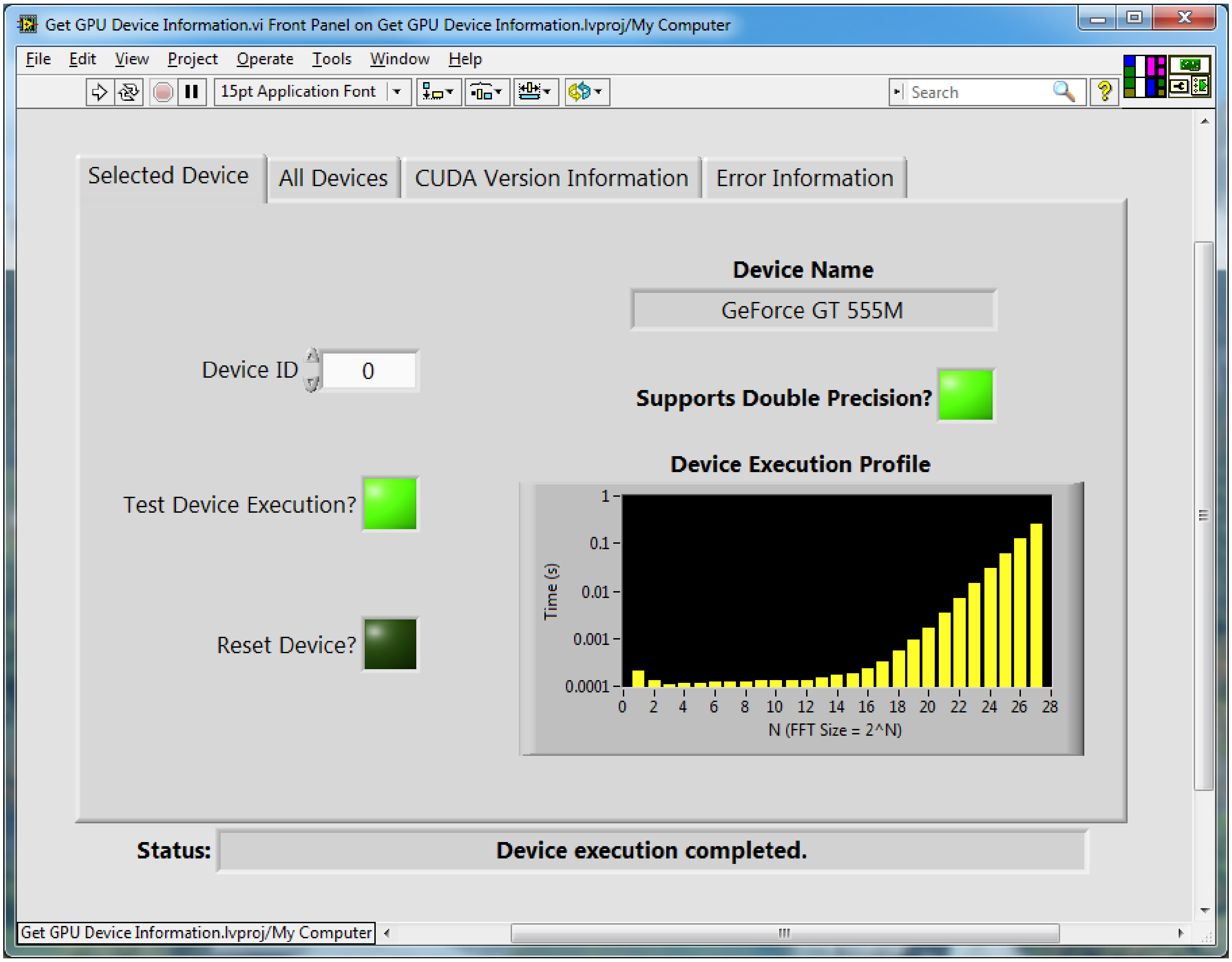
Get GPU Device Information



Device Name: GeForce GT 555M

Device ID: 0

Supports Double Precision? ●

Test Device Execution? ●

Reset Device? ○

Status: **Device execution completed.**

Overview

This example investigates GPU devices installed in the system and optionally executes code on one of those devices. The information gathered includes the number of GPU devices detected, the properties for each detected device including model name and computing capability (major and minor), and the version information for each CUDA component used during the inspection.

An optional device execution test iteratively computes a 1D complex single-precision FFT on progressively larger signal sizes until the maximum size is reached or a time limit is exceeded (1 sec). The execution time for each FFT call is graphed in **Device Execution Profile**. A device reset option is supplied to restore the GPU device state in case exceptional conditions are encountered.

Usage

Inputs

Tab»Selected Devices

Device ID selects the GPU device in the system used for execution testing (*optional*).

Test Device Execution? determines if the execution test is run on the GPU device identified by **Device ID**. FFT execution times are displayed in **Device Execution Profile**.

Reset Device? stipulates whether the GPU device is reset after running the execution test.

> *NOTE: Resetting a device frees all resources and stops all computations on that device even those created and run from other LabVIEW applications – not just this example.*

Tab»Error Information

error in contains an error condition (if any) passed to this function.

Outputs

Tab»Selected Devices

Device Name contains the model name of the selected device.

Device Execution Profile charts the execution time of the FFT computation in each test iteration. It does not include time for any initialization processes such as allocating device memory.

Tab»All Devices

Available Devices shows the number of GPU devices available for GPU computing.

> *NOTE: If the number of available devices is 0 or less, the cause may be due to Remote Desktop. See* **Troubleshooting** *below for issues related to remote connections.*

Device Properties is a collection of device properties for each available device.

Tab»CUDA Version Information

CUDA Runtime API Version is the version of the CUDA Runtime API used to compile device information. This also corresponds to the version of the CUDA Toolkit in use.

CUDA Driver Version is the version of the NVIDIA display driver in use. This should not be confused with the version of the CUDA Driver API which is the same as the CUDA Runtime API and the CUDA Toolkit.

CUDA Build Version is the version of the CUDA Toolkit used to build the GPU Analysis Toolkit. This version influences the interfaces and internal settings used to communicate with NVIDIA's GPU devices.

Tab»Error Information

error out contains the first error condition (if any) generated by this example.

Troubleshooting

During the inspection or execution phases of this example, any number of error conditions is possible. Status and error information are updated at runtime to help resolve when an error occurs.

The following list enumerates the different error conditions states generated by the example when an error is detected. Each item describes the condition and offers possible solutions.

Running this example from a remote connection can produce most of the error conditions below. Under Remote Desktop, most NVIDIA display drivers cannot detect GPU devices installed in the system.

In most cases, disconnecting a remote session will not allow the NVIDIA display driver to see GPU devices. The system must be logged into locally or restarted.

Error Information

Previous error detected.

Situation: An error was passed to this example.
Cause: Unknown.
Resolution: Examine the code executed prior to calling this example.

Device count inquiry failed. Is remote desktop active?
No devices detected. Is remote desktop active?

Situation: The version of NVIDIA display driver is insufficient.
Cause: The CUDA Toolkit version is more recent.
Resolution: Either install the version of the CUDA Toolkit that matches the driver or update the driver to a version which is at least as great as the toolkit's. See **Online NVIDIA Resources** below for links to available installers.

Situation: The NVIDIA display driver used to communicate with GPU devices is not active.
Cause: (In Windows) Remote desktop is active.
Resolution: Log into the workstation locally to reactivate the NVIDIA display driver.

Situation: The function used to enumerate devices is not available.
Cause: A supported CUDA Toolkit is not installed.
Resolution: Install the missing CUDA component(s).

Error retrieving version information.

Situation: An external function used to gather version information is not available.
Cause: One of the required external libraries is not installed.
Resolution: Check that all required CUDA components are installed and satisfy the minimal version. This includes the CUDA Toolkit (***cudart*.****) and NVIDIA display driver (***nvcuda.****). An error related to the build version implies the GPU toolkit installation is corrupt (***lvcuda*.****). The **error out** cluster has additional information regarding which of the external libraries is involved.

Error retrieving device properties.

Situation: An external function used to gather version information is not available.
Cause: The NVIDIA display driver is not active or has the wrong version.
Resolution: Check the version of NVIDIA display driver installed in the system. If its version is not the same or greater than the CUDA Toolkit version being used, download a newer driver from NVIDIA's website.

Device execution failed.

Situation: An error is returned from the device execution test.
Cause: Unknown.
Resolution: The test is designed to produce only one failure when the FFT size limit is exceeded. When this error is produced, it is masked and not returned as an error condition. Details in the source string of the **error out** cluster may pinpoint which of these device operations were responsible:

- initializing the device,
- initializing the FFT library,
- allocating the device memory for the FFT,
- computing the FFT,
- freeing the device memory,
- releasing the FFT library,
- releasing the device, or
- resetting the device (if **Reset Device?** is ***true***)

Use this error information to determine which online NVIDIA document may help resolve the issue. For a GPU device with limited memory, the test may fail when allocating device memory. Such an error case is rare and not trapped. The **Device Properties** output contains the amount of global memory available for each device.

CUDA Version Information

CUDA Version entries are empty

Situation: An error occurred before calling the functions which return version details.

Cause: Acquiring the number of devices failed.

Resolution: Check Error Information for any errors reported.

CUDA Runtime API Version is 0.0

Situation: Retrieving the version from the CUDA Toolkit failed.

Cause: The function in the CUDA Toolkit returning this information did not execute properly.

Resolution: While this failure should produce an error code in most cases, certain special conditions will not report a function call failure. For example, using a 32-bit CUDA Toolkit from 64-bit LabVIEW produces this result.

CUDA Driver Version < CUDA Runtime API Version

Situation: The version of the NVIDIA Display Driver is insufficient for the CUDA Toolkit version being used.

Cause: The CUDA Toolkit installed is a later version than the current NVIDIA display driver.

Resolution: This scenario also produces an error code with details explaining the version conflict. Under most cases, Windows Update recommends and installs the most recent stable NVIDIA display drivers. If you installed a release candidate of the CUDA Toolkit or have not installed a recent version of the NVIDIA display driver since installing the CUDA Toolkit, this is an indication that your current driver is inadequate. See **Online NVIDIA Resources** below for links to available installers.

CUDA Build Version < CUDA Driver Version or CUDA Runtime API Version

Situation: This is expected.

Cause: The GPU Analysis Toolkit used a current version of the CUDA Toolkit to derive its interfaces and internal settings during development.

Resolution: None is required. The GPU Analysis Toolkit queries the system for an installed CUDA Toolkit and uses it to communicate with the NVIDIA GPU devices. To date, this toolkit has been successfully tested with CUDA Toolkit v4.x and v5.0 (release candidate 5.0.7).

Online NVIDIA Resources

http://developer.nvidia.com/cuda-downloads : Links to the latest CUDA installers.

http://developer.nvidia.com/cuda-gpus : Lists the compute capability of an NVIDIA GPU which determines what CUDA operations are supported.

http://developer.nvidia.com/nvidia-gpu-computing-documentation : A collection of user guides and manuals on the CUDA Toolkit and related CUDA-based libraries (e.g. CUFFT and CUBLAS).