TANGO Interface to Industrial Systems

Pavel Bastl, Ondrej Janda ELI Beamlines/Institute of Physics,

June 7th, 32nd Annual TANGO collaboration meeting
ELI Beamlines, Dolni Brezany, Czech republic
INDUSTRIAL INTERFACES

- Long history of industrial control systems
- Standard solutions are used as observed by industry
- There are many industrial systems used in facilities
- There exists 'standard' specifications for signals/interfaces
- Industrial components are already build with these type of interfaces
- Good examples: Vacuum, Switch yard, Motion control etc.

INDUSTRIAL DISTRIBUTED SYSTEM

- Distributed control system is used in facilities
- Remote IO with well defined interfaces are available in industrial market
- Remote IO components from industry are cheap → Stepper drivers are good example

INDUSTRIAL COMMUNICATION

Called fieldbus
- The history of fieldbus is long
- There are many historical fieldbuses available, originaly based on RS2xx
INDUSTRIAL FIELDBUS

Modern fieldbuses are available on the market, most important requirements:
- Based on standard HW (ethernet)
- Support for optical fibers (naturally when based on ethernet)
- Scalable

INDUSTRIAL FIELDBUS IN ELI BEAMLINES
INDUSTRIAL FIELDBUS IN ELI BEAMLINES

Common approach is necessary with abstract layers

Two Abstract API are to be used
• Abstract API for fielbus control → csfbus
• Abstract API for IO interface → csiio (cs industrial IO)
/**
 * \brief The class defines generic field-bus interface
 */
class cSysbus : public cBus
{
public:

virtual FieldbusType getFieldType(void) = 0;

virtual bool connect(void) = 0;
virtual bool disconnect(void) = 0;

virtual PbusState initialize(void) = 0;
virtual PbusState shutdown(void) = 0;
virtual PbusResponse discover(ModuleSet &modules) = 0;
virtual PbusState start(void) = 0;
virtual PbusState stop(void) = 0;
virtual PbusState abort(void) = 0;

virtual const bool operator[](const std::size_t idx) const = 0;
virtual uint32_t size(void) const = 0;
virtual uint32_t ports(void) const = 0;
virtual uint32_t getPort(const uint32_t port, bool &isValid) const = 0;
virtual uint32_t portSize(const uint32_t port) const = 0;
virtual bool getBit(const uint32_t bit, bool &isValid) const = 0;
};

class DigitalInput : public Module
{
public:

DigitalInput(uint32_t bus, uint32_t slot, uint32_t function) : Module(bus, s
virtual ~DigitalInput();

CSIIOAPIType getApiType(void);

virtual const bool operator[](const std::size_t idx) const = 0;
virtual uint32_t size(void) const = 0;
virtual uint32_t ports(void) const = 0;
virtual uint32_t getPort(const uint32_t port, bool &isValid) const = 0;
virtual uint32_t portSize(const uint32_t port) const = 0;
virtual bool getBit(const uint32_t bit, bool &isValid) const = 0;
};

class DigitalOutput : public Module
{
public:

DigitalOutput(uint32_t bus, uint32_t slot, uint32_t function) : Module(bus, s
virtual ~DigitalOutput();

CSIIOAPIType getApiType(void) {return CSIO_API;};

};
**API Structure**

**FIELDBUS API**
- FBUS Interface
- Stack → ELI
- Discover()
- HW Interface

**CSIIO API**
- TANGO Interface
- CSIIO Interfaces
- XML Network
- Configuration

<<implements>>
<<creates>>
<<use>>
START TANGO SERVER ...

... CONNECT, FOR EXAMPLE GUI
CAMERA RELATED GUI

MOTION RELATED GUI

Example
Thank you for attention!