

Porting Connected and Automated Vehicle Controller Software Across ATC Platforms

Vincent Mayeda, Econolite

Econolite, United States of America

Abstract

As connected and automated vehicle (CAV) technologies rapidly march ahead, the often-overlooked part of the CAV equation is how to cost effectively incorporate and make these technologies functional and practical in the existing infrastructure. Despite the advancements and capabilities of CAV on-board vehicle and roadside equipment, there are two very substantial challenges the industry faces to make CAV a reality. These technologies are rendered mostly unusable if: the fundamental part of intersection management – the controller – is not CAV-ready; or if the transportation agency has to replace all of its installed base of traffic controllers for CAV-ready controllers. There are thousands of 2070 Advanced Transportation Controller (ATC) controllers currently in use. For many agencies and cities, replacing their fleet of controllers all at once is cost prohibitive. This presentation will detail the work and progress of porting ATC controller software across hardware platforms to make existing installed traffic controllers CAV-ready.

Keywords:

ATC Controller, 2070, ATC Software, CAV

Cross Platform Porting of CAV-Ready ATC Software

As expected, a number of issues were encountered while porting the EOS ATC software system to different controller manufacturers and platforms. The software porting focused on the 2070-1C platforms based on e300 PowerPC cores since these were part of the Michigan Department of Transportation (MDOT) installed base of traffic signal controllers. For the purposes of this examination, the primary issues addressed in porting CAV-ready software to other 2070 controller platforms are related to: 1) controller card, 2) Linux O/S version, and 3) driver version. As all of the other manufacturers' device drivers are bound to specific O/S versions, the data space and porting processes addressed are reduced to two dimensions: controller and O/S version. This examination takes a close look at the functional differences between controller platforms and makers, and how the differences were resolved to successfully port the CAV-ready software.

Different Controller Platforms

For the basis of this paper, the software porting work centers around Design Acceptance Testing (DAT), which was run on third-party traffic controllers in order to identify areas where the controllers differ from the Econolite controllers. The DAT tests were then modified to function correctly on the test controllers. This work was a joint effort with Michigan Department of Transportation (MDOT) to update an existing fleet of 2070 ATC traffic controllers from several different manufacturers and platforms with new CAV-ready software.

Controller Pairings

McCain M6977 rev B	Linux 2.6.9
Intelight ASI-06-18182-B1 [MPC885 CPU model]	Linux 2.6.34
Intelight ASI-06-18183-B2 [MPC885 CPU]	Linux 3.4.33+
Intelight ASI-06-18182-B1 [MPC885 CPU]	Linux 3.14
Intelight ASI-06-18183-B2 [MPC8248 CPU]	Linux 3.4.33+
Intelight ASI-06-18183-B2 [MPC8248 CPU]	Linux 3.14
Pairings Addressed: Not Tested	
Intelight ASI-06-18182-B1 [MPC885 CPU]	Linux 3.14
Intelight ASI-06-18183-B2 [MPC8248 CPU]	Linux 3.14

There are planned upgrades for the Intelight controllers with the goal to reduce the total variations associated with the platform to as few as two versions. At the time of this paper, none of the Intelight controllers were upgraded due to a) risk of failure normally associated with a full system upgrade; and b) instructions are still being created for the “1882-based product, which may not support USB hotplug.” In (b), the instructions being created refer to the older MPC885 systems that use a DataKey device for removable storage instead of a USB memory drive device.

General Controller Differences

For the basis of this paper, the software porting work centers around DAT tests, which were run on third-party traffic controllers in order to identify areas where the controllers differ from the Econolite controllers. The DAT tests were then modified to function correctly on the test controllers.

Functional differences between the controllers were relatively minimal, although critical enough to cause EOS failure until resolved. A number of the differences caused failure of the DAT, however the capabilities involved are not used in EOS as it currently stands.

Several failures were observed in EOS, however these relate back to the DAT tests, and are discussed in the related section.

DAT Testing

Each DAT test is designed to verify compliance with the ATC Standards 5.2 specification. In some cases, ambiguities in the specification were found, rather than non-compliance.

OnLed – Front Panel Activity LED

The light emitting diode (LED) control values are documented in the ATC Specifications 5.2b Section B and ATC 6.25 Section B examples.

It was observed that the device driver [node /dev/cpuactive] expects character control values:

Character Control Values

0x00	NUL	turn LED off
0x01	SOH	turn LED on

Presence of Functional Data-Key

The data key presence bit is discussed in both the FIO and other sections of the ATC Specifications. Some interesting references are the ATC 5.2b and 6.25 Sections 8.2.1.7.1.3, 8.2.1.12 and B.2. In the reference sections the bit returned from FIO interrogation is set zero (0) for data key present, and one (1) for data key absent or when an error occurs. In the Section B.2 example, a non-blocking read from the device driver [/dev/datakeypresent] is shown, but no meaning is attached to the value returned. Thus, we must conclude that the availability of blocking read and the meaning of the return value are both unspecified.

The EOS software does not use blocking read, even though the DAT tests for it.

The controllers support detection of the data key as follows:

Econolite

0 –	absent or error
1 –	present

Blocking and non-blocking I/O are available

Intelight

0 –	present
1 –	absent or error

Only non-blocking I/O is available

McCain

0 –	absent or error
1 –	present

Only non-blocking I/O is available

Real-Time Clock and Line Synchronization

For all ATC controllers, the time-of-day clock is supposed to be synchronized with the power-line frequency, and should be capable of generating ‘tick’ signals as expected from the device driver [/dev/tod]. In all cases, it was determined and confirmed that the controllers tested passed the real-time tests in the DAT. There is a requirement in the ATC Specification 6.25, Section B.7 for the /dev/tod device node. This requirement is not there in ATC 5.2b.

The controllers support the clock requirements as follows:

- Econolite
 - /dev/tod – present
 - ‘tick’ signal – available from /dev/tod
- Intelight, Linux 2.6.34 and 3.4.33
 - /dev/tod – absent
 - ‘tick’ signal – should be available from CLOCK_MONOTONIC with Posix
- Intelight, Linux 3.14
 - /dev/tod – status is unknown
 - ‘tick’ signal – should be available from CLOCK_MONOTONIC



- ‘tick’ signal – may be available if /dev/tod is present
- McCain
 - /dev/tod – present
 - ‘tick’ signal – available from /dev/tod

Front Panel LCD Operation

The front panel of a 2070 crate is designed to behave as a serial terminal. Access to the panel is performed using a serial port device [/dev/sp6]. This test verifies the ability to send commands, read responses, and present reasonably legible output to the user. As the front panel is independent of the controller card, this test illuminates port handling differences between the controllers. In all cases the tested commands and displays functioned correctly. The one notable difference is that the Intelight MPC8248 controller running Linux 3.4.33 would not correctly receive responses from the front panel until the “read delay” was increased by approximately 50 percent, from 120 msec to 180 msec, in the DAT test. Whether the root cause is CPU speed, UART performance, port configuration, clock granularity, O/S efficiency, O/S thread dispatch, or otherwise, is unknown.

Controllers:

- Intelight MPC885, Linux 2.6.34
 - Pass
- Intelight MPC8284, Linux 3.4.33
 - Requires extended delay after front-panel query from CPU
- Intelight, other
 - To Be Determined
- McCain
 - Pass

Check System Memory and Performance

The purpose of this test is to verify memory availability and performance. All systems that were tested passed this test. It is presumed that the Intelight systems running Linux 3.14 will also pass.



Check Field I/O Performance

Field I/O behavior is described in detail in the ATC Specifications. The Field I/O tests perform raw, first-fail analysis of the synchronous I/O using loop-back through 2070-2A, 20709-8E, or 2070-8E+ interface cards. The tests involve configuration, read-back, and loop-back of and through the interface. The EOS software implements communication retry up to 10 times (10x) before forcing the cabinet into “flash.”

Verify Flash Storage Size and Capability

Under the ATC 5.2 standard, the flash storage requirement is 6 MB for O/S and application storage. Under the ATC 6.25 standard, this was increased to 16 MB. All systems meet the ATC 5.2 specification for total storage. This is adequate storage to host EOS and provide for its correct function. There are observed inconsistencies of the McCain controller file systems among the tested controllers, which requires changes to the software being ported.

Verify Front Panel Input

As noted in the previous section: “Front Panel LCD Operation,” all boards passed this test after introduction of I/O delay for the Intelight MPC8248 controllers tested.

Check SRAM Size and Performance

The ATC versions 5.2b and 6.25 both state the SRAM requirements in section 5.3.3. All controllers met the throughput requirement and size requirement.

Verification of USB Capability

There are known inconsistencies and issues with USB capability among some of the controller platforms. Rather than itemize conformance to ATC behavior, this presentation will illustrate USB capability on a per-system basis.

The evaluation was completed according to the selected USB requirements, per the ATC Specification versions:

- ATC 5.2b and ATC 6.25:

- Section 5.4.3, “Serial Interface Ports”, states “To facilitate the transfer of files between dissimilar equipment, all USB memory devices shall be formatted using the FAT16 file system. This provides for a maximum per-device storage capacity of 2GB (assuming 32kB clusters).”
- Section B.4.1 states “The BSP shall support hot-plug configuration.”
- ATC 6.25:
 - Section B.4.1 states “During boot-up, the ATC shall check for the presence of an executable binary or POSIX shell script named /media/usb/startup on an attached USB device. If found, the ATC shall automatically execute that file.”

LARGE DEVICES: One must observe that the Linux 2.6.x operating system builds have been in the field for just under 10 years, and that the embedded FAT filesystem support likely includes FAT-16 and FAT-32, even though the specification is FAT-16. Currently, all COTS USB storage devices should be supported, not just those under 2 GB.

Note that “auto-mount” is not a requirement of the ATC, except at boot time as specified in the ATC 6.25 standard.

References

1. Advanced Transportation Controller Working Group, ATC Joint Committee, ATC Standards (2006). *ATC Standard 5.2b*, Final Standard.
2. Advanced Transportation Controller Working Group, ATC Joint Committee, ATC Standards (2018). *ATC 5201 Advanced Transportation Controller Standard v06*.