

- [Freelance IoT, Industrie 4.0 & Digitalization Consulting Service](#)
  - [AWS IoT Based Smart Urban Parking&Meterage PoC](#)
  - [Microservice PoC for a global brand- An International Hotel Booking System](#)
  - [MES&PLC Upgrade for a global automotive brand](#)
  - [HiPot Test SCADA System for a global semiconductor brand](#)
  - [Vision Inspection & Finish Handler System, 50+ systems](#)
  - [Atlas Copco Open Protocol C++ Implementation](#)
  - [SCPI & IEEE 488.2 C++ Implementation](#)
  - [RT-Thread&Pandora IoT Board on Linux](#)
  - [IIoT Semiconductor 2D Traceability System](#)
  - [IIoT Upgrade for Digital Oven Management System for Semiconductor](#)
  - [AGV&Milk Run Control Tower for Intelligent Factory](#)
- [Husky Injection Molding Systems](#)
  - [Team&Department Startup](#)
    - [Startup of Shanghai Machines Plant](#)
    - [Setup of Global Development Engineering Department\(Software\), Shanghai](#)
    - [Top-Ranked Team Leader](#)
  - [Non-Development Project Management](#)
    - [Global Product Launch Trip](#)
    - [Build Agile development and CI/CD process](#)
    - [Merging Moldflow's Manufacturing Solutions Unit](#)
  - [Product Architect&Development Management](#)
    - [Altanium Hot Runner Realtime Controller](#)
    - [Realtime Injection Molding Control System](#)
      - [Polaris Injection Control SCADA System](#)
    - [Injection Molding - PLC](#)
    - [Injection Molding - Robots&3rd-party Integration](#)
    - [Injection Molding - IoT&MES Integration](#)
    - [HyMET Hotrunner Heat logger](#)
    - [HMI Touchscreen Label Control](#)
    - [.NET HMI Architecture&Development](#)
    - [Shotscope C/S SCADA&MES System](#)
    - [Celltrack C/S SCADA&MES System](#)
    - [Shotscope NX B/S SCADA&MES System](#)
    - [HMI Internationalization\(i18n\) Database Design](#)
  - [Product Development - IoT, IIoT, Industry 4.0 & Cloud](#)
    - [Molds & Tooling Lifetime IoT Solution](#)
    - [Injection&Molding Industry-4.0 Integration - RT Communication](#)
    - [Injection&Molding Industry-4.0 Integration - Remote HMI](#)
    - [Injection&Molding Industry-4.0 Integration - Intelligent Controller Pairing](#)
    - [SSNX on Azure IIoT System](#)
    - [Hostlink Remote Machine Interface](#)
    - [Machine Intelligence Algorithm Modules](#)
    - [TwinCAT Realtime visualization microservices](#)
    - [TwinCAT 2 TSM Viewer](#)
  - [Big Data & Digital Transformation - DevOps](#)
    - [Whiteboard R&D Process BI Dashboard](#)
    - [Polaris Digital Production System For Injection Machines](#)
    - [Husky Software Internationalization\(i18n\) Management System](#)
    - [DevOps Console - Console3](#)
    - [Zebra search Engine](#)
  - [Bianque Digital Transformation - A Industry 4.0 System](#)
    - [Bianque Core Microservice Platform - Lancet](#)
    - [Bianque DevOps Platform - Acupuncture](#)
    - [Bianque Elastic Computing Platform - Transformer](#)
    - [Bianque Engine - HySearch](#)
  - [Open Source Software](#)
    - [Boost.Chrono C++ Library](#)
    - [The POCO C++ Libraries](#)
- [Dediprogram Inc.](#)
  - [SF100 Flash Programmer](#)
  - [WinUSB++, USB driver in C++](#)
- [STMicroelectronics Inc.](#)
  - [Embedded IC Verification & Reference Design Kit\(PC BIOS Memory\)](#)
  - [Embedded IC Verification & Reference Design Kit\(SPI/I<sup>2</sup>C Flash/EEPROM Memory\)](#)
  - [Egret Flash Memory Programmer](#)
  - [RTOS Porting - uCLinux](#)
  - [RTOS Porting - uc/OS-II](#)
  - [Anatidae Project Tracking&Syncing System](#)

- [Oscilloscope Simulator in C++](#)
- [Schneider Electric](#)
  - [Automatic Transfer Switching Equipment\(ATS\)](#)
  - [Air Circuit Breaker Localization Design](#)
  - [Development of XB2 industrial-strength LED alarms](#)
  - [Lab thermostat Design](#)
  - [Switching Power Supplies Localization](#)
  - [Lab Data Manager - LabUtils](#)
- [Harbin Institute of Technology](#)
  - [Researches On Digital Industrial Control&Automation](#)
  - [Automatic Test Equipment for Relays, Contactors & Circuit Breakers](#)
  - [Automatic Test Equipment for Electronic Ballast](#)
- [Sparetime/Open Source](#)
  - [A Literal Converter for Integers](#)
  - [Duplicate File Finder](#)
  - [2D Water Effect in WTL](#)

# Freelance IoT, Industrie 4.0 & Digitalization Consulting Service

## 1. AWS IoT Based Smart Urban Parking&Meterage PoC

2019 - 2019 Architecture, MCU, Firmware, AWS IoT

A Proof of Concept(PoC) demo requested by an international company, an Arduino&AWS based IoT solution for urban parking spots searching and meterage, see <https://smartparkingspots-doc.netlify.app> (<https://smartparkingspots-doc.netlify.app>)

Designed a smart parking solution that helps motorists find parking in downtown areas.

Requirements	Solution
1. <b>Hardware/Sensor technology</b>	Arduino Yun / simulated with EC2
2. <b>Connectivity technology</b>	MQTT, websocket
3. <b>scalable data processing architecture</b>	AWS IoT, Lambda, DynamoDB
4. <b>Parking data is made available</b>	API Gateway, SNS, S3
5. <b>security and protection</b>	IAM, Cognito, x.509 certificate authentication

## 2. Microservice PoC for a global brand- An International Hotel Booking System

2018 - 2019 Architecture, MCU, Firmware, AWS IoT

A Proof of Concept(PoC) demo requested by an international company, an international hotel booking system, see <https://hotelbooking.netlify.app/> (<https://hotelbooking.netlify.app/>)

Designed a smart parking solution that helps motorists find parking in downtown areas.

Requirements	Solution
1. <b>Hardware/Sensor technology</b>	Arduino Yun / simulated with EC2
2. <b>Connectivity technology</b>	MQTT, websocket
3. <b>scalable data processing architecture</b>	AWS IoT, Lambda, DynamoDB
4. <b>Parking data is made available</b>	API Gateway, SNS, S3
5. <b>security and protection</b>	IAM, Cognito, x.509 certificate authentication

## 3. MES&PLC Upgrade for a global automotive brand

2018 - 2019 Architecture, MCU, Firmware, AWS IoT

Digitalization upgrade its MES&PLC of 200+ workstations for an international auto company.

## 4. HiPot Test SCADA System for a global semiconductor brand

**2018 - 2019 Architecture, MCU, Firmware, AWS IoT**

Digitalization upgrade of an international semiconductor company for its HiPot Test System.

## 5. Vision Inspection & Finish Handler System, 50+ systems

**2019 - 2019 Architecture, MCU, Firmware, AWS IoT**

Digitalization upgrade of Vision Inspection and Finish Handler systems for a global semiconductor brand.

## 6. Atlas Copco Open Protocol C++ Implementation

**2018 - 2019 Architecture, MCU, Firmware, AWS IoT**

Support up to 200 devices to communicate in realtime, using Boost.Spirit & Boost.Asio TCP/IP sockets for a global semiconductor brand

## 7. SCPI & IEEE 488.2 C++ Implementation

**2017 - 2018 Architecture, MCU, Firmware, AWS IoT**

using Boost.Spirit & Boost.Asio Serial Port(RS232)

## 8. RT-Thread&Pandora IoT Board on Linux

**2019 - 2019 Architecture, MCU, Firmware, IoT, ARM, ST32**

Ported RT-Thread development on Pandora IoT Board STM32L475 to Linux, see<https://github.com/rockonedege/stm32l475-atk-pandora>  
(<https://github.com/rockonedege/stm32l475-atk-pandora>)

## 9. IIoT Semiconductor 2D Traceability System

**2020 - 2020 Architecture, MCU, Firmware, AWS IoT**

A semiconductor 2-Dimensional Barcode traceability system using Docker, MongoDB, Flask, Python, C++, Shell Scripts

## 10. IIoT Upgrade for Digital Oven Management System for Semiconductor

**2020 - 2020 Architecture, MCU, Firmware, AWS IoT**

A Digital Oven Management System for 100+ multi-brand equipments, using Docker, Flask, Python, C#, Pandas, Plotly, MongoDB

## 11. AGV&Milk Run Control Tower for Intelligent Factory

**2020 - 2020 Architecture, Vue, 3D, AGV, Milkrun**

An AGV&Milk-Run 3D Control Tower for an Industrie 4.0 Lighthouse Factory

# Husky Injection Molding Systems

## 1. Team&Department Startup

**2005 - 2007 Teams Startup**

Successively started up global an engineering team and then lift it as a global R&D department

Startup of Shanghai Machines Plant

**2005 - 2007 Teams Startup**

Awarded Special Contribution Certificate by the VP, Global Machines. Commuted between Canada&Shanghai, built the production system IT infrastructure and a software engineering team, contributed to the US\$ 1M profit ahead of schedule

- split office time in half between Shanghai and Toronto
  - set up controls software team
  - set up IT infrastructures, Polaris production systems, assembly & test processes.
  - integrated data&process with global factories in Canada and Luxemburg.
- contributed to the ahead-of-schedule break-even of \$1M profit

**Setup of Global Development Engineering Department(Software), Shanghai****2007 - 2007 Teams Startup**

A first&only offshore R&D department, quickly ramped up to 16 members, topped in loyalty, engagement, performance for 11+ years

All with Master of Science/Engineering degrees.

Team size grew from 5 to 16 gradually.

**Top-Ranked Team Leader****2011 - 2011 Teams Startup**

Result report picked by Kenexa into its shortlist for Husky CEO of a global employee survey

## 2. Non-Development Project Management

**2005 - 2016 leadership**

Team and people management, including global PLT, agile process etc.

**Global Product Launch Trip****2011 - 2016 Project Management, project management**

Global machines setup on customers' site for Coca cola, TetraPak etc. factories in China, Vietnam, Mexico, Australia etc.

It's been Husky's tradition to set up in-development machines at customers' sites directly by teams from R&D.

Shanghai team's participation expanded for China to global customer sites since 2011.

**Build Agile development and CI/CD process****2006 - 2008 Project Management**

Modernized software development by a forward leap of 10 years

Introduced and fostered Agile Software culture including

- SCRUM/TDD Development Process
- Toolchain

such as SCRUM/Test-Driven Development(TDD) , Microsoft Team Foundation Servers.

Multiple trip to Canada to process&toolchain evaluation such as IBM Rational Rose, McCabe Software and Team Foundation Server(TFS)

**Merging Moldflow's Manufacturing Solutions Unit****2008 - 2008 Project Management**

Took over software, hardware and customers from French&US teams and sustaining development for TetraPak, L'oreal etc.

- MES Productline:
  - Celltrack
    - Java, PostgreSQL
    - Data Collector
  - Shotscope
    - Delphi, C++, SQL AnyWhere
    - MCU embedded system
- SCADA Productline:
  - Matrix hotrunner controller
    - C++/QT, Linux/Fedora, MySQL
    - embedded computer

See [media reports \(http://www.manufacturing.net/news/2007/06/husky-buying-moldflows-manufacturing-solutions-unit\)](http://www.manufacturing.net/news/2007/06/husky-buying-moldflows-manufacturing-solutions-unit)

### 3. Product Architect&Development Management

#### 2005 - 2016 SCADA, MES, DevOps

various software development including SCADA, MES, DevOps, 3rd-party integration, value-added modules etc.

#### Altanium Hot Runner Realtime Controller

#### 2008 - 2009 Product Development, SCADA, C++, Linux, MySQL, SCADA, C++, QT, MySQL, EtherCAT, C#, QT, SQL Server, TWinCAT, IPC, Linux, MCU

An embeded real-time SCADA based on Linux(low end), Windows(high end), C++11, C#, MySQL, IPC

Hot runner Controller with advance temperature and motion control algorithm.

- Low-end series
  - MCU, Elo touch screen Linux/Fedora, QT/C++, MySQL.
- High-end series
  - Beckhoff IPC, TWinCAT, EtherCAT, C#, SQL Server

#### Realtime Injection Molding Control System

#### 2005 - 2016 Product Development, SCADA

A complex SCADA system with HMI, PLC, Robotics and Industrial IoT(IIoT) integration, semi-annual/quarterly release for 10+ years covering 10+ product lines and 1000+ machines shipped around the world running 7x24h

#### Polaris Injection Control SCADA System

A foundational software platform for all 10+ machine product lines, it's to Husky as much as Windows to Microsoft.

Features:

- embedded Windows image customization
- realtime servo/motion control
- robotic(Husky, ABB etc.), EOAT control
- PLC, PID Algorithm, thermostat control, field bus, industrial IoT
- HMI, machine operation, parameterization, diagnostics, Events&Alarming, History and OEE(Overall equipment effectiveness)
- MES, ERP, Edge computing & Cloud connectivity

Technologies:

- Beckhoff Industrial PC
- TwinCAT, CODESYS, PLC, IEC61131
- UART, RS232/485, Profibus, SERCOS etc. field buses
- EtherCAT etc. industrial ethernet
- OPC, OPC-UA etc. industrial internet
- RFID, Rubeer etc. wireless protocol
- C++, C#, Ruby, Python, VB6
- WPF, Silverlight, WCF, WinForm, ATL/COM, .NET, SQL Server
- Microsoft Robotics Studio, Visual Studio, Team Foundation System
- Microsoft Azure, IoT Hub

#### Injection Molding - PLC

#### 2005 - 2016 Product Development, SCADA

A TwinCAT/EtherCAT based PLC program for real-time control of servo drives and robotics etc.

PLC programming for real-time control with TWinCAT/CODESYS technologies for the

- injection unit
- clamp unit and
- robotics systems, sending status to and receiving commands from the HMI.

Algorithms were written in IEC61131/ST Language and modularized for max reuse.

All Polaris HMI&PLC features are wrapped into reusable and composable *modules* like LEGOs, Polaris Production System

- manages the 6K+ and growing modules and
- generates deployable packages(11K+),

one for each machine, be it physically real or simulation for development purposes), so that each machine is equipped with a unique software package of HMI and PLC combined.

## Injection Molding - Robots&3rd-party Integration

### 2005 - 2016 Product Development, SCADA

Integration of 3rd-party auxiliary systems(ABB Robots etc.) into Husky's HMI through GPIO, RS232, ethernet, OPC etc

- complete a whole production cycle/process
- facilitate the operators with a single one screen.

The task usually involved

- data transfer through standard or proprietary(e.g. RS-485, digital I/O etc.) protocols
- embedding screens into the Polaris HMI UI

examples included:

- ABB robots, 2006
- Resin Dryers from [Plastic Systems \(http://plasticsystems.com/\)](http://plasticsystems.com/)
- dehumidifier
- conveyers etc.

## Injection Molding - IoT&MES Integration

### 2005 - 2016 Product Development, SCADA, SCADA

Edge computing software development for collecting IoT info and relaying to clouds.

## HyMET Hotrunner Heat logger

### 2007 - 2007 Product Development, SCADA

A most beautiful software for real-time machine data acquisition and storage

A colleague from the HyMET team wrote to me saying:

this is by far the most beautiful software I've ever seen in Husky.

Lack of experience alloys injection, designers wished to monitor the heat parameters at the barrels and sprues in the long term. It was then decided to develop this tool to be deployed onto every sold HyMET machine, logging all interesting parameters. A husky service guy would visit the customer to copy the data out every 6 month or longer.

The project is to read from up to about 100 temperatures values from thermal couplers and save them into logs.

The critical requirements include:

- no threat to normal machines operations
  - no system resource contention with the Polaris HMI/PLC, i.e. minimum CPU cycles and memory footprints
  - no disk space running out
- non-stoppable by the factory operators
- up to 100 channels
- records at an interval from 1 sec to 180 seconds.
- flexible scheduling based on calendars and shifts
- intelligent start/stop of logging with on/off statuses of injection cycles
- data must be power-failure safe, i.e., no corrupt file due to accidental power loss
- tight schedule imposed by machine shipping date

I coached an intern to have developed the code to read temperatures through TwinCAT API, and personally wrote the UI in [WTL \(https://en.wikipedia.org/wiki/Windows\\_Template\\_Library\)](https://en.wikipedia.org/wiki/Windows_Template_Library) and the logging in [POCO \(https://pocoproject.org/\)](https://pocoproject.org/), I also contributed back the enhancement I made since it's an open source project.

## HMI Touchscreen Label Control

## 2006 - 2006 Product Development, C++, COM, MFC

A C++/COM control with adaptive multi-language display for touch-screen HMI

An in-place replacement of the built-in Visual Basic 6 Label control, adapted to HMI touch screens, solving the problem of string truncation during switching display languages, e.g. from English to German.

A companion utility was developed to insert 1K+ of usages into the whole source code tree (hundreds of Kloc) at one go.

## .NET HMI Architecture&Development

### 2006 - 2008 Product Development, SCADA

Architected a new HMI into layers and implemented with .NET/C#/WinForm/SQL Server

- 2 Polaris HMI platforms
  - COM/VB6 HMI platform for legacy generations of injection machines
  - re-Architecture and re-implementation of a .NET/C# HMI platform for later generations of injection machines
  - value-added modules, e.g.
    - development of Statistical Quality Control (SQC) and Statistical Process Control (SPC) module
  - hostlink, a proprietary communication protocol to talk to Husky machines remotely

A GUI/touch-screens enabled application running on Windows XP/7 Embedded with rich features enabling operators with an ultimate control

- from cycle breakdowns to production jobs schedules.

The UI is **configured** during package generation with the Production System per customer orders.

2 versions are maintained:

- Legacy version
  - started around 2000
  - default for all product lines roughly between circa 2000 and 2007
  - being slow and corrupt-prone for being resource demanding as a result of accumulative abuse of system resources over the years.
  - coded in VB6 using COM technology, and source controlled by SourceSafe
  - running on Windows XP Embedded
- Current version
  - started around 2007
  - default for all product lines starting from around 2008
  - coded in C# using .NET technology and modern Tools
    - A new architecture leveraging C#/.NET and with lessons(a lot of them!) from the previous version
      - fully OOP
      - decoupled layers for devices, domains and UI logics, with message-based communication
      - less dependent on TwinCAT opaque file formats
    - using new Application lifecycle management (ALM) tool and methodology
      - Microsoft Team Foundation System(TFS)
      - TDD/SCRUM process
  - running on Windows 7 Embedded

I was wholly involved since the inception of the architecture of the .NET version, such as

- early concepts selection
  - e.g. meeting at Canada the representatives from Microsoft, IBM and McCabe and voting for the final choice of TFS
- leading the year-long re-implementation of every VB6 screen in C#/.NET in Shanghai
- complete a whole production cycle/process
- facilitate the operators with a single one screen.

The task usually involved

- data transfer through standard or proprietary(e.g. RS-485, digital I/O etc.) protocols
- embedding screens into the Polaris HMI UI

examples included:

- ABB robots, 2006
- Resin Dryers from [Plastic Systems \(http://plasticsystems.com/\)](http://plasticsystems.com/)
- dehumidifier
- conveyers etc.

## Shotscope C/S SCADA&MES System

### 2008 - 2008 Product Development, MES

Based on MCU/PC Linux/Windows server, Delphi, Borland C++ and MySQL, Linux

When it bought Moldflow manufacturing division, Husky dismissed the original French&US teams and had all the assets(software&hardware) sent to Shanghai

Some of the code dated backed to early 1990s, and the technologies based on Delphi, Borland C++ and MySQL, Linux

My team successfully supported existing European/American customers with sustainment development until the next .net Browser/Server replacement , i.e. SSNX, came out.

## Celltrack C/S SCADA&MES System

### 2008 - 2008 Product Development, MES

Based on MCU/PC Linux/Windows server, Java, MySQL.

Some of the code dated backed to early 1990s, and the technologies based on Java, Sybase database, MySQL, Linux

My team successfully supported existing European/American customers with sustainment development until the next .net Browser/Server replacement , i.e. SSNX, came out.

## Shotscope NX B/S SCADA&MES System

### 2008 - 2015 Product Development, MES

A first web based B/S MES system featuring predictive maintenance, connected factory etc., a new generation of product replacing Celltrack and Shotscope

The successor to Shotscope and Celltrack, with modern IT technologies. To quote its [homepage](http://www.husky.co/EN-US/Shotscope-NX.aspx) (<http://www.husky.co/EN-US/Shotscope-NX.aspx>):

The industry's most advanced plant-wide process and production monitoring system

the industry's most advanced plant-wide process and production monitoring system with a yearly sales of \$3M

A first ever Husky made manufacturing monitoring system aimed to be an integration of 3 SCADA/ERP software products: SmartLink, Shotscope, CellTrack, embracing modern technologies at the time to achieve:

- Friendly and intuitive UI and better data presentation ability
- More distributed supervisory ability
- More data storage and processing power
- Built-in communication with other Husky software such as Polaris IMS Control system and Altanium Hot Runner Controller

The first release was delivered on time and received immediate orders at its debut at NPE show 2009, Chicago.

The majority of implementation was done in Shanghai under my supervision with a 3rd-party test team hired from Toronto for feedbacks and bugs reporting.

My team was a core party from the start and kept improving it until it was feature stable and transferred to After Market Service(AMS) engineering team for customer-specific changes in 2015.

Major Technologies include WCF, Silverlight, SQL Server, written in C#.

## HMI Internatinalization(i18n) Database Design

### 2006 - 2006 Product Development, DevOps

Designed a database and a set of access API in C++/COM for managing multi-language resource used by all Husky software, e.g. HMI, SSNX

With the growing number of supported languages from a mere 1, i.e. English, to over 22, the original single-file based solution suffered from

- too large file size up to 70+ MB
  - prone to corruption due to HMI crash or power loss
  - poor read/write performance
  - all-or-nothing rigidity, while only 2 languages(English and local) are requested in most cases.

By keeping its client interfaces intact, I personally

- re-designed the storage schema



- re-implemented the COM server with ATL/C++
- leveraged memory-mapped file to access the files

thus getting rid of all the aforementioned shortcomings, without a single line of change in the HMI code.

## 4. Product Development - IoT, IIoT, Industry 4.0 & Cloud

### 2005 - 2016 IoT, IIoT, Industry 4.0 & Cloud

various products leveraging industrial ethernet&internet, big data, cloud technologies etc.

产品架构&项目管理

### Molds & Tooling Lifetime IoT Solution

#### 2009 - 2009 Product Development, IoT, Industry 4.0 & Cloud

An cloud-ready system managing lifetimes of all Husky mold tooling plates , starting from orders, through design, production, sales and service. Data uploaded from and downloaded to tagged-things over the air (OTA) through RuBee(IEEE 1902.1)

嵌入式无线

- a CMS server
- workstation for Rubee tag programming/refurbishing
- HMI integration of Mold ID

Latest Husky injection molding systems are equipped with the mold identification(Mold ID) feature, enabled by [RuBee](https://en.wikipedia.org/wiki/RuBee) (<https://en.wikipedia.org/wiki/RuBee>) tags attached to mold tooling plates.

When a mold is installed, parameters are read from the tags over the air,

- enabling mold identification and quick alignment if matches, or
- reject installation if not

to avoid accidental damages.

My team built

- a web portal
  - global data center for all mold toolings, sold or under development
  - for daily cross-function collaboration.
    - mold designers to specify parameters
    - production engineers to program tags
    - service engineers to refurbish tags
- Integration into Polaris HMI
  - wireless detection of Rubee tags and parameter fetching
- Tag Programming/refurbishing Tool
  - writing of mold tooling parameters through RuBee(IEEE 1902.1)
  - compress data into the 256-byte memory space
  - enabling engineers to program over the intranet, e.g. writing a Shanghai tag from Bolton

SharePoint, Silverlight, WinForm and Socket programming were involved.

I oversaw the whole project as it was completely developed in Shanghai and deployed in Shanghai's IT servers for global access. The maintenance and response to requests raised through IT Help Desk are handle solely by the Shanghai team.

### Injection&Molding Industry-4.0 Integration - RT Communication

#### 2016 - 2016 Product Development, IoT, Industry 4.0 & Cloud

---



---

TwinCAT/EtherCAT based rea-time industrial communication between IPC controllers. Awarded Ringier Technology Innovation Award for Industry 4.0 level features

Made a great debut on K-Show 2016 and won [Ringier Technology Innovation Award](https://omnexus.specialchem.com/news/industry-news/husky-ringier-technology-innovation-award-000186265) (<https://omnexus.specialchem.com/news/industry-news/husky-ringier-technology-innovation-award-000186265>) later .

[Altanium controller](http://www.husky.co/EN-US/Altanium-Controllers-Overview.aspx) (<http://www.husky.co/EN-US/Altanium-Controllers-Overview.aspx>) has stayed an independent business from the machine systems since its purchase from Moldflow, based in US.

It was decided from the Horizon(a.k.a HyperSync) product line, the Altanium controller would be an integrated part of the injection system for branding promotion and cost benefits:

- *one single touch-screen panel* and integrated HMI instead of one for each.
- *premium Altanium feature/performance* when ordered together, starting from HyperSync.

which means a design of a new unified & proprietary interface

- hardware
  - one ethernet cable instead of various digital I/O, RS422 and other industrial interfaces
- software
  - design of a protocol
    - unifying all in-use commands for temperature and servo control commands
    - extensibility of future functionalities
    - identification of Husky equipments from either side
      - protection of Husky IP
      - protection of malice penetration from the network
  - remote desktop viewing
    - an VNC/RDP-like solution for an operator to control the Altanium controller from the machine

This was completely done by my team in shanghai

- started with a few slides of ppt from the Product Manager, mentioning *Altanium Integration*,
- collected requirements and consensus from all teams at stake and
- Concepting
- Design execution and validation

It was an unforgettable experience for all the challenges we had to cope with.

- The one and only Prototype machine built in Bolton
  - late nights of remote connection&debugging on Bolton's shopfloor till 4:00AM
  - several tight validation trips to and from Bolton
    - compete with other design teams for the availability of the prototype machine
- all teams except for software design are in North American
- had to borrow servo and motors from [Baumueller](http://www.baumueller.de/en) (<http://www.baumueller.de/en>) at the [ChinaPlas](http://www.chinaplasonline.com/CPS15/Home/lang-eng/Information.aspx) (<http://www.chinaplasonline.com/CPS15/Home/lang-eng/Information.aspx>)
- had to figure the Altanium side technologies by ourselves through code reading

Due to departmental priorities conflicts between US and Canada teams, we stayed in the middle of one party too demanding and one party too uninterested.

The features were successfully shipped to the first customer TetraPak at Mexicali, Mexico. One designer from my team was part of the PLT team to startup the system .

**It was quite an experience and I was proud that my team made it.**

## Injection&Molding Industry-4.0 Integration - Remote HMI

### 2016 - 2016 Product Development, IoT, Industry 4.0 & Cloud

Award-winning Industry 4.0 level features leveraging rea-time industrial internet technologies for the HyperSync machines.

Designed for Hypersync Machines, debuted on K-Show 2016 and won [Ringier Technology Innovation Award](https://omnexus.specialchem.com/news/industry-news/husky-ringier-technology-innovation-award-000186265) (<https://omnexus.specialchem.com/news/industry-news/husky-ringier-technology-innovation-award-000186265>) later .

[Altanium controller](http://www.husky.co/EN-US/Altanium-Controllers-Overview.aspx) (<http://www.husky.co/EN-US/Altanium-Controllers-Overview.aspx>) has stayed an independent business from the machine systems since its purchase from Moldflow, based in US.

It was decided from the Horizon(a.k.a HyperSync) product line, the Altanium controller would be an integrated part of the injection system for branding promotion and cost benefits:

- *one single touch-screen panel* and integrated HMI instead of one for each.
- *premium Altanium feature/performance* when ordered together, starting from HyperSync.

which means a design of a new unified & proprietary interface

- hardware
  - one ethernet cable instead of various digital I/O, RS422 and other industrial interfaces
- software
  - design of a protocol
    - unifying all in-use commands for temperature and servo control commands
    - extensibility of future functionalities

- identification of Husky equipments from either side
  - protection of Husky IP
  - protection of malice penetration from the network
- remote desktop viewing
  - an VNC/RDP-like solution for an operator to control the Altanium controller from the machine

This was completely done by my team in shanghai

- started with a few slides of ppt from the Product Manager, mentioning *Altanium Integration*,
- collected requirements and consensus from all teams at stake and
- Concepting
- Design execution and validation

It was an unforgettable experience for all the challenges we had to cope with.

- The one and only Prototype machine built in Bolton
  - late nights of remote connection&debugging on Bolton's shopfloor till 4:00AM
  - several tight validation trips to and from Bolton
    - compete with other design teams for the availability of the prototype machine
- all teams except for software design are in North American
- had to borrow servo and motors from [Baumueller](http://www.baumueller.de/en) (<http://www.baumueller.de/en>) at the [ChinaPlas](http://www.chinaplasonline.com/CPS15/Home/lang-eng/Information.aspx) (<http://www.chinaplasonline.com/CPS15/Home/lang-eng/Information.aspx>)
- had to figure the Altanium side technologies by ourselves through code reading

Due to departmental priorities conflicts between US and Canada teams, we stayed in the middle of one party too demanding and one party too uninterested.

The features were successfully shipped to the first customer TetraPak at Mexicali, Mexico. One designer from my team was part of the PLT team to startup the system .

**It was quite an experience and I was proud that my team made it.**

## Injection&Molding Industry-4.0 Integration - Intelligent Controller Pairing

### 2016 - 2016 industry 4.0

Fast pairing between injection & molding controllers based on Internet Control Message Protocol(ICMP) and async & parallelism algorithms

While integrating Altanium controllers into Horizon machines, one of the features was to detect the connection of a controller, and fetch its APIPA address, which became a challenge due to a maximum of delay was allowed.

The built-in .NET classes did not cut as the it resulted in **10s at best and 40s on average**, and cost about 700MB memory or higher till the host HMI application crashes.

Guessing that it might be the wrapping of .NET over the raw system sockets, I implemented the functionality with C++/Boost.Asio and leveraged asynchronous techniques, and was able to complete the detection within **2s** using around **5MB** memory.

The algorithm was then wrapped

- through C++/CLI as a .Net Assembly for consumption of Polaris HMI
- as a trouble-shooting utility application for testers on the shopfloor and service people in the field.

## SSNX on Azure IIoT System

### 2015 - 2016 Windows Azure

Re-designed the best-selling Shotscope NX MES product from on-premise(Windows Server/SQL Server) to cloud based(Windows Azure services), An injection molding focused Industrie 4.0 solution more comprehensive than products like GE Predix/Bosch PPM

## Hostlink Remote Machine Interface

### 2012 - 2012 Product Development, Value-Added Modules, C++, TCP, socket, Boost.ASIO, Boost.spirit

A C++11 (STL, Boost) implementation of Husky's proprietary inter-Machine Communication protocol Hostlink, a binary protocol enabling realtime access&control of Husky injection machines, based on raw TCP sockets, built with CMake

e.g.

- issuing commands
- reading statuses and cycle data

It has been adopted by some major customers to integrate Husky systems into their OPC/ERP systems, who requested more commands(e.g reading injection cycle breakdown data) to be supported, hence the project.

I personally wrote the whole code, based on the original MFC based code written in around 2000, as an illustration and teaching material of standard C++ for efficient, cross-platform system related code.

- added more advanced commands to fetch extended cycle data available in new generations of machines
- fixed bugs specific to 64-bit, which was not considered when the original code was written
- replaced ad-hoc command construction&parsing with Boost.Spirit and Boost.Qi
- replaced MFC based sockets communicated with Boost.ASIO
- supported platforms beyond Windows to Linux(tested Ubuntu)
- supported Python binding with Boost.Python
- supported .NET binding with C++/CLI

This project leverage quite a few [Boost](https://www.boost.org/) (<https://www.boost.org/>) libraries such as [Boost.Asio](https://www.boost.org/doc/libs/1_63_0/doc/html/boost_asio.html) ([https://www.boost.org/doc/libs/1\\_63\\_0/doc/html/boost\\_asio.html](https://www.boost.org/doc/libs/1_63_0/doc/html/boost_asio.html)), [Boost.Spirit](http://boost-spirit.com/home/) (<http://boost-spirit.com/home/>), [Boost.PP](https://www.boost.org/doc/libs/1_63_0/libs/preprocessor/doc/index.html) ([https://www.boost.org/doc/libs/1\\_63\\_0/libs/preprocessor/doc/index.html](https://www.boost.org/doc/libs/1_63_0/libs/preprocessor/doc/index.html)), [Boost.Python](https://www.boost.org/doc/libs/1_63_0/libs/python/doc/html/index.html) ([https://www.boost.org/doc/libs/1\\_63\\_0/libs/python/doc/html/index.html](https://www.boost.org/doc/libs/1_63_0/libs/python/doc/html/index.html)) and advanced modern C++ techniques such as metaprogramming.

Built with CMake.

## Machine Intelligence Algorithm Modules

### 2011 - 2011 Product Development, Value-Added Modules, Hands-on, 软件设计, 数据科学, C++, DirectX, 图形编程

realtime Statistical Quality Control (SQC) and Statistical Process Control(SPC) algorithms and visual charts with fine control, OEE.

Applied to cyclic injection parameters and best production control, e.g. 6-sigma.

## TwinCAT Realtime visualization microservices

### 2016 - 2016 PLC, motion control, IPC

---



---

Beckhoff TwinCAT 2 TSM information visualization in Javascript/CSS3/HTML5, also hosted as microservice/JSON APIs through a Rest Server in C++11(ATL/COM, STL, Boost, CMake), Python, JavaScript

A binary opaque format, named tsm, is used in TwinCAT 2 for configurations, which is where hardware *meets* software in the TwinCAT-based control system, specifically it maps software *variables* with Hardware I/O *signals*

- loads variables from software logic written in PLC/CNC/C++ etc.
- scans hardware I/O devices such as cards, boxes, terminals
- scheduling: exchanges values to-and-from in a timely manner(i.e. as tasks).

Reading directly from the tsm file is the most reliable way at runtime to extract the information for extension.

Husky's HMI has been structured in a way that's closely coupled to a local TwinCAT runtime, which has hindered its evolution in two ways

- engineering efficiency
  - even simple non-PLC HMI change requires to fire up the whole simulation environment in complicated VMWare image, which, in turn, makes it hard to bring in new bloods to the team, as the learning curve for such simple tasks is is very unusually steep and daunting for new engineers
- innovations for industry 4.0 To Quote Microsoft, this world is heading toward **cloud first** and **mobile first**. A universal set of UI that enables people to monitor, or operate with properly configured access rights is a must to be industry 4.0 compatible.
- product roadmap while it's been long desired to unify the platforms for the Machines and HotRunner controllers business to strengthen the one-Husky brand, duplicate development and features have to be paid and the customers have to operator on two separate HMI/Touch screens to get their job done.

I figured deciphering the tsm file and hosting it through a web service would be an ideal decoupling solution

- minimal risk
  - leaving the PLC algorithms intact largely.
- maximum extensibility of HMI
  - with modern UI frameworks, the HMI can be develop only once to fit all client platforms. declarative style UI can be developed any platform, mobile or desktop, deployable to local, on-premises servers or on cloud.
  - dependency on the PLC can be easily simulated with a simple JSON file in most cases.

I personally completed the proof of concept with some open source tool

- [websocketd](http://websocketd.com/) (<http://websocketd.com/>)
- [ADS, Beckhoff protocol to communicate with TwinCAT devices](https://github.com/rockonedege/ADS)(<https://github.com/rockonedege/ADS>) and in-house coding based on TwinCAT APIs.

C++, Python, JavaScript and some Go were used in the prototyping.

I did not have a chance to present the concepts and the work I've done before leaving Husky, Though planned to.

- added support of TwinCAT 2
- [link \(https://github.com/rockonedege/ADS\)](https://github.com/rockonedege/ADS)

## TwinCAT 2 TSM Viewer

### 2016 - 2016 C++, industrial ethernet

Added TwinCAT 2 support to the open source implementation of Beckhoff ADS protocol

## 5. Big Data & Digital Transformation - DevOps

### 2005 - 2016 DevOps

#### Whiteboard R&D Process BI Dashboard

### 2011 - 2016 Project Management

---



---

A REST microservice based ERP, CRM, R&D etc. Big Data collection & Business Intelligence(BI) application with C#/ASP.Net, Python/Django, and Dojo Toolkit, Angular.js.

Aggregated information from sources like

- Microsoft Team Foundation Server (TFS)
- Visual SourceSafe
- SharePoint
- ERP/BAAN
- Oracle HRMS
- UNC/mapped network drives

and assisted R&D Agile process with **machine learning** algorithms.

- Reatime&visual dashboard
  - projects progress
  - resources allocation and loading
- Boosting communication efficiency
  - daily standup meetings
  - weekly team meetings
  - milestone meetings
  - manager meetings

## Polaris Digital Production System For Injection Machines

### 2005 - 2016 Product Development, Digitalization

An in-house Digital Operation System supporting orders handling, development, assembly, final test&shipment lifetime management across Sales, R&D, Production, Service etc.

- daily used by engineers across departments(DevEng, Production and Service etc.) , such as
  - Translation System, supporting 22+ languages for Husky software
  - Mold ID System, managing RuBee(IEEE 1902.1) based tags mounted onto Husky made molds
  - ...
- **regular releases**(quarterly/semi-annually) of the aforementioned platforms for over a decade

The Production System assembles

- from 6k+ modules a package of the Platform System
- specific to the machine configuration ordered by customers, and
- delivers it to a target machine on the shopfloor for testing and shipping
- through network .

A loosely-coupled collection of utilities, including CLI, GUI and web sites, for

- **creation and versioning of tens of thousands of modules** of HMI and PLC features
  - source code fragments
  - pre-built binaries
  - customizable templates etc.
    - assembly of deployable **platform installation packages and deviation/retrofit patches** per injection molding machine serial number
    - bridging development, production and service engineers for service issues
    - improving engineering time performance

For its nature, dealing with the Product System is a part of everyone daily work, and fighting with it to make it work or workaround it stayed a main theme.

## Husky Software Internationalization(i18n) Management System

### 2006 - 2006 Product Development, DevOps

A centralized web-based application managing all language-related resources, such as phrases and its context, and automated translation management

The Polaris HMI supports 22+ languages. For any phrase to be added to the screens, it's required to determine whether new translation is needed early on, and if so, among others

- applications should be filed for approval from team leaders for payment to a 3rd-part translation company
- contexts of phrase usage, e.g. screenshots, should be provided to assist translation
- only selected languages should be packed and deployed onto the machine

A fresh college graduate, under my coach, developed a website using ASP.NET and SQL Server to have the goal achieved.

## DevOps Console - Console3

### 2012 - 2016 C++, WTL, STL

A commandline console emulator tailored for Husky DevOps, based on Console2, an open source tool

- fixed bugs
- replaced rendering GDI from to [Direct2D][#]/DirectWrite ([https://msdn.microsoft.com/zh-cn/library/windows/desktop/dd368038\(v=vs.85\).aspx](https://msdn.microsoft.com/zh-cn/library/windows/desktop/dd368038(v=vs.85).aspx))
- added Husky-specific feature with integration to Husky tools

This tool was widely used by the Shanghai team.

## Zebra search Engine

### 2009 - 2016 machine learning, algorithm

---



---

A fast search engine for Husky jargons & documentation, written in C++ for Windows&Linux

This native version inspired Bianque Hysearch, a cloud based solution.

## 6.

## Bianque Digital Transformation - A Industry 4.0 System

### 2013 - 2016 IoT Hub, REST, microservice

---

A cloud-based Digital Operation System supporting orders handling, development, assembly, final test&shipment lifetime management across Sales, R&D, Production, Service leveraging virtualization & Cloud technologies, e.g. Microsoft Azure, IoT Hub, REST, micro service, and mobile apps

The Polaris Production System dated back to circa 2000 and evolved into a daunting complexity with a very steep learning curve over the years, and a daily drain of engineering efficiency, To name a few,

- fail frequently
  - unmanaged and undocumented heavy dependency on assumed fixed paths which are not
- poor errors mechanism
  - pop-up dialogues in the middle of automated processes, which invalidated the automation itself.
- convoluted system configurations

70-100+GB decade-old VMWare image one can hardly create from the scratch.

- overuse of large VMWare images file, a.k.a under use of virtualization technologies

every engineer has **a copy for each task**. It's normal to have 2-3 such images running at the same time to get work done.

Named after the ancient Chinese doctor, Bianque was meant to fix it with a it-just-works solution with modern DevOps and virtualization technologies by distinguishing the requirements between

- Development Engineering, which creates software packages

introduced Vagrant for virtual environment management

- Production Engineering, which consumes software packages

replaced the legacy VB6 desktop application with a single page application(SPA) site backed up with REST web services

- cloud(Microsoft Azure) based
  - removing the waste of gigantic duplicate VMWare image files(which IT complains all the time) with efficient cloud solutions
  - accessible from multiple platform
  - accessible from anywhere to benefit field service
  - easy upgrades
- service-oriented with APIs
  - extensible for specific needs

Bianque largely consisted of

## Bianque Core Microservice Platform - Lancelot

**2013 - 2016 REST, Product Development, IoT, Industry 4.0 & Cloud**

A collection of REST Service, mostly in Python, including wrapping dependency to ERP, SQL Server, MongoDB, Redis, SharePoint, COM Server, to name a few.

Core algorithm library of Bianque.

## Bianque DevOps Platform - Acupuncture

**2013 - 2016 REST, Digital Product Development, IoT, Industry 4.0 & Cloud**

An HTML5 Front End of Bianque. A single page application(SPA) using Django/Dojo Toolkit/Web Socket, supporting mobile platform such as Android/iPhone tablets and phones.

- **Acupuncture**
- **Acupuncture on mobile devices**

## Bianque Elastic Computing Platform - Transformer

**2013 - 2016 REST, Product Development, IoT, Industry 4.0 & Cloud**

Bianque's backend, consisting of a farm of VMWare images/Docker images, leveraging microservice, messaging technologies and Redis/MongoDB

- dispatch requests among images to maximized resource utilization
- concurrently execution of the Polaris Production System commands to have reduce from 40m to 5m-10m typically
- exposing Polaris Production System commands as REST services

## Bianque Engine - HySearch

**2013 - 2016 REST, Product Development, Machine Learning, AI, Python, MongoDB, Redis, Django, Bootstrap, Industry 4.0 & Cloud**

A machine learning powered search engine in Python/MongoDB/Redis/Django

An engine with a web portal providing sophisticated recommendation, results refining leveraging machine learning algorithms like naïve Bayes classifier, aggregating information from CRM, ERP, Sharepoint, TFS for R&D documents, order, customer, service ticket etc.

## 7. Open Source Software

### 2005 - 2016 Teams Startup

OSS contributions

#### Boost.Chrono C++ Library

### 2009 - 2010 Teams Startup

Helped design and review Boost.Chrono, a set of time facilities which was adopted into C++11 STL later.

Helped design and review Boost.Chrono, which was adopted into C++11 STL later; improved Spirit.X3 document and examples.

#### The POCO C++ Libraries

### 2008 - 2008 Teams Startup

Contributed code about the XML and logging handling

## Dediprogram Inc.

### 1. SF100 Flash Programmer

#### 2005 - 2006 Architecture, USB, MCU, C++, MFC, STL, Firmware, Protel, Altium

A fast SPI memory programmer supporting R&D and shopfloor usage

- support usages in R&D lab(GUI) and shopfloor gang programming(CLI)
- Outstanding software&hardware architecture
  - dual core ST7 MCU system, connected through USB to PC/Windows
  - PC GUI/CLI with C++ (MFC, STL, Boost)
  - MCU Firmware with C, Assembly
  - stable code base with exponential growth of chip portfolio till today
- quick market acceptance, e.g Intel R&D center at Israel
  - much faster programming speed
  - friendly UX
  - efficient service response

### 2. WinUSB++, USB driver in C++

#### 2008 - 2008 USB, MCU, C++

USB port control library, An OOP wrapper of WinUSB in C++

A Windows user mode USB driver in C++

## STMicroelectronics Inc.

### 1. Embedded IC Verification & Reference Design Kit(PC BIOS Memory)

#### 2003 - 2004 C, PCB, Protel, Altium, MCU

PCB design&firmware driver in C language for R&D IC verification and customers' design reference.

Based on ST's own MCU(uPSD3300), Covering all productlines of M50xx Firmware Hub(FWH), Low Pin Count(LPC) flash memories from KBs to MBs.

- Used ST's own MCU(uPSD3300) CPU
- PCBs with Altium



- code and application notes published on ST's website for freely download

e.g.

- showcased driver code with an application demos rotating quotes from *Albert Camus* on a LCD display, of which my favorite one was

Don't walk in front of me;

I may not follow. Don't walk behind me; I may not lead. Just walk beside me and be my friend.

- authored&published corresponding application notes, in English, elaborating the design.

## 2. Embedded IC Verification & Reference Design Kit(SPI/I<sup>2</sup>C Flash/EEPROM Memory)

**2004 - 2005 C, PCB, Protel, Altium, MCU, ARM**

MCU based PCB&firmware reference design for IC verification and application, mainly in portable C language

PCB&firmware driver for IC verification and application references of SPI Flash/EEPROM chips, covering M25Pxx, M45PExx etc. code, data and parameter storage flash memories from KBs to MBs.

- Used ST's own MCU(uPSD3300) and ARM(ST7) CPUs
- PCBs with Altium
- code and application notes published on ST's website for freely download

Demos including embedded OS booting e.g. ucOS/II and Linux, and Point of Sale(POS) machine reference design

## 3. Egret Flash Memory Programmer

**2003 - 2005 C++, USB, MCU, PCB**

---



---

A PC based SW/HW kit for NAND/NOR flash memory verification and firmware image burning through USB, Serial Port(UART) and Parallel Port(LPT) and then SPI, I<sup>2</sup>C ports.

In line with Company's effort to provide more accessible e-tools for the mass market and shipped with

- a Windows-based GUI software
- an ST7-based embedded gadget, connected through the parallel port (LTP) or USB.

This kit was to facilitate the usage of ST memory products, supporting SPI, I2C Initially.

Specifically,

- verification of page/sector read, program and erase commands
- free in-production usage of chip programming in small volumes(e.g. R&D designers)

Starting half-done & abandoned VB6 code base, created the best tool in class, which was licensed to a startup [Dediprog Inc.](http://www.dediprog.com/) (<http://www.dediprog.com/>), branded as SF1xx series.

## 4. RTOS Porting - uCLinux

**2005 - 2005 Linux, boot, assembly, C**

H/W&S/W kit illustrating booting embedded OS from M25Pxx SPI data/Code Storage flash using ARM7 ST7 CPU(STR71x), Assembly and C with RealView Development Suite/Keil

## 5. RTOS Porting - uc/OS-II

**2005 - 2005 Linux, boot, assembly, C**

H/W&S/W kit illustrating booting embedded RTOS from M25Pxx SPI data/Code Storage flash using ARM7 ST7 CPU(STR71x), Assembly and C with RealView Development Suite/Keil

## 6. Anatidae Project Tracking&Syncing System

2004 - 2004 C++

A progress tracking tool synchronizing projects between Shanghai, Czech, Italy and France, written in C++, MFC, STL, BOOST, XML

Initiated the project and completed it alone.

- About 50Kloc of C++ with MFC / STL / BOOST / ADO+.

## 7. Oscilloscope Simulator in C++

2003 - 2003 C++

An oscilloscope simulator drawing clocks&waveforms in C++ with WTL, STL, Boost

Drawing oscillograph based on chip I/O, e.g UART, SPI, I<sup>2</sup>C etc.

# Schneider Electric

## 1. Automatic Transfer Switching Equipment(ATS)

2002 - 2003 PLC, NPD

PLC and circuit breakers based system guaranteeing reliable, seamless power when outages occur

Essential for emergency power in buildings or other circumstances. ATS can automatically transfer load between the main power and the emergency power (generating set)without the operator

- Achieved all design goals, lead a team from R&D Center, Marketing, Field Application,
  - more intelligence
  - friendlier HMI
  - higher reliability
  - lower cost
- Using all-Schneider products
  - NEZA PLC
  - NS circuit breakers for reliability
  - safety with interlocks
    - software/electrical interlock with PLC
    - built-in mechanical interlock with NS
- Completed investment report and submitted for Chinese Management Committee(CMC)
  - 3 iterations of concept-proof prototypes
  - cost, saving 2/3 cost of solution on sale
  - technical risks & qualification plan
- Auto-Transfer System(ATS), an automatic switches on/off between two independent power supplies in no time in case of emergency for sectors like hospitals
  - based on NS Compact circuit breakers and NEZA PLC.
  - from the scratch to 3 full-functional prototypes
  - 1/3 cost saving from existing market solutions
  - submission to Chinese Management Committee(CMC) for final production investment evaluation

## 2. Air Circuit Breaker Localization Design

2002 - 2003 PCB, Protel

Localization design of VC65, DPN Vigi ASICs, PCB and Schematic

- VC65
- DPN Vigi

## 3.

# Development of XB2 industrial-strength LED alarms

## 2002 - 2003 Schematic, Protel

Development of industrial-strength LED alarms

research in preparation for launch Schneider branded alarms for factory shopfloors.

- investigate LED technologies.
- collect LED samples from select vendors
- devise tests for IEC conformance tests regarding lumen, product life etc.

## 4. Lab thermostat Design

### 2002 - 2003 MCU, PID Algorithm

An aging test device (up to 75°C x 1000 Hours) using PID algorithm and MCU

A lab device designed for breakers, contactors and PLC aging test for IEC&GB compliance.

- MCU PCB design with Protel
- PID algorithm
- mechanical design

## 5. Switching Power Supplies Localization

### 2002 - 2003 Schematic, PCB, Electrical Engineering

Electrical test design&reports of localizing switching power supplies originated from Poland

Electrical design to measure&and test various brands of switching powers on the market and Schneider's own, including reports on

- compliance with China Compulsory Certificates (CCC)
- technical performance / reliability / cost reports with competing local brands.

## 6. Lab Data Manager - LabUtils

### 2002 - 2003 Visual Basic, VBA

managing R&D test data, including machine learning, statistics algorithms and database, written in VB6, VBA & C++

Electrical design to measure&and test various brands of switching powers on the market and Schneider's own, including reports on

- compliance with China Compulsory Certificates (CCC)
- technical performance / reliability / cost reports with competing local brands.

# Harbin Institute of Technology

## 1. Researches On Digital Industrial Control&Automation

### 1998 - 2002 Industrial PC(IPC), RTOS, MCU

R&D on industrial control and automation, focusing on digital computer systems(IPC&Embedded) and fieldbus(Profibus, CAN, DeviceNet etc.), in HIT Railway Apparatus Research Center & HIT Rockwell Automation Center

- Research on industrial control and automation system
  - Electrical machines and Apparatus
    - transformers, generators and (servo) motors(Rockwell Automation brands)
    - relays, contactors and circuit breakers
  - Computer-based control&automation systems
    - Direct Digital Control System
    - Distributed Control System (DCS)
    - Fieldbus Control system (FCS)
  - real-time control & measurement methodology and reliability, e.g. FMEA
  - real-time embedded system (RTOS) and embedded application development
  - Research on electrical/electronic interface/bus protocols(ISA, PCI etc.)
- Development of control & automation solutions

- based on industrial PCs(IPC) and/or MCUs(Intel MCS-51 family)
- HMI development in C++/Visual Basic
- device driver development in Assembly/C/C++
- Electrical schematic&PCB design with Protel, and simulation with pSPICE/MATLAB/SIMULINK
- mechanical design(cam wheel based vibration platforms) with AutoCAD

**Most notably, 10+ sets of the following systems I developed were sold, having generated sales of over 1 million RMB.**

- The Electronic Ballast Test Platform
- The Low voltage Apparatus Automatic Maintenance Platform

## 2.

### Automatic Test Equipment for Relays, Contactors & Circuit Breakers

**2000 - 2001 PCB, Schematic, C++, Visual Basic 6, MATLAB**



An award-winning Industrial PC-based (IPC)&MCU automation system for the Ministry of Railways. ISA/PCI-based I/O and A/D PCB design, Inverter and motor control.

I took full charge of and design from scratch, including hardware, software and some mechanical design.

The objective of the project was to provide an automatic and all-in-one platform to cover the routine maintenance check of all low voltage apparatuses used on the train:

- circuit breakers
- contactors, and
- thermal relays

The checklist included

- electrical performance, e.g.
  - contact resistance
  - make/break time
  - over-voltage
  - under-voltage
  - over-current
  - short circuit etc.
- mechanical performance
  - sinusoidal vibration.
- Hardware design
  - ISA board for digital I/Os
  - PCI board for data sampling and A/D conversion
  - Timer/counter circuit design
  - digital-coded power supply ranging between 1A-1300A or 0-440VAC/220VDC
  - Precise resistant load (resolution: 1/512 ohm)
  - High current sample and switch circuit.
- Software development
  - Motor control & Hardware driving
  - Precise voltage/current control algorithm
  - Complete and user-friendly performance parameter display
  - Failure report and Lifetime estimation(FMEA)
  - Test data management

#### Achievements:

- Awarded 1st Prize of technical innovations by Provincial Bureau of Education, 12/2002
- Awarded 2st Prize of technical innovations by Provincial Bureau of Science & Technology in 05/2003
- An academic paper was published on *the 8th International Low Voltage Apparatus Reliability Conference*

### 3. Automatic Test Equipment for Electronic Ballast

1999 - 1999 PCB, Schematic, C++, Visual Basic, AutoCAD

An Industrial PC-based (IPC)&MCU automation system for the Ministry of Railways. ISA/PCI-based I/O and A/D PCB design. C++, VB6, power harmonic distortion analysis (FFT)

Objective

- up to 8 electronic ballasts (15W, 20W, 30W, 40W) concurrently,
- detailed performance parameters such as
  - voltage,
  - current,
  - power factor,
  - harmonic distortions(THD),
  - temperature rise etc.
  - real-time voltage and current waveform
  - FMEA reports
  - historic events database
  - pretty print
- hardware
  - ISA I/O board, A/D board
  - Power supply board (DC 47V-57V, AC 200V - 240V)
  - Sensor board (voltage, current and temperature)
  - Target-switching board
- software
  - Hardware driving
  - Real-time sample and analysis (FFT algorithm)
  - Rendering visual waveform
  - Test data management

This platform greatly sped up the routine check efforts of technicians who were dealing with tens of electronic ballasts daily. The usages were spread to another 5 railway bureaus soon after the completion for its satisfactory performance.

A paper was published on a national academic periodical explain the design.

It was all of my work from initial research, hardware design, software design spec and debug, except for the software UI design.

## Sparetime/Open Source

### 1. A Literal Converter for Integers

2004 - 2004 C++

A small utility to convert between decimal, hex and octal integer literals

for myself to help process the data file from the oscilloscope, which I used to record data going through Serial Peripheral Interface(SPI) ports while verifying the flash memory chips.

- Open sourced with an article on [The CodeProject](https://www.codeproject.com/Articles/10379/A-Literal-Converter-for-Integers) (<https://www.codeproject.com/Articles/10379/A-Literal-Converter-for-Integers>)
- Written in C++
- [link](https://www.codeproject.com/Articles/10379/A-Literal-Converter-for-Integers) (<https://www.codeproject.com/Articles/10379/A-Literal-Converter-for-Integers>)

### 2. Duplicate File Finder

2008 - 2008 C++

Find duplicated files in C++11(WTL, STL, Boost, CMake)

### 3. 2D Water Effect in WTL

2011 - 2011 C++, Image, Graphics, Direct2D, DirectWrite

---

An open source code and article using image processing and graphics rendering.

- published on [The Codeproject \(https://www.codeproject.com/Articles/188236/D-Water-Effect-in-WTL\)](https://www.codeproject.com/Articles/188236/D-Water-Effect-in-WTL).
  - written in C++/WTL
  - [#2DWTL \(https://www.codeproject.com/Articles/188236/D-Water-Effect-in-WTL\)](https://www.codeproject.com/Articles/188236/D-Water-Effect-in-WTL)
- 

This page is generated from Markdown text with Python, PDF with wkhtmltopdf and MS WORD with Pandoc. Last updated at 2024-09-14.