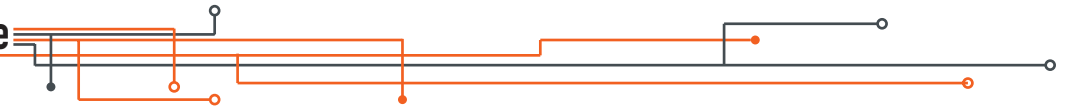




# IN TIME TEC

Creating Abundance



## Center of Excellence at In Time Tec Embedded Systems Software

Limited CPU power and memory, quick turnaround time, and minimal margin for error characterizes the constraints within which one is expected to develop embedded systems software. Compared to what is generally termed as the application domain, software development in embedded systems requires a strong fundamental understanding of the processor cycles, memory allocations, operating systems internals, etc.

Engineering talent at In Time Tec has extensive experience in the area of firmware development in imaging and printing domains. We have developed software for our customers across different families of microcontrollers and operating systems.

### 1. Secure Printing Solution on Custom Hardware

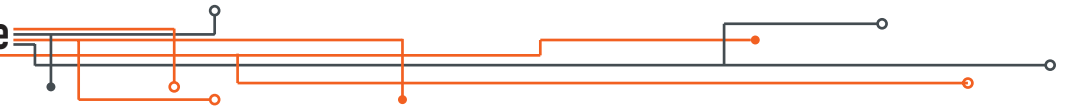
Hardware: 32-bit Atmel Microcontroller, 2-Port Network Switch, 1 USB Port  
Software: FreeRTOS Operating System, uIP TCP / IP Stack, USB Protocol  
Programming Language: C

In Time Tec developed a secure printing solution that involves reading proximity card data over a USB connected reader, user authentication with server over network, and release of print jobs. The hardware connects to a printer and releases the print jobs only after successful user authentication.

### 2. Secure Printing Solution Using Vendor-Provided SDK

Hardware: Intel Microcontroller  
Software: Embedded Linux, Windows CE and other RTOS  
Programming Language: C++, Java, C#

Printer vendors like HP and RICOH provide embedded SDK providing access to the lower level hardware. In Time Tec has experience developing secure print solutions that install into and run as part of the printer firmware.



### 3. Device Driver Development

Hardware: Intel Microcontroller

Software: Embedded Linux

Programming Language: C++

Designed and developed driver for USB based proximity readers. Control and Interrupt transfers were implemented.

### 4. Networking Stack

Hardware: 32-bit Atmel Microcontroller, 2-Port Network Switch

Software: FreeRTOS Operating System, uIP TCP / IP Stack

Programming Language: C

We programmed the 2-port network switch to implement packet sniffing and disabling/enabling one of the ports. As part of secure printing solution, a customer requirement was to sniff and scan the packets meant for printers and allow them only after authorization. We provided experience, working on the Address Resolution Protocol (ARP) and Dynamic Host Configuration Protocol (DHCP) implementations in the uIP stack.

### 5. Smart Card Protocol

Hardware: Intel X86 and ARM Microprocessor

Software: Windows CE / Windows OS, Linux OS

Programming Language: C / C++

Protocols/Standards: ISO / IEC 7816-3/4/8/9/15, PKCS#15, PKCS#11, CCID

We developed the secure printing solution requiring authentication through contact/contactless smartcard, i.e. Smart Card Logon, Digital Signature, and Email & Print Encryption. The Smart Card solution is integrated to work seamlessly with Kerberos - Network Authentication Protocol. All of the implementations run embedded along with the printer firmware.