

DICE III - TCD30xx FIRMWARE DEVELOPMENT

Firmware Developers create device applications for products based on the DICE III family with the TCAT Firmware Development Environment (FDE) and the comprehensive firmware source code tree (SDK).

The TCAT FDE is available as a downloadable archive, which is a self-contained Linux Virtual Machine (VM). The VM can be used on the developer's host operating system of choice.

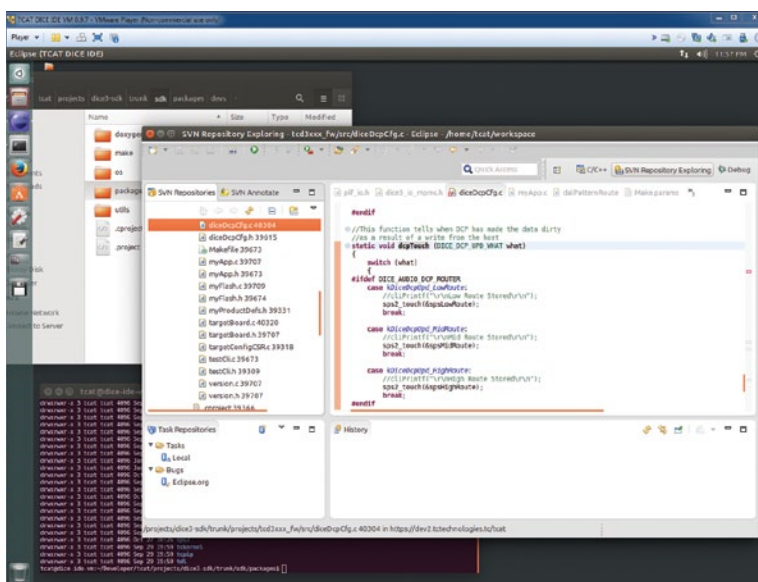
The SDK is available via SCM and contains all of the sources necessary to build device firmware applications. Sample applications are included for TCAT TCD30xx Evaluation Modules and can be used as a basis for customer device applications.

TCAT also provides branded drivers, host applications such as Control Panels where appropriate, and other utilities for OS X and Windows, at no extra cost.

TCAT FDE Features

Pre-configured Linux Virtual Machine

- ▶ Ubuntu LTS Release
- ▶ Version-stabilized ARM toolchain
- ▶ GNU ARM EABI
- ▶ Eclipse IDE, CTD
- ▶ eCos configuration tools
- ▶ Scripting tools (Python, etc.)
- ▶ SCM Command Line tools (Subversion, Git, Mercurial)
- ▶ Doxygen
- ▶ No per-seat developer fees, royalty-free



- ▶ Streamlined, developer-oriented Ubuntu distribution
- ▶ Flexible host computer options: Windows, OS X, Linux
- ▶ Install your VM player, open the FDE Virtual Machine and go
- ▶ Minimal developer configuration required
- ▶ IDE and firmware ready for Segger J-Link debug probes
- ▶ GNU Tools running on Linux provide lightning-fast build times
- ▶ eCos configuration tools and hardware interface
- ▶ Familiar edit, build and debug environment
- ▶ Well-established Eclipse support community

FIRMWARE SDK

The SDK is provided from TCAT's Subversion repository. The core SDK implements all peripheral drivers, protocol stacks, streaming state engines, RTOS scheduling, flash management, memory management, etc. that may be required for your applications.

Customer Applications are created by overriding standard functionality. You need only implement the parts that make your device unique. Streaming configurations, routing, mixing, CODECs, descriptors, branding, etc. are added using logical, well-documented APIs. Core modules are configurable using standard eCos CDL methodology.

- ▶ Languages: C, C++, asm, Python
- ▶ eCos RTOS
- ▶ DICE I/O peripheral drivers, implemented in the eCos HAL where appropriate
- ▶ Support for Segger J-Link debug probes integrated into project settings (JTAG Interface hardware not included)
- ▶ Flash file management
- ▶ Firmware update protocols
- ▶ Ethernet Network Stack
- ▶ Serial, USB, Telnet CLI options
- ▶ USB Stack and Audio Control
- ▶ AVB Stack and Audio Control
- ▶ IEEE1394 Stack and Audio Control
- ▶ Various project templates can be used with TCAT EVM's and as a starting point for your devices.

