

TOSHIBA TELI CORPORATION

4-7-1, Asahigaoka, Hino, Tokyo 191-0065, Japan

GenICam: What does it taste like?

■ Expected readers

Software developers whom design/use an image acquisition software in a machine vision domain, or project managers whom have to manage such developers and make them more productive.

■ Keywords

Camera, compatibility, image acquisition software, machine vision, porting, productivity, replacement

■ Introduction

Perhaps you could have heard about GenICam more or less so far but probably I should firstly tell you why we decided to write this article. The reason is very simple: GenICam gains your productivity on software development process and makes your software robust and stable. And of course, Toshiba Teli Corporation can support you with the technology! In this article, I'll introduce you what GenICam is and the benefit you would reap from it.

■ What is GenICam?

GenICam is a set of the machine vision standards which consists of GenApi, GenTL, SFNC, GenCP, and PFNC (and some more? Yes, it's been extending the domain where it covers even today). Due to the numbers of subsets, people may make its meaning vague but you should simply memorize it as a kind of useful standardized technologies. By the way, why should we rely on them? I would say it's because they provide you a unified and consistent way for everything related to image acquisition and device controlling tasks; the unified way blows away awful and redundant processes from your development cycle.

So let me ask you a question here: Why do you need a camera? Yes, because you need to get an image of a target object; it's a very primarily purpose. And why do you need an image acquisition software? Yes, because the image acquisition software acquires images from the camera for you; unless you get an image from the source, your job will not start. In addition, your purpose in the task should not be interfered by the specific equipment that you use; I mean, in principle, you should be

able to acquire images from any source just following a unified manner.

On the other hand, we as a camera manufacturer regularly releases products based on a latest technology at the moment. In a typical scenario, a customer might want to replace their owned camera with another one which equipped new features or drastically improved its quality. However, in the older generations, it was not always true that each camera is compatible from a controller side of view even though the cameras were released from the same manufacturer. One register is here but another is there, one register is an integer but another is a float, or some registers might be missing. Ahhh, it's annoying. Even today, you may have a chance to face such a situation where you have to handle such incompatible products even though you just wanted to make exposure time a little bit shorter! Oh, so ridiculous, I wish you could have cameras they have a single and identical interface to control its exposure time!

As I said in the earlier, you had faced a situation where you had to maintain cameras work in a different manner anyway. So can you guess what would happen if you have to support such a case in your code base? It's obvious that your source code will immediately get so inflated with a lot of patches which look very similar but not identical to cover the all cameras you control. Do you really like to support ugly and very large code chunks? Phew, here comes the place for the standardization benefit which GenICam provides you.

Following the unified technologies provided by GenICam, your work should be reasonably minimized and the code base turns into maintenance friendly with less effort (like a well-designed programming paradigm makes your software smaller/cleaner). As following a fact that I said "unified" way of GenICam, it allows you to easily switch the target camera from one to another because the interface provided by GenICam is still identical between those compliant cameras. It's no doubt that it is a good option for users because you can be released from the influence of compatibility issue and can use one which meets your any requirement such as price, feature, or quality at the moment. It should help you to shorten the time-to-market factor on your project because you can start the work relying on the existing well-maintained software. In addition, if a camera prove your software really works, you can be confident with yours. I feel it's very wonderful news for the programmers. If you found a bug in its life cycle? Let's just fix it because the next version you'll release should definitely be more robust and work with known bunch of cameras you've tested so far. This kind of positive cycle should make you more relaxed on your business working days.

■ What can Toshiba Teli Corporation do for you?

Toshiba Teli Corporation had the first participation for the GenICam face-to-face meeting held in 2004 and has been one of the active contributors on the GenICam activity since 2013 (especially active on a development of GenApi 3.0). In addition, our software development kit (SDK) for image acquisition is compliant with GenICam and can handle cameras which are compliant with GigE Vision and USB3 Vision standard. Even if you would like to design an image acquisition software by yourself

from scratch, we can support your development with pleasure if you need it; of course please don't forget to buy our cameras to make us happy! In addition, we can provide U3V GenTL Producer module if your software works as a GenTL Consumer module. Anyway, please don't hesitate to ask us if you have any question or concern on the software development process based on GenICam.

■ Terminology

GenICam™	A set of the following standardization activity: GenApi, GenTL, SFNC, PFNC, and GenCP.
GenApi	A C++ library which provides you a unified interface to control a camera.
GenTL	A C based library specification which is applied between an image producer and image consumer. It hides an on-the-wire technology behind a producer module.
SFNC	Stands for Standard Feature Naming Convention. It defines unified terminology which is used to control a camera or a general peripheral device.
PFNC	Pixel Format Naming Convention. It defines unified terminology which is used to handle various pixel data format (pixel data alignment, data depth, etc.) which is delivered from a camera.
GigE Vision™	A standardized protocol which is applied on a machine vision system based on gigabit Ethernet.
USB3 Vision™	A standardized protocol which is applied on a machine vision system based on USB3.
GenCP	A standardized camera control protocol.

■ Feedback

Your feedback is our motivation! Please drop us your message on the following URL if you have any question, suggestion, or findings.

https://www.toshiba-teli.co.jp/cgi/form/en/contact.cgi?toi_div=1

■ Resources

Toshiba Teli Corporation Support Page

<https://www.toshiba-teli.co.jp/cgi/ss/en/service.cgi>

You can download our SDK or other resources from the URL having created your account. You can create your account at the same URL.