

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

**Ken Leonard:** ITS standards can make your life easier. Your procurements will go more smoothly and you'll encourage competition, but only if you know how to write them into your specifications and test them. This module is one in a series that covers practical applications for acquiring and testing standards-based ITS systems.

**Ken Leonard:** I'm Ken Leonard, the director of the U.S. Department of Transportation's Intelligent Transportation Systems Joint Program Office. Welcome to our ITS standards training program. We're pleased to be working with our partner, the Institute of Transportation Engineers, to deliver this approach to training that combines web-based modules with instructor interaction to bring the latest in ITS learning to busy professionals like yourself. This combined approach allows interested professionals to schedule training at your convenience without the need to travel. After you complete this training, we hope that you'll tell your colleagues and customers about the latest ITS standards and encourage them to take advantage of these training modules as well as archived webinars. ITS standards training is one of the first offerings of our updated professional capacity training program. Through the PCB program, we prepare professionals to adopt proven and emerging ITS technologies that will make surface transportation safer, smarter, and greener. You can find information on additional modules and training programs on our website at [www.pcb.its.dot.gov](http://www.pcb.its.dot.gov). Please help us make even more improvements to our training modules through the evaluation process. We will look forward to hearing your comments, and thank you again for participating, and we hope you find this module helpful.

**Dave Miller:** Today we are doing T307, which is Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06.

**Dave Miller:** I'm Dave Miller. I'm the principal systems engineer at Siemens Industry, Incorporated. Been chair of the ATC joint committee that developed the ATC 5201 standard as well as some others. Also chair of the NEMA 3TS Transportation Management Systems technical committee.

**Dave Miller:** Our first learning objective is to identify the key elements of the standard equipment for testing documentation. In other words, what's in the standard and what's available. So we have to identify what needs to be tested. We also need to identify what doesn't need to be tested, because this is a very broad standard and it has quite a bit of interfaces you may or may not use. And then the other thing we want to really focus on is capture the open source Linux intellectual property obligations within the contract terms and conditions, because this is an open Linux platform so you have to consider the intellectual property rights, et cetera. Our second learning objective is to describe—within the context of the systems lifecycle—the role of the test plan and the testing to be undertaken. For that we're going to recognize the purpose, structure, and content of the test documentation. As with the other modules, we're basing that on the IEEE 829 standard. So what we're doing here is the test documentation for the ATC 5201. We have a test plan, test specification, test cases, test procedures, test reports, so it follows the workflow that's familiar to the other modules that we've taken, like ramp metering, that sort of thing. The third learning objective is we want to describe the application of a good testing documentation for transportation controller based on the standard, so we want to identify the key elements of the standard and what is relevant to what is covered in the test plan. We want to identify the key elements of a conformance statement. We have to make sure that our ATC in the end conforms to the standard. And then finally our fourth one is we're going to describe the testing of ATC and we're going to create a sample testing documentation for the test process. We're going to walk through a sample test plan, test design, test cases, and test procedure.

**Dave Miller:** Let's move onto it then. So first of all, our first learning objective was to identify the elements for testing. This was what needs to be tested and what doesn't need to be tested.

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

**Dave Miller:** In this module we're going to teach the agencies how to create test documentation; that includes the test plan specific to the roadside controller needs based on the ATC standard and also on your agency requirements. We're going to use IEEE 829 as the format, so that it's very familiar among different projects.

**Dave Miller:** Why do we need testing documentation? We need it to conduct a test process to establish conformance to the standard, compliance to the configuration, and compatibility. It's the three Cs. We have to make sure we conform to the standard so we're interoperable. We have to comply to the agency configuration so it meets your local needs of what you're trying to do that another agency may not be trying to do, and you want to make sure that you're compatible among the objects so that it works everywhere throughout the country.

**Dave Miller:** What is an ATC? I know a prerequisite for this module—we're kind of picking up from the 307a and the 307b that you've probably taken before, so you can refer to these modules for more detail. We're not going to go back through what is an ATC. We're just going to kind of pick it up as it relates to the testing aspects. So the ATC 5102 standard: what is the scope of the standard? Well, the standard defines the minimum required functionality of the hardware and software for controllers that comply with it. It includes things like a minimum processing speed, the ability to multitask operating system threads so you can do more than one thing at once, a minimum set of user communication interfaces. There's no really limit on the technologies as long as these minimums are met. You can get an ATC from a vendor that may have things that are beyond. That's fine, but at a minimum they have to have these basic things and we have to verify those in our test plan. Keep in mind that the open architecture approach allows software to be purchased independently from the controller. So it's not really like an appliance—you take it out of the box—it's not necessarily the software comes with the controller. You can buy that independently from a software developer. It enables ATC to be used for traditional traffic applications or lots of other applications. We're going to show you a few of those. And other than the minimum physical interfaces, the ATC standard is really not specific on the mechanical aspects of the controller, so this allows the controllers to be built in compliance with the standard but your software runs across different manufacturers and can be interoperable in a wide variety of cabinets—such as NEMA, 332 style, Model 2070, ITS cabinets—and in Section 1.4 of the documents it shows those different configurations. We're not really specifying a specific cabinet layout here or the controller size and shape. It just has to conform to certain overall maximum size, and the interface has to be plug-compatible at a minimum for interoperability.

**Dave Miller:** This is kind of a review. How is ATC used? So the 5201 is used with the companion standards for the API, which is ATC 5401, which is covered in some separate modules, so these all go together. This is general purpose in nature. The ATC, is an open Linux platform, and it may be used by ITS applications that we don't really know about yet. There's lots of connected vehicle things and applications on the horizon. So that's why the open nature. The functions may over time expand operational user needs, but even not knowing what's in the future or what the future holds for us, a number of the basic operational usage scenarios must be tested for conformity. We'll show how we've arrived at CV applications and how those will be supported. We'll kind of touch on that in this module.

**Dave Miller:** What is included in the standard? The standard has major objects and interfaces. So there's a large number of objects—there's about 73 that are included in the standard—which an agency will use to create a procurement configuration. There's mandatory objects: there's objects that are always included in the procurement, and we'll go over what those are. These have to be in place, so that if you buy software from a third-party, you develop it yourself, it runs on all the different ATCs from the different manufacturers. There's core standards that always have to be included. And then there's optional objects

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

that are selected per user need, and these are things like: What type of cabinet do I have? Do I have a 332 cabinet that has the large 104-pin connector? Do I have a cabinet that's a NEMA TS2? So you wouldn't always put all the objects in the standard; you would put the mandatory ones in, and then you would select the ones that meet their local needs. And then finally testing documentation includes what is being purchased by an agency. So when you put together your testing documentation, what you want to do is have a document that you can use. After your agency puts together a procurement, the equipment comes in, you want to verify that it works, you want to validate it to the standard, make sure it conforms—so that would cover the mandatory. And then the selection of optional objects that you used for your local needs. In the end we call it a Procurement Configuration. There's lots of different interfaces for your local needs and cabinet styles. Your procurement will go out showing the configuration of which pieces of the optionals you want from the ATC 5201 standard. That configuration is what we're going to address in our testing.

**Dave Miller:** What is included in the standard? So the standard is really good. If you look at Section 8, you can see it has environmental and test procedures. In the past there's been a lot of questions about, "Well, do I test all of this all the time for every unit that comes in the door? What do I test for when I am conducting qualification testing to accept a vendor?" So this is really good that it lays out—it has some normative references to TS2; it's useful for your Master Test Plan; and it can be conducted by the end-user agency laboratory, or it can be done—if you don't have an in-house laboratory, you can have it done by an independent laboratory. A lot of times the manufacturer is expected to do the independent laboratory testing, and they can supply that to you.

**Dave Miller:** It's important to know what is not included in the standard. So as we talked about previous in a couple slides back, the API—the application program interface—is not included in this standard. It's what we call a companion standard. If you want more information on that, we covered that in A308a and b; that is something that runs on the ATC, but it's really not part of the standard. If you want that, you would have to specify it explicitly in your procurement documentation. Since this is an open Linux platform that can do things other than signal control; signal control software actually doesn't come with the controller unless you order it, and it's not covered by the standard. This is a standard for the hardware platform, the operating system, and a lot of the libraries that you would need to support applications such as signal control. The history of the standard—it's been around a while, so it means it actually had its roots back before we were doing system engineering processes. In the newer standards, you'll see that the standard starts with what the user needs were—usually from a workshop of agencies, the owner/operators of the equipment who will be buying it. The standard includes requirements and in the standard, if it were an SEP-based standard—this one wasn't, it was done earlier—you'll see some of that in there. But when you do your procurement document, you really have to make sure you have requirements that meet your needs. That's one of the things we're going to cover. You can go back to the earlier modules in this training scenario for more information.

**Dave Miller:** Diagnostic Acceptance Test. Diagnostic acceptance test is performed on the first article of an ATC unit as part of the preparation or pre-deployment—typically not performed on all units. Most of you are familiar with getting a piece of equipment qualified on the Qualified Products List. What test would you run to be qualified? That would be the Diagnostic Acceptance Test, and we'll show you here what part of the Diagnostic Acceptance Test may be a piece of self-test software, since it doesn't come with signal control. It's not typically performed on all units. You probably—not necessarily want to put every unit you're getting on a vibration table and expose it to three dimensions of vibration, that sort of thing. But then beyond the Diagnostic Acceptance Test, there's production testing, those are performed on all units. So if I'm manufacturing an ATC, I would have a production test at the end of my assembly line. If you're procuring them, you would probably have some sort of an acceptance test on all the units that come in, to

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

make sure they're running out of the box; we'll talk about that. The ATC 5201, if you look in that section—and I have a little piece of it showing here at the bottom—you can see that every one of those sections in parentheses, if it says DAT, that means you would expect to run this only for qualification, like a manufacturer. You probably wouldn't want to run it on every single one that came in the door. If you go through this section, you'll find other sections—like this is 8.5.3. You'll look at some others and it will not have this DAT off to the side, which means that would be something you'd probably run in production. So it's kind of nice how this is set out, so it sort of shows in each section which one you would do one time at qualification, and if it doesn't say DAT, you would probably want to do that on your production end for like an incoming. And agencies typically use automated self-test diagnostic test software. Those of you familiar with the 27s, there's a DAT available from some of the agencies. You load it in lieu of the traffic control application, you plug the outputs back to the inputs, you run the automated test, that sort of a thing.

**Dave Miller:** We're going to suggest an approach to preparing testing documentation. You might want to go back and consult the A307b module for guidance on identifying mandatory and optional objects in the requirements. But here it's important to find the essential areas that must be addressed for conformance to the standard. We want to know if it operates correctly with operational voltages. What if the voltage is too low? What if the voltage is too high? What is safe operation? The input/output requirements; what type of a cabinet you could have; user interface requirements; do you want push-button and display, or do you not want a push-button and display, you just want to do it remotely from a tablet? And we have to have some minimum CPU—central processor unit—performance and memory requirements. We want to make sure that there's enough room and speed to run your applications, and there's a lot of other requirements that we'll talk about. But the cabinet standards influence this, because there's different interfaces to different field I/Os. We have to assess the agency procurement configuration, and we'll talk about that here shortly.

**Dave Miller:** We're going to start to review the mandatory—what we're calling here the core part of the ATC standard and the optional objects. What we call the core, if we take a look at this, in this green box here that I'm showing—we'll go into that in more detail—but this slide is to show you that this part is always required—the core. And then there's different things—elements—and the objects that you may or may not want. So you always get four mandatory ethernet ports; you get this core here, which we'll show in the next slide. It always has to come with a USB—universal serial bus—that has to have at least 2.0 compliance. And then we go into the optionals. If there's a keypad and display, that's optional. In this day and age, if you're doing something from a tablet computer over Ethernet or over a Wi-Fi link, you may not even want a display. But if you have a display, we have to make sure that they're the same among vendors. If there is an 8-Line display, it's optional that you buy it. But if it's supplied by the manufacturer, it has to have the same interface, so they all work the same way and your software operates the same way. Same with the field I/O that we're showing down here: this happens to show a NEMA TS type 2 with the round connectors. This could be a 2070-2 module that has the 104-pin connector to a NEMA cabinet—or to a Caltrans-style cabinet, serial ports. It may have a—back in the day we had the communications interfaces that were the frequency shift key modems. We're going more to IP address now; those are still compatible with the standard. Even the housing is optional. You could mount these core boards into a rack, if you wanted a rack mount system. You could get a shelf mount or you could get a rack mount, like in the 332 cabinets. What we're trying to get across here is you always have to have the core—the mandatory Ethernet ports, the USB—so the software runs correctly across all manufacturers; but from then, it's your local needs. In your procurement spec, you say what type of user interface you want and what type of a user field I/O to your field devices, your load switches and detectors, whether or not you need a comm., and then what sort of a housing you want.

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

**Dave Miller:** This is a little bit of an expansion on the green box area in the center that we saw before. So this is mandatory. To be conformant to the standard ATC 5201, you have to have an engine board, the manufacturer has to have a host module, it has to have a power supply. So these are mandatory. If I go from Vendor A to Vendor B, I should open it up; I should see an engine board; I should be able to unplug the engine board from one, plug it into another; the pins match, the signals match. So the idea behind the engine board was when we didn't have that, we had obsolete electronic parts. The most vulnerable to obsolescence are the computer chips and the memory chips that change quickly. Instead of having to redesign boards and go try to find sources of old stock of parts, change the engine board. It has modern, up-to-date components. Say it's ten years from now, the computer chips in this engine board aren't made anymore, but you can find the same manufacturer or other manufacturers that have the engine board, plug it into the same place, and you're good to go without having to worry about obsolete parts. Then the engine board plugs into the host module. It has the two Ethernet switches—I think we covered this in the prior modules. So we have the Ethernet connections traditionally to a network connection, a diagnostic port, roadside equipment; and then we have this thumb drive that we have a USB port so that we can store setups for databases—that sort of thing. And then mandatory at the end is a power supply. The power supply is important because we have signals that we bring in the time base from the service frequency. So among manufacturers, the power supply may be different sizes and shapes but the signal is going to be clean if they have the same signal names, same signal characteristics among manufacturers.

**Dave Miller:** This is what needs to be tested. We have to make sure when we get the board—ATC 5201—our test plan should make sure that we conform to the standard in that it has microprocessor, Ethernet ports, everything shown here. So that's what goes on the engine board. A lot of this doesn't necessarily have to be tested; we just have to make sure in our test plan that we are conformed to the standard. We can do that by inspection; we can do that by looking at the data sheets; we can see what microprocessor does it use and how many instructions per second does it do. But we have to make sure that we have engine board that conforms to the standard.

**Dave Miller:** The next thing we're concerned about is the microprocessor. Does it have the speed that we need?

**Dave Miller:** It has to have—the engine board itself—has two Ethernet ports, two independent ports.

**Dave Miller:** It has a real-time clock, so we're getting time-of-day from the Linux operating system that might be corrected by a GPS or pushed out from the central. During power fail, it's backed up with a supercapacitor so it keeps the time running. So when we come back up, it accounts for the time that was lost during power down.

**Dave Miller:** It has to have memory, which includes dynamic RAM—random access memory; that's what executes the software at full speed. It has static memory to hold configuration during power outages; and it has—in this day and age, it has flash memory, pretty much acts as a solid state disk drive so that that stores your files and/or different applications, software applications, so it can boot up. It would take that from the flash memory and run it on the microprocessor.

**Dave Miller:** We have serial inputs and outputs. I'm not going to go into these in detail.

**Dave Miller:** Serial peripheral interface. This is used to sense what host port it's plugged into. So I may take an engine board from Vendor A and plug it into a host port from Vendor B five years from now. How would it know what it's plugged into? So the host board has a serial peripheral interface; it has a small memory; so when you plug it in, it's an identifier. So the engine board can actually read what host board

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

it's plugged into, know the manufacturer, know what configuration it is, so it knows how to access the peripheral devices.

**Dave Miller:** And then finally the P1 and P2 connectors that are shown there are standardized among manufacturers—I think we talked about that. That's so that you can replace the board later and not have to worry about obsolete parts.

**Dave Miller:** So we go on to the host module. A host module is mandatory as well. You have to have something to plug the standard connectors—the engine board—into. That's pretty evident that you have to have a host module that has the same connectors with the same pinout—that's covered by the standard. That picks up the power, the signals, the Ethernet ports, but also it's required to have two Ethernet switches or it's also—the standard allows one VLN—virtual LAN—there, so it can have one device that implements two switches. But what we're trying to get at here is you want to have a firewall or some physical or virtual barrier between what goes on at the roadside and what goes on at the central. Back in the day when we had one Ethernet port, you'd go out to service it, you'd unplug that, plug in your laptop computer, get the alarms at the central. So this was an attempt to go ahead and have a security separation between what goes back to the central—so that could be managed by whoever manages your central. Then you could have a local area network at the roadside and the two don't disturb each other. And then off to the side here we have the mandatory USB port. Has to be at least USB 2 compliant.

**Dave Miller:** Host module. This is blown up a little bit higher scale. You can see the two Ethernet switches, and that could be a virtual LAN that implements two at the firewall. It has the four Ethernet connectors that come out on the controller, a place for a thumb drive, and then we have a communication slot. We are requiring in the standard that you supply at least one communication slot that would be able to accommodate the modems—like the frequency shift key modems—because there's still some old legacy twisted pair out there. There's a lot of it, works terrifically fine at low speed; so we wanted to make sure that when you change your controller you could unplug your legacy modem card, plug it in, and the pins still match and it still works.

**Dave Miller:** So what do we have to test here out of all this? The mandatory user interface requirements must be tested. This may be a little confusing in the standard, but it also allows quite a wide latitude of things for user interface. So you have to have at least an active LED indicator. You look at the front, you can see it blinking on and off, you know it's healthy, and at least it thinks it's running and it's executing code. Has to have one Ethernet port, has to have a USB port for removable memory devices. We're not really saying that we're setting up like a personal USB network here that you can expand out infinitely. It was really there—the need was for a removable memory device. To that, there's two things you can add. You can add a nine-pin D connector if you want, have a laptop that still has the RS-232; so you can plug that in for a console port; or a modular jack for a serial connector to the console; or you can go with a D connector for an external front panel or liquid crystal keyboard and bell. So it basically says you can have either the traditional front panel with the LCD, the 8-line or 16-line display with the keypad, and it has the—they call it the bell, it's the buzzer—and that would be if you stood in front of it, you don't have any sort of a smart device, you can push the buttons just like you've always been able to. Or if you don't want to do that, the standard also allows you to have a console connector and modular jack, and you don't necessarily have to have the front panel of the keyboard and display. That would be more popular, since we have to have an Ethernet port. That can go to some sort of a Wi-Fi device or it can go to a laptop computer, tablet—whatever you're dealing with—you can use that in lieu of the standard keyboard and display. You can have a graphic display with a touch on it to operate the keys.

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

**Dave Miller:** The final mandatory hardware object here is the power supply. What we're doing here—that's what we show down here in the ATC Core. So you have the engine board, you have to have a host board, and you have to have a power supply.

**Dave Miller:** So what we're doing here—this is a picture of an ATC power supply. Basically what it does is it converts service voltage—high-voltage, usually 120 volts—and it converts it down to the level of DC that electronic parts use. It also provides a time base; it provides a power-up signal so your electronics don't start to run until the power is stabilized; and it has a power down signal that interrupts the running application that says, "This power supply is going to hold up for about a half a second, but be aware it's going to go down shortly." So if there's any housekeeping you have to do before the power down, you can do that. And it also does power conditioning. We all know that there's nearby lightning strikes, power surges—so the power supply, per the standard, has power conditioning and filtering.

**Dave Miller:** So let's just say in the voltages for the power supply—let's just pick out one here. What would be mandatory to test? Operating voltage. If you look in the standard, it says the controller shall meet the operating voltage requirements per the section of the TS2 standard. There's a lot of places in the standard, in the testing part where we don't really replicate it, we just make normative references to other standards. Operating frequency same thing, and power interruption same thing. So this would say: What are the operating voltage levels? High line and low line where it would operate correctly. What are the frequency around 60 Hertz in the U.S. that would be tolerable? And power interruptions. If I have a very short power out—say less than have a second—does it run or does it reset? Well, in this case it'd run. If it's longer, at some point you want it to force a reboot, because if the power's been off long enough you want it to reboot and start over when power comes back. So these are the types of things we're going to be testing. We'll show you some test cases when we get farther along here.

**Dave Miller:** Optional object examples. We've touched on this already, and user interface is optional. The USB interface is mandatory, but the device is optional. You can plug it in or not plug it in. We talked about field I/O; we're showing here a NEMA TS2 style field I/O, but you can have the 104-pin connector for rack mount system, and you have a choice of housings. So these are things that you'll either have in your procurement spec or not, or you will specify from a list of different types. The agency—it must be tested, if it's included in the procurement specs. So you wouldn't want to say that, "I want this NEMA style ABC connector," I procure it, and you don't verify that you got that and it works properly when it comes in. So we would want to make sure we include that in our testing documentation.

**Dave Miller:** At this point we're going to go through a little case study.

**Dave Miller:** This is probably a pretty simple example. Throughout this module, we're talking quite a bit as examples we're showing the NEMA cabinet configurations; and I know a lot of you out there—lots and lots—have 332 style cabinets. We're sort of showing the NEMA here because there's a little more difficulty in the different types of sensor—the TS2 type 1, TS2 type 2, the older TS 1 standards. When you have a 332 cabinet you have it a little bit easier, because the 104-pin connector has always been there. So just imagine the same thing; in lieu of the ABC connectors, you have the 104-pin connector. The case study here—we're going to go up a little bit in difficulty. So we're going to say that the current situation—so my current situation is: I'm an agency; I have to buy some ATCs; I don't want to change out my cabinets, I already have TS2 style cabinets; I want to do ATC 5201 installations; and the procurement configuration requires the NEMA TS2 cabinet interface. So our testing task is: lets make sure that we're compatible with such a configuration of what we're receiving.

**Dave Miller:** So we would need to require a NEMA field I/O. So if you look in the standard, it shows an option for a NEMA field I/O with the round connectors. I probably want a shelf-mount housing instead of rack mount. And say I want to be able to go out and push the buttons on a display if I'm called out into the

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

field and I'm off duty or whatever, and I don't really carry a laptop with me. I want to be able to just open the cabinet door and push the buttons. So I do want a display, and I do want a keypad that becomes part of it. The need for compatibility—requirement for TS2 power supply and it's optional for the field I/O with NEMA ABC connectors—that's the need. How are we going to verify that? For procurement testing, our test flow must be developed to ensure—to verify that the ATC came in configured correctly with the right options; each ATC element can be tested; and ATC objects configured to an ATC will work correctly in the cabinet. So did it come in correct? We have to test everything, and once it's tested, we pass the work in the cabinet, if it's passed the test.

**Dave Miller:** So what we do here in our little use case study here is—so this is a needs to requirements traceability matrix—we're bringing this forward from A307b, so this should look familiar if you've taken the other module. So what we do here is we have a user need identification. Then the need, the functional requirement—FR is functional requirement—identification. What is the requirement? What's the conformance? Are we supported? And the additional specifications. So you'll see here that "shall include an engine board," that's mandatory; "shall include a host board. Shall include a power supply. Configuration shall include an IP communications to the transit management center." So it shows functional requirements to needs. To be compliant to the ATC standard—that's under User Need 1—it has these four requirements. Those have to be met or you're not compliant.

**Dave Miller:** Another thing that needs to be tested—that previous slide touched on the hardware, so now we're going into the software objects. We have a need for the software I'm purchasing; maybe from a third-party operates on here. So this shows the buildup—and this should be familiar with from the earlier modules on the ATC. We start out with the hardware layer, which is the engine board. It's Linux based, so it has an operating system and driver. So the ATC standard—there's an Appendix A and B in the back of it and it says, "This is the stuff that manufacturers shall supply as part of their ATC." It has to have these Linux parts in it, so that your software will run on it. We talked about the API. This is where it would fit. It's not part of the standard, but it's a companion standard. So your application—and let's say this DAT—diagnostic acceptance test—application is a self-test. We're going to take the outputs and connect them to the inputs; I'm going to run that when I take it out of the box to see if it works. So the application layer I could run it directly on the controller operating system if I didn't want the API or if I didn't have it, or I could also run it on the API. We're not going to go into the API in any detail because that's a complete set of other training modules, but it's basically a way for a piece of middleware that would go out and read the inputs, store the outputs, get data from the key, store data to the display so that it's interoperable among all the manufacturers. That piece of code to go get all that stuff, put all that stuff back out—inputs and outputs—and run the real-time clock. You don't have to have code to do that every time; the API takes care of that and it gives you a standard place to go get that information and put that information.

**Dave Miller:** So the DAT application layer—we talked about that. This would be—if you go back to the standard, it shows what should be tested and what isn't tested. So this application, in this use case, if I'm doing a NEMA TS2 style cabinet that has the ABC connectors, I would go out and look for a loopback cable that I could plug on there, so the outputs from the load switches would—instead of going to the load switches, it would loop back and go to inputs—detector inputs—and you can run that on power-up and that would—it would be a very good self-test. So that would run right here.

**Dave Miller:** We're going to talk a little bit about open source obligations. This is actually pretty important, and you really should consider it when you're writing your procurement specification. Since this is a Linux open source, you have to have a clear and unambiguous list in who owns the intellectual property and the derivative works. What that means is, if I go to an open source portal and I put it in my ATC, I need to have a document that comes with it that declares where all of that came from, who



## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

actually owns it. I didn't develop it, I got it as an open source; I'm not claiming that I own it; this is where I got it; it's under a license like Apache that says it's okay to use, it's open source. It's very, very good practice to have that in the procurement and make that come from the ATC manufacturers, so that you know that once it's on the street, somebody is not going to come to you and say, "I think some of my intellectual property is running in your controller." You can pull out this open source declaration and say, "Actually, this is where all the software came from. It's all above board, I got it properly." And I have a link to where you can go read more about it. But all we're trying to get across here is that when you procure an ATC, as part of your procurement you should specify that the manufacturer shall supply an open source declaration statement. I got a little graphic here that shows it's a document; it's maybe no more than two or three pages in most cases; but it will tell you every piece of code, where it came from, and who owns it, and if it's licensed to be reused.

**Dave Miller:** The other thing is the ATC standard requires the delivery of a board support package. This goes along the lines of it's an open source Linux. So what this has is a library and tools to compile software for the ATC; it's required by the standard. If the manufacturer of the ATC claims that they're compliant to the standard and they don't supply that when you ask for it—it doesn't necessarily have to ship with every controller—but you should have a copy of that. The reason for that is if I'm getting my software from a software developer—so I may be buying my controller from Vendor A and I may be getting my software from Vendor B—okay, fine so far. Five years from now, I want to buy my hardware from Vendor C. I need to have Vendor C's board support package to give to my software developer, so that they can take their code, compile it, link it, and load it on Vendor C's hardware. So that's very important to get that. You may never use it, but if you switch vendors in the future, you're not using the same software from that vendor—which is starting to become more common today—you want to make sure you have the board support package so when you want to bring in another vendor, you have to take the board support package, send it back to the software developer who's doing the code, and say, "I'm now running my code on Vendor C. Here's their board support package. Please compile it, link it, and load it, and give me a loadable file so that I can put it on there." So it's very important that you get this, and it's required by the standard.

**Dave Miller:** This is going back to open source terms and conditions. You want to make sure that your terms and conditions, when you go out for a bid—it should be very clear that the manufacturer shall supply an open source declaration that identifies the source of all ATC software that's used; shall supply copies of all open source distribution licenses; and shall supply a board support package and instructions. Those are the three "shalls" you should have when you're contending with an open Linux platform like we are with this ATC standard.

**Dave Miller:** We're up to the point we're going to do an activity here.

**Dave Miller:** So the question here is—and this is a "not", so this is a negative. What is NOT an ATC procurement deliverable? If I'm running a procurement, what is something I would NOT want to include as a mandatory deliverable? We'll go ahead and open this up for a while; I will go ahead and move on to the answers here. Your choices were open source declaration, is that part of it; board support package; API, program interface; or the open source distribution license. So of these, which is not a deliverable? We'll move on to the answers.

**Dave Miller:** So the correct answer is the API. We said that the API is a companion standard. It's becoming more and more popular. But when you're writing a procurement specification for an ATC 5201, you wouldn't necessarily have to include that, because there's lots of applications out there that may or may not use it. However, you should have an open source declaration for the reasons we talked about. You want to make sure that you know—when you procure equipment on the street, you know who owns

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

the software and you're licensed to use it. And you want to include a board support package so that when you want—if you have a third-party software developer, that's what they will need to compile and to load their software onto the hardware platform. We talked about the open source distribution licenses. The one I referenced here is there's—a lot of what the USDOT is doing—and some of the universities that are doing open source software that you can download from their open source portals—you will see an Apache license there. It basically says that it was developed by this university, it's available for general purpose distribution for the intent. "We don't claim any liability for what we do. You're on your own, and here's the license. You're free to distribute it." So if somebody comes and says, "Where did you get this software?" You can hold up the Apache license and say, "That's where I got it."

**Dave Miller:** So we're going to move on to the next learning objective, which is describing within the context of a systems engineering lifecycle—so we're going to talk about the role of a test plan and what testing we'll need to do.

**Dave Miller:** This is the role within the context. If you remember back—we've been over it a number of times in the other modules—the V diagram and, going up the right side of the V, we're verifying that we conform to the standards and we meet our user needs.

**Dave Miller:** Let's talk about test documentation. What's the purpose? Why do we need these test documents? It's a general term used to describe unambiguous and common understanding among all stakeholders to outline what to test, describes how to test—what process do we go through to test—and we put it in the format that matches IEEE 829. IEEE 829—I show the cover of it here—doesn't tell you how to test things. It shows you a common format and a common process to do the testing. So if today I'm testing an ATC 5201 and tomorrow I'm testing a connected vehicle system and the next day I'm testing a ramp meter system—if you use this common format, it'll always look familiar. So you can reuse it for another one. So if another agency that is one of your stakeholders that you are working with—if they have a ramp meter and you have a ramp meter—an ATC 5201—you could pick it up and look at it and say, "Oh yeah." The table of contents looks familiar, the organization looks familiar, the process we go through looks familiar. So it's just to make things look familiar and make it so that it has something that's very common between them all.

**Dave Miller:** The key part of the testing is a test plan. The test plan—it's what it says. It identifies the test design, the test cases, and the test procedures. It kind of shows here in our graphic off to the left, we're implementing the test plan. You start out with it and you do a test design, like a unit test; then you do test cases and test procedures. So that all comes across in the documentation. This is the documentation you would do before you run the test. There's documentation again after we get the test run—which we'll go into—but you should have a test plan documentation in 829 format, and it includes the things shown on the left side here of the graphic.

**Dave Miller:** The test plan—if you look in IEEE 829, it'll show a table of contents, and there's a nice little section describing each one of these—but just in general, the test plan describes the overall approach to testing. The test design specifies the details of the test approach, so it's like a unit test; and a design exists for integration test and acceptance test, like we talked about before. The test case specification outlines a set of test inputs, the conditions under which you execute them, and the expected result. And the test procedure defines steps, so it's like a step-by-step process of going through it and selecting the results of the test.

**Dave Miller:** Test plan outlines the testing process that's used to verify the requirements. The test plan includes a description of the approach; it includes a high-level overview of how the test will be conducted; and in addition to that, the test plan lists the resources needed, like the personnel, test equipment needed to conduct the test, skill sets, materials, maybe samples of the units you're going to test, and lots of other

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

things. And the plan includes high-level schedule—how long it's going to take to complete the test at scope—and what resources are needed. The schedule has to conform to, "I can get it done in this amount of time, using the resources I'm saying in this plan, and the approach I'm describing in the plan." The plan is going to identify what needs to be tested, the features, and the list of testing tasks, and also identify risks that require contingency plans. When you write one of these, you're going to pretty quickly discover that, "Hmm, this might be a risk here. I don't know if I have enough time. I don't know possibly what happens if it doesn't work the first time and I have to have somebody identify the anomaly." So you want to list those as they come up—just list the risks. Make sure it's in the plan that says, "When we're planning this, we think that there might be something that's going to come up here that's unexpected, and this is our mitigation. This is what we're prepared to do about it." It's always good to think about the risk and document things as you go along, so that when everybody reads it you can say, "Oh yeah, we did think about this, and we think that there's a Plan B here that would work better if this comes to fruition."

**Dave Miller:** This is the familiar V model that we've talked about in the prior modules. All three components are used to verify the ATC requirements, and this diagram is used to point out the requirements verification. So these red arrows here are—you began writing your procurement specification with needs, so we have a Need For in our use case. We have a NEMA TS2 system validation. "Okay, we have to validate that it has the correct interface and it's functioning." The requirements—we want to verify that we met the requirements step by step. So that's the test design specification, test cases, and procedures. What we're doing is connecting the source on the link-backs, so that when we get all the way done that the folks that are going to be using it—owning it and operating it—it works like expected from the beginning of the project.

**Dave Miller:** Test reporting documentation. What we talked about documentation in the prior slide, that's what you would prepare ahead of the testing. The testing reporting documentation is prepared both during and after. So before, when we're doing our test plan, you should do that only before; but the test plan execution—these reports are used for interpretation and recommendation. So you can see from the boxes here that you would prepare a template for a test result. So you would always—when you have your test procedure, the next step would be to do test reporting documents, and we'll show you some examples of this, but really what you're doing is—if you're putting a template—is, "When I run the test, this is the data I'm collecting. I'm going to set up the inputs like this. I'm going to apply a certain voltage. I'm going to measure what comes out of the ATC. I'm going to record it here." So before you run the test, you have the template, you have your expected results, and after you run the test you'll have what the real results were. So when I run the test, if I turn the voltage down to, say, from 120 volts down to a little bit lower voltage, I would expect it to operate. If it doesn't, that's an anomaly; and I think we've talked about this in some of the others, but it's basically—when you talk about pass/fail, pass means that your test results match what was expected. If the test results do not match what was expected, it's not necessarily a failure; it's an anomaly. It's unexpected. So you write down in your test report, "This was what was expected. This is something different that I got that was unexpected," and then it goes to investigation, because it may not be that your unit that you're testing, your ATC, is at fault. Maybe your test equipment is at fault. Maybe the operator made a simple error—that sort of thing. So you record—you don't just necessarily say "failure" and stop. You would say, "Here's an anomaly I found. It wasn't expected." Go for further research, then it's all sorted out later. Then you can finally decide whether it was actually a failure of the unit or test or it was something else. So then at the bottom you have a level test report, and that summarizes everything. We have some examples of that.

**Dave Miller:** We're going to do another activity.

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

**Dave Miller:** The question here is which part of an agency's ATC testing document—which is part of the testing documentation? So if you're writing test documents like we just talked about, which of these four are parts of it? Is the test scope part of the document? Is the test approach part of the document? Is the resources and schedule part of the document? Or are all the ones that are shown here—A, B, and C—are all part of the document?

**Dave Miller:** We'll go ahead and answer the question. So the correct answer is all of the above, and that's what we just talked about. The test documentation includes the scope, what's tested and what's not tested; what's the approach, what's the process, how are we going to go about doing this; and also resources and schedule—who's going to do the test, what equipment do I have to buy or rent, what lab facilities do I need, et cetera. So all of those are included as part, so that would be D, all of the above.

**Dave Miller:** Next we're going to move on to our next learning objective, which is—and that's to describe the application of good testing documentation for transportation controller, based on the standard.

**Dave Miller:** This is, "How do we apply our test to the equipment—exactly how do we go about doing that?"

**Dave Miller:** So there is a process we go through. We start out with our requirements traceability matrix—Needs to Requirements—and I think we showed an example of that a few slides back. So the agency that's going to own and operate the equipment has needs; and when you get all done and you have an ATC in the door, it has to meet the needs. So we always trace the needs to the requirements. That answers the question: "What do I need to test? What's required to be tested?" We also do a requirements traceability matrix that answers the question "What sections or parts of the standard is affected?" So it's what needs to be tested and how do I relate that to the standard; because, in the end, we have to meet the needs of the user that's going to own and operate, and we have to make sure we conform to the standard. So if I'm buying equipment later from another manufacturer to expand the system, interfaces work and the software works. I've got this box down here that says "ATC v06 standard does not provide the NRTMs and RTMs." We have to do that on our own. So this standard was done quite a while back in the beginning, and it did not have that included. We're going to have to show some examples of how to do that, and it's really not that complicated once we get into it.

**Dave Miller:** Here's a summary of a test plan. This is the testing process. The testing process is all about verifying the requirement. Each requirement needs to be tested. Each requirement is validated by testing and is traced to a test case in the Requirements Test Case Traceability Matrix. Each test case lists one or more test procedures that end with a pass/fail declaration. Here we see that each identified requirement is to be tested. There's other ways to evaluate. You can evaluate by inspection, you can evaluate by testing—there's ways to do it, but since this is the module on testing, we're going to focus on how to make sure we conform to the standard by the testing portion of it.

**Dave Miller:** This is an example of the NRTM. So, what needs to be tested? We're still kind of going along here with our NEMA TS2 ATC configuration with its requirements. The user need was "I've got to make this fit into a NEMA TS2 cabinet." The functional requirement is a TS2-type interface. It's mandatory that I conform to that—that's what the M means—and the standard does support that. If I look in the standard 5201, I can find the section that describes that interface, and then on and on down here. So I won't go through these again, but operating voltages, frequencies, power interruptions, and display size. So these are the needs by the end owner-operator, these are the functional requirements, and this is how we trace it over to make sure we're conforming to the standard. And you look in the standard, if it says "Yes," there is a section that supports that.

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

**Dave Miller:** RTM traces hardware and software objects for the standard to be testing. This is another example, so this is a cross-reference. On the left is the requirement ID, then the stated requirement, the standard reference, and then the standard itself. This first one is my requirement: "I have a requirement to go into a NEMA TS2 type 2 cabinet," so I need that interface. I look in the standard: there it is, Section 7.2.2. I see that interface, and it describes a parallel connection to NEMA TS1 or TS2 type 2 cabinets. These are assigned—these are the references to the standards, and this is how we verify it. This could be tested. You could plug it in and run a self-test. You can inspect it to make sure it's right. But in the end, you have to make sure that it conforms to the standard, so if you buy a controller from someone else later, it's the same—it works the same way. Your software runs on it.

**Dave Miller:** So this is the NRTM plus the RTM—this would be your procurement configuration. We talked about that earlier. So this answers the question on what an agency desires in an ATC unit. Both of these are used, and they're sort of summed together as they show this. The project needs are the first one. "What do I need to make sure the owner-operators get everything they need to run their system?" You add the RTM from the standard—which parts of that comply to the standard—and then finally the test: the RTCTM. This shows the testing that must be done. "So what do I need, what does the standard support?" Put that in the document, so I can make a test procedure and get test results for it.

**Dave Miller:** RTCTM traces each requirement to the test cases which are used in the testing process. We're back to our functional requirements. We state the requirement. This is the test case. So the test case is, "I want to validate that Section 7.2.2, what I'm receiving from the manufacturer conforms to that standard." It's got the round connectors, it's got the pins in the right place, the signals are the same, and the voltages respond, inputs and outputs to the same levels, on down. The next one is, "Signal control source code shall compile and use," so that means you have to have a board support package. You look in the standard, it says 3.3.1, it talks about the board support package. We don't need to talk about all of these, but this is how we show—to make sure when you're all done and you get the ATC, that your software will run on it, the plugs are compatible among vendors, and is supported by the standard here. So you'd go through those one by one by one and check them off in your test cases to show "This is how I'm validating that this was done correctly."

**Dave Miller:** So here is: How does the test case perform conformance to the ATC standard? This would something—this would be like a test case. So this is an example Test Case 1. The title is Power Fail/Restart. This is mandatory, so the objective is to make sure that the power fail and restart is mandatory to be done. The test case verifies that the ATC is unaffected by service power interruptions. The test case verifies that the ATC restarts when the power is interrupted for a period longer than 550 milliseconds. Inputs your service voltage. So I have my unit under test connected to service voltage, and some means to varying it up and down—say from 85 up to 135 or more—and you can interrupt it. The outcomes. So the outcomes are—we have some expected outcomes, and if you look at the standard—we won't really go through them, but it's expected per the standard to operate over a certain variable range of service voltage levels. It's expected to operate or not operate depending on how long the power has been interrupted. And the outcome is we would have—the test case would show our expected outcomes.

**Dave Miller:** The last step in the testing process brings up the results. This is actually what we would have in a test procedure. Step 1 in our procedure—and what we're doing here is—the example we're using is testing the power supply section. What we would do is say, "Set the input voltage to 120." We would expect that—DUT means Device Under Test—so that's our ATC. We set the input voltage to 120, attach it to the controller. We would expect it to restart and run the fitness test. The test configuration would be ATC that I'm testing, I have some cables that loop outputs to inputs, I've loaded a fitness test in there, which I'm calling Fitness Test—it would be part of your DAT—and it would run that test

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

continuously. So you would expect that. Set the service frequency to 60 Hertz. You would expect this test to still report this running continuously. Interrupt the power for half a second and bring it back up again. We would expect the device—the controller—to keep running as if nothing happened. And if we interrupt power for more than a second, we would expect it to reboot. So that would be like a test case. This is what the procedure is, and this is what we expect for the result.

**Dave Miller:** There's some issues here we should talk about just to point out in passing. Conformance to ATC 5201 v06 conformance. ATC 5201 v5.2b—so this has been around a while, like we said. It was first published—which means it went into effect and became a published usable standard—on September 25, 2006, and the way the standards work, there's maintenance done on them by the working groups. After five years, the standards are reviewed, and we can still say it's reaffirmed, which means, "Oh, that standard is still good. Nothing's really changed." Or we have to maintain it, which means we actually want to go in and collect some more user needs and we want to update it. So after the five-year period, 5.2b remained in effect; but at the end of the five-year period the working group decided that we really should update it because there's some things that have come along to make it more modern, and there's some obsolete technologies we shouldn't be concerned about anymore, and there's new technologies you want to put in it. So once it's published, ATC 5201 v06 takes effect on the publication date. All of the standard is no-cost download from the ITE website, if you go look there. As you can imagine, you really can't conform to both of them because there were some changes made—there's some very subtle changes that were made—and it's good to have that information in your procurement documents.

**Dave Miller:** What we would recommend is that we would go to our manufacturer. We're replying to a Request for Quotation. "Please provide a conformance statement. Tell me which parts that you comply to." These are some very detailed things, but in ATC 5.2b, the serial peripheral port—the way you address it with 4-bit—is slightly different from v6.0. So if you sent the same 4-bit number to one that compiles to 5.2, and you set the same 4-bit binary number to a 6.0, it would work slightly differently. We talked about previously 5.2b required two physical hubs or switches. 6.0 allows that, or configuring one device as two virtual LANs (VLANs). It's good to have a conformance statement that comes—when you get it back it says, "They conform to all of v06 requirements. This is some subtle differences, interpretation," and if there's anything like that, you want to get that in the conformance statement. It may come back and say, "We're compliant. We conform completely to v06, and we've completely gone away from v5.2." Then you wouldn't have to have these things listed out. But it's always good to ask for a conformance statement, because there are some subtle differences.

**Dave Miller:** Okay, we're going to do another activity.

**Dave Miller:** So the question is here: What is the primary purpose of the RTCTM? And the answer choices are: A) Sets the ATC testing workflow sequences; choice B) Correlates ATC user needs to requirements; choice C) Contains only ATC test cases; and the final choice, D) Traces ATC requirements to ATC test cases. So what is the primary purpose of the RTCTM?

**Dave Miller:** The correct answer is D; that's the document that identifies the test cases that will be used to verify each requirement with the test procedures. So it doesn't set the testing workflow. The testing workflow is part of the test plan. We talked about that earlier. The User Needs to Requirements are part of the NRTM, not the RTCTM. If you just go back a few slides, that's where we had our examples of those two. An incorrect answer, C, is "Contains only test cases." It contains the test cases, but it also has the inputs and expected outputs. So the test case—it really has parts of the test documentation, includes the cases, what inputs are going to be applied, and what the expected outcomes are.

**Dave Miller:** We're going to move on to our final learning objective here, which is: Describe the testing of an ATC using sample test documentation. So what we're going to do here is kind of through one. We're

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

going to show testing documentation, the contents of the lifecycle, and show some of the examples in a real test case.

**Dave Miller:** We're going to kind of walk through an example ATC test plan, test design, test cases, and procedures using kind of a case study that we kind of laid out in the beginning here. We're going to kind of discuss consequences of testing boundary conditions and maybe get into some test tools.

**Dave Miller:** This should be familiar if you've taken some of the other testing modules, like the ramp metering and some of the other ones, but this shows—the graphic shows at the top—we start out with the test plan, level test plan, and then below that we do a test design, test cases, and test procedures. The testing documentation for different projects is driven by the procurement configuration. We talked about that: the procurement configuration, as a refresher here, is ATC has a core, always comes with it—has to have an engine board, has to have Linux, has to have a power supply, has to have a host board to plug all of that into. In addition to that, there's options: What type of cabinet do I have? What type of housing do I have? Rack mount? Shelf mount? You put all that together, shop through the different variations and options, you come up with a procurement configuration. Like we showed before, the one with the example used is a TS2 type 2 with ABC connectors, but you could have a configuration for a 2070 version. The documentation is based on IEEE standard 829 format—we talked about that, so it's familiar—and test cases are carried out with one or more test procedures. So, a quick example of that, we did have—let's say one test case was we wanted to test the range of the input voltage. Say it's supposed to run from 100 volts to 135 volts; we want to test that. Another test case is what frequency does it run on, or what's the power interruption. So this one might be two or three independent test cases. A third one might be what's the power interruption—this one's the frequency, this is the voltage range. You could have one test procedure do all three of those. So there's always one test case for one thing you're testing, but you can have a procedure that does multiples. Usually that's done because the test procedure might involve putting the controller on the bench, getting a variable power supply to do power interruption at variable frequency. The test setup could do all three of them—line up three different test procedures. If we're going to use the test same equipment, set it up one time to run the test, you might as well, as you're set up, run all three of the test cases while you're at it.

**Dave Miller:** Let's review what should be in a test plan. This is—it's a document that clearly describes the scope—what will be tested and what will not be—written in a standard format, using a standard definition of terms. Test plan includes a description of the test approach, high-level overview of how it will be tested, and then in addition the test plan lists the resources to conduct the test, the people required, skill sets, test equipment, samples of material, how many units do you need to test, and others. And it always includes high-level schedule, describes the scope, approach, and describe resources. So the plan identifies the items to be tested, the features to be tested, and a list of testing tasks, and identifies the risks. So this would be what would be shown if you looked in IEEE 829. There's an introduction section that details the level testing, and it's only done in a familiar format.

**Dave Miller:** Part of review. Here we're saying we'll begin developing the test design that specifies the details of the approach. As we learned in the prior slide, we always verify what its mandatory objects are, the core: engine board, power supply, host module—and the underlying Linux software operating system and the drivers, and that's all in ATC 5201 Appendix A and B. And then beyond that, the test design list verifies the optional objects, cabinet style, inputs, outputs, mechanical housing, rack or shelf mount. Finally it must verify the inclusion of the board support package. So we want to make sure all of that is intact when we write our plan. This is verify mandatory, optional, and what's required and we talked about the board support package so that we can be able to put our software on more than one manufacturer's ATC.

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

**Dave Miller:** This is what should be in the test plan. Test design—that specifies the details of the approach, the features, and the requirements. Then we verify the mandatory elements, optional elements, make sure that we have all the software that's supposed to come with it per the standard, and verify the board support package is there. And then the fitness test, which is sort of something—kind of a term we coined here for the diagnostic acceptance test. So this would be something that you would run to qualify your units, if you wanted to put in a qualified products list. It's a piece of source code that could run and just verify that the function of all the elements is there. We want to verify that the manufacturer delivered a board support package that we compile and load. So one of your tests may be that you get this self-test source code, you use the board support package to compile it, link it and load it, and verify that the board support package works, the test software that came with it. If it comes with it, it compiles and loads, and environmental compliance. So we want to verify environmental compliance. If we have the lab in-house, we can create a test plan to run that entire Section 8 of the standard, or we can just require that the manufacturers supply a copy of an independent laboratory test per Section 8.

**Dave Miller:** Okay, the testing process. So here we're back to the test plan, the test design, test case, and the procedure. This guides the overall testing during and after results are recorded and evaluated. So this is sort of the input and the output. So we have our test plan, we have our test design, we have our test cases, we have our procedures. We install them on the ATC shown here. We have our test equipment and our test environment that runs the test; we get the test results; we get the logs and anomaly reports. So this is what's done—documentation done ahead of time. This is what you get out of it as a report at the end to see that you comply with the testing.

**Dave Miller:** We're going to end up here with a case study.

**Dave Miller:** Let's say we want to try to apply the process and write a test plan to verify an ATC to 5201 v06. Let's say that this is—we're calling it the City of Midsized. We're kind of using the TS2 type 2 because there's a number of different TS2-style cabinets, whereas the 332s are more straightforward with the 104-pin connectors. So we're kind of taking the more difficult case here. Our approach to the ATC test plan is three steps. This will make it easier to explain the outline—a good test plan on IEEE format—and we'll tailor this to the available ATC standard. The first thing we'll do is we know that we have these cabinets on the street. We may be taking out legacy NEMA controllers. We have to first have a configuration. You always have to have the core objects to make sure our software is going to run, and we have our options to make sure it fits our local conditions of—we're not getting new cabinets, we're going to reuse the old ones. Next we prepare a test plan. It's a good idea here that—what we're suggesting is that you might want to prepare the test plan and include it in the procurement specification. I know that that creates a lot of extra work up front, but that makes it completely unambiguous that if I get the procurement specification, I want to do a bid, it'd be very nice from my standpoint that, "Here's the test plan." So make sure that when you're bidding this, that you take into consideration you will be required to run this test, you'll be required to show us the results, and show that it meets our expected results.

**Dave Miller:** So let's start with the cabinet retrofit. So again we have this existing cabinet. We're going to take out the NEMA controller, put in the ATC. It has to be compatible with the existing signal control software application. We already have the license from a company we see, or some company that maybe not necessarily manufacturers ATCs but they develop software for ATCs. And we have a front panel text display, so we want to have a text display of keys and data entry. And we want to have Ethernet IP communications—we don't really have serial modems anymore. This is going on a new corridor. We've had to get a fiber cellular back haul. We're kind of abandoning lines or maybe it's—out in a new subdivision we never put it in there to start with. It's always had fiber buried, so we're not really interested in serial modems anymore. And then finally the project user needs of each stakeholder and the



## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

traceability matrix are included. So this is what we would start with, all of this. We have to use this to configure it, we have to make sure that in the end that you can own it and operator are going to be happy with the result.

**Dave Miller:** So the way we do this is step by step. First we'd examine and determine: configure an ATC from the core and options. The next thing we would do is prepare a testing plan—a test design that includes that configuration. And the test plan itself is made part of the ATC documentation.

**Dave Miller:** Let's determine what needs to be tested. This is the graphic we had in the beginning. From what we had from our user needs and our requirements, we have to have the core—that has to be in our procurement. We have to have the Ethernet ports, because they're part of the standard. It has to have the USB 2.1. We are going to add the screen circle around this that says we are going to use the 8-line display; so if we use it, it has to conform to the standard. I'm going to have the field I/O—I know it's optional. I could use a 104-pin connector if it's a rack mount, but this is hitting legacy NEMA cabinets. I don't need the communications interface. We're going to X that out. And then off to the left here, housing is optional; we're going to pick a shelf-mount housing. So this is really the steps you go through. You just take this diagram and configure it to meet your local needs.

**Dave Miller:** So we have to verify that we have the core, we have the Ethernet ports, we have the interface front panel, and we have the shelf-mount housing.

**Dave Miller:** Mandatory objects to be tested. Engine board interchangeable; the host board, power supply—we've talked about that. Operating system, board support package, and diagnostic and acceptance test source code—we might want to have the manufacturers supply that.

**Dave Miller:** So what we would do is we would make this NRTM. It's user need compliant to the standard, so we would list out these functional requirements. We'd number them. We'd list the requirement by name, and then we would say conformance means that, yes, we do conform to it. It's mandatory—it's not an option. And support—yes, the standard supports it. I can go into the standard and it will show me a section that describes that. So you go through this one by one, like that.

**Dave Miller:** So then moving on, let's look at this. This column means that each line is mandatory. Support means that each must be tested. To comply to the standard, conform to the standard, you have to—those are mandatory, you have to test all of those.

**Dave Miller:** And then moving on to the optional objects. So this is—okay, the standard describes lots of different options; we're going to pick and choose. We're going to pick the field I/O for TS2, and on like we configured at the 8-line display. We're not going to need a communications module for a modem. What this says is these are options: the standard does support it. So you don't have to get it, but you can look in the standard and you can find it.

**Dave Miller:** This is going on with some more NEMA TS2-type cabinet signal control source code for the board support package—that's needed to support compiling the linking and loading, and again the front panel. Supports in the standard, conformance, optional—those are the ones we've selected. But they all have to be tested: they're optional, but I've included them in my configuration; therefore I have to test them.

**Dave Miller:** And then finally we end up with this Requirements to Test Case to Test Procedure. So functional requirement: "The Device Under Test shall not restart when subjected to a power interruption of less than half a second." First test case: let's go ahead and interrupt power for less than a half a second. Verify that it's unaffected; verify that there's no power-down signal. Next, restart operation: power failure of more than a second. It must restart; it's expected to restart. Verify that the power-down signal did

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

occur, so that the software knew the power was going away. Functional requirement 2: shall maintain time of day. Test case 3: vary the power frequency; vary it from, say, 57 to 63 Hertz. Interrupt 500 milliseconds. Turn it off and back on over an hour. So some different things we can test here to make sure that it maintains the accuracy.

**Dave Miller:** So all requirements from the NRTM now are entered into the RTCTM; this is to do the traceability. For every functional requirement, we have a test case, we reference the standard right down the line; we connect them all together so that we know we've done everything. Once we have accepted this, we turn on production, they all come in, it's all going to work with the software I have. So a test case will be carried out in the testing process.

**Dave Miller:** This is what it looks like; this is a little example. So in our case, we decided that we were going to have a front panel. The front panel has to be interoperable among manufacturers. If you look in the standard, it has these key codes. So if I press the zero key, this is the ASCII code that comes out. It's an X30—a 1, a 2, a 3—you get all these different codes out. So no matter who I purchase my ATC from, those codes will always be the same. The front panels will—software will interpret each one of those keys the same among different software vendors and hardware manufacturers. So this is what's expected. If I press this key—if I press the V key—I expect to code out of it, a 42. There's a lot of things to test here, and there's not a lot of time. You have limited resources and budget. So what we do is negative testing. We would put in a positive test, as I push the button and I would expect that. A negative is to do something that's wrong. Simulate a button that's not on here. How does it respond to it? An invalid input. So you want to do something that's expected—you want to do something that's unexpected. Errors are okay, but it shouldn't make the machine do something unexpected. It should give you an error code or it should not respond to it. So positive testing is expected results; negative testing is give it something unexpected and see how it handles it; make sure it has safe and reliable handling of something that's unexpected.

**Dave Miller:** Boundary conditions. We're limited by resources. Say we have to test over the whole voltage range. It's hard to run the test at every single voltage, so it's better to say that it's supposed to run at 95 volt AC—it's supposed to run at 135, for example, if that was what it says—or 100 volts. What should we do instead of trying to test every possible combination that you just don't have time or resources for? Test it just below what the standard says, test it just above what the standard says are the limits, and test exactly on. Record the results. So if the boundary is valid, it should process successfully and respond accordingly. If it doesn't, you should get error conditions. It should remain in normal operation, but it shouldn't lose communication; it shouldn't do strange things. So you try to apply your resources in a sensible fashion to get the most value out of it. Instead of going through every possible combination, find the limits and show the standards, test just above and below, and test at random samples across each one, and record the results and see whether it's expected.

**Dave Miller:** This is a boundary, so what we would do is—what we're trying to do here is power fail. The standard says unaffected between zero and 475 milliseconds interruptions, and it's supposed to restart when it's above 550. Instead of trying to interrupt every possible millisecond, I do it just below, just above. So I would test it at 550, 750, and 1000, and write down the expected results. You can do that at room temperature—power interruption, use an oscilloscope—and then I'd repeat it again at like 100 volts at 135. If you tried to get every possible combination of input voltage and power interruption, you just don't have the practical resources to do that, so you would do that by boundaries. You'd look in the standard and see what's supposed to happen at certain boundaries; go above and below, and maybe do a combination of two or three different combinations of high voltage, low voltage, with different power interruptions, and then record the results.

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

**Dave Miller:** This would be like my boundary condition. This would be my test log. So I'd say at service voltage 100, 475 milliseconds, I would expect it to be unaffected, should run normally. Write down what's the actual. You go down through all of these. Service voltage 100 volts AC, 1 second interruption, I would expect it to restart. 135—so you'd go through. This would be a good example of boundaries. Instead of going through thousands of different combinations, just do these and you would expect these results. Maybe sample a few in the middle, and that should be the best use of your time and resources, instead of trying to do every possible combination.

**Dave Miller:** Finally we're going to end up with testing tools. This may seem daunting, and it is. There's lots to test. But the diagnostic acceptance test is not really a required deliverable for compliance to the standard. It'll talk about—in Section 8—what is a diagnostic acceptance test that you would run to accept the unit in. You probably wouldn't want to run those tests all the time, but it might be a good idea since manufacturers do this for a living. Pretty much all of them have some sort of—something they use in their manufacturing line. It's a good idea to at least talk to them about what you're testing at the end of your assembly line—maybe a piece of software—before you put it in a box. "Can I license that? And I'm not going to distribute it. I'm just going to use it in my shop when I receive it, to see if I get the same result, make sure nothing was damaged in shipping," that sort of thing. So if they delivered a source code, you can say, "If you'd be so kind as to give me the source code—I'm not going to distribute it—just for the self-test. I'm going to compile it and load it using your board support package, and I'm going to write a set of instructions on how to do that so my software supplier later on can do the same thing." This is not required by the standard; but it's a good idea, since the manufacturers are doing this sort of a test anyway, to ask if you could get it just for your own internal testing, not for distribution outside of your agency. So DAT performs a self-test. So this would do a lot of the automated parts. It would make sure your inputs work, make sure your outputs work. You can turn the voltage down, see if it still operates. You can run power interruption tests. And the nice thing about that is in the manufacturing process, when something does go wrong, they usually have somebody that records error logs so you can see that, "Well, this one didn't pass, but here's a clear list of why it didn't. I put this output out—these four outputs out—I only got three of them back. There must be one output that's non-functional" —that sort of thing. And you can ask about custom loopback cables. A lot of that stuff's out there because it has to be done in manufacturing. It's a good idea if you can use the same sort of automated testing and leverage what's already been done.

**Dave Miller:** That brings us to the end of this one. This is kind of at the end of the PCB training we've gotten. These are the related ones. So this is what—others that are very similar to the testing ones. So like T309 is ramp meter; NTCIP 1209—you'll see a lot of the same concepts here; like in T309, you'll see references to the same IEEE 829. The documentation is the same. You'll see the same matrices that we went through. So this kind of brings us to the end of this module.

**Dave Miller:** We're going to have one more activity here.

**Dave Miller:** So we're going to ask—the question here is: Manufacturer will have a single ATC that conforms to all versions of the standard. So if I'm a manufacturer and I say, "My ATC conforms to 5.2 and it conforms to 6.0 and 6.1, and all the future standards," is that really possible or is there a difference between them? So is this a true statement or not?

**Dave Miller:** The answer is false, and we talked about this. There are some subtle differences between the different versions of the standard, going from 5.2b to the 6. versions that are being adopted right now. We said it's good to ask for a conformance statement to say which version you conform to. If there's differences that would cause interoperability, supply a document for that. So the few conflicting sections, it's always good to get a conformance statement to make sure that you know which one—this is valuable

## Module 58: Applying Your Test Plan to the Advanced Transportation Controller Based on the ATC 5201 Standard v06

to the software developers, so that the developer knows if I'm going to go out and look at that SPI input or I'm going to look at this other thing, I know there's differences. Instead of me spending my time trying to figure out which it is, just give me a list of which version to conform to and in which areas, and that will speed things along quite a bit for the software validation.

**Dave Miller:** So to summarize, what we've done is—the first learning objective is we identified the key elements of the standard that relate to testing documentation. We described within the context of the system engineering lifecycle the role of the test plan, why do we have a test plan, and what tests need to be undertaken. We applied testing documentation. We went through an example based on the standard, which parts of the standard are referenced, how do you find those sections, how do you make sure that they're in there, how you trace it to make sure you conform to all of them. And then finally we went through a short ATC testing example using our power supply interruption, power fail/restart. So we sort of showed that. That's sort of a simple example. You may not have to do that from your agency standpoint if you get that as part of the independent lab report, but that's an example of what it should look like when you get it and make sure that it's working correctly.

**Dave Miller:** At this point we've completed all of the ATC curriculum. So we started out with A307a, the user needs for ATC based on the v06 standard; A307b, requirements—we went through the requirements in that module—and then finally here at T307, we applied our test plan to the ATC standard based on v06 of the standard. So once you've gone through this, you should be able to figure out how to get your user needs from the end owner-operators, take those needs and turn them into a requirements matrix, so it's clear what we need to do. Then in this module we described how to take the standard and take the mandatory and optionals to configure it so you get a final configuration; and then how to make sure to put all that together into the matrix to apply the test plan so that the outcome—we can make sure we conform to the standard. We also can do test cases and procedures to make sure we meet the needs. So when it comes out of the box, installed on the street, it runs the software, the plugs match, the pins will match, and it's interoperable among manufacturers. So 307a, 307b, and then finally this one. So that completes the series.

**Dave Miller:** We'd like to thank you for completing this module, and please use the feedback link below at the end of the module. We really want to hear your thoughts and comments about how valuable this training was and any suggestions and improvements. We thank you a lot for participating.