BLKQCL SDK Control API Overview

Overview

The BLKQCL device is a smart, discoverable, networked device, accessible via ubiquitous networking, and controllable via a SOAP API.

This means that control is managed via XML-formatted messages exchanged over HTTP (TCP/IP). The API is as stateless as practical, so as to facilitate flexible programmability and control. It allows for concurrent (multiplexed) access, so that separate threads of control (communications channels) can be simultaneously reading back sensor status, while other control channels are driving the lasers and/or performing scan operations.

Discovery

In order to connect to the BLKQCL, you must know its IP address. The BLKQCL is completely standards compliant (SSDP and DHCP), and can also display its own IP address via the touchscreen UI.

However, the simplest programmatic way to 'discover' a BLKQCL device is via SSDP (Simple Service Discovery Protocol). If you have a windows computer, it already knows about SSDP, and your BLKQCL device will just 'show up' in your network neighborhood after you turn on the device (and plug it into your network).

SOAP & WSDL

SOAP (Simple Object Access Protocol) is a standards-based way to define networking control interfaces. It's just a way to format remote function calls as XML, send them over HTTP (TCP/IP), and get back the function call results in XML.

WSDL (Web Service Description Language) is a standards-based way to describe the format of the XML messages, and what operations are available on the BLKQCL.

This SDK comes with a WSDL file, which can be opened and imported into a very wide variety of WSDL aware tools. Supported tools include:

- SoapUI (URL)
- Java (eg. Eclipse)
- C#.net (2k13)

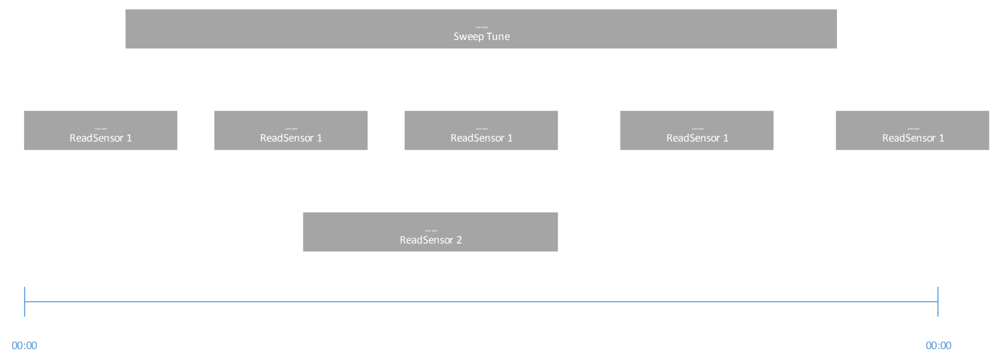
Developers can also quite easily programmatically generate SOAP messages (custom coded proxy) and use these to control the BLKQCL. We provide one such example using HTML5 and JavaScript.

Message Sequencing and Timing

The BLKQCL Control API allows for very flexible control over how you time operations. You can setup as many concurrent communications channels as you wish, to perform operations 'at the same time' as one another.

You can sequence commands one after the other, to enforce a particular order relationship with commands.

And after the fact, you can re-establish the exact relative timing of operations and sensor readings by examining the timestamp returned with each message.



Interfaces

WSDL allows us to logically divide the BLKQCL API into logically related sections. The most important interfaces for users getting started would be:

- IConfiguration
- IDeviceManagement
- ILaserOperation

IConfiguration

This interface is concerned with persistent state. Information set or received through this interface is preserved across reboots, and is unlikely to require change from use to use.

The interface provides these important operations:

- GetFactorySettings()
- GetUserSettings()
- SetUserSettings()
- ResetToFactoryDefaults()

IDeviceManagement

This returns a variety of operational status details which have little to do laser operations, but are more generally associated with generic device management and control.

- GetVersionDetails()
- GetDeviceName()
- SetDeviceName()
- GetPowerState()
- SetPowerState()
- GetAlarms()
- ClearAlarms()
- GetBatteryStatus()

ILaserOperations

Laser operations are the primary functions of the BLKQCL system. These APIs are stateless, in that none of their parameters are preserved, though they are modal in the sense that they drive the behavior of the BLKQCL device – and report back on that state.

- StopLasers()
- GetLaserPointerOn()
- SetLaserPointerOn()
- ReadSensors()
- MoveTune()
- StepTune()
- SweepTune()
- StepScan()
- SweepScan()
- InterleavedScan()

Proxies

These are important code utilities to help interface to the BLKQCL web service API from popular languages.

Reference API

See the BLKQCK-SDK-API-Reference.pdf for more details.