

Nicola Tavares: Welcome to the ITS standards training.

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. 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 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. Please help us make even more improvements to our training modules through the evaluation process. We look forward to hearing your comments. And thank, again, for participating and we hope you find this module helpful.

Nicola Tavares: Throughout the presentation, this activity slide will appear indicating there is a multiple choice pop quiz following this slide. The presentation lecturer will pause at each quiz section to allow you to use your computer mouse to select your answer. There's only one correct answer. Selecting the submit button will record your answer and the clear button will remove your answer if you wish to select another answer. You will receive instant feedback on your answer choice. Please help us make even more improvements to our training modules by completing the post course feedback form. This module is A207B Building an ITS Infrastructure based on the Advanced Transportation Controller 5201 Standard part two of two. Your instructor, Ralph Boaz, is president of Pillar Consulting Inc. He is a transportation, engineering and marketing consultant. He was formerly vice president of Econolite Control Products and transportation section manager for Boss System Engineering. He has been a project manager and a consultant to the ATC and NTCIP standards program.

Ralph Boaz: Hello. It's my pleasure to be your instructor today. Let's take a look at our target audience. You can see we have a broad spectrum of people and capabilities and

knowledge. So please bear with us as we go through some technical details. Hopefully, we have given you enough background information from the previous course to help you understand what's going in here. We also hope that those of you who maybe are used to traditional ITS systems that you really catch a vision for what we're showing you today. Here's our recommended prerequisites. This is typical of our course work. We tend to have courses that run with system engineering content and those without system engineering content. This is kind of you'll find out that this kind of has a combination of that as we go through these ATC modules. I would like to emphasize that the course A207A Building an ITS Infrastructure based on the 5201 Standard part one of two, this is part two of two. The part one we set up kind of the very, very basics of transportation field cabinet systems and discussed the evolution of those cabinet systems. And the purpose of that was to set the stage for this course so that users less familiar with field equipment understand the why for the architectures we're presenting here. Here's just a graphic illustration of our path to get to this course. And you can take a little look on that on your own. So let's talk about the learning objectives. There are the learning objectives for both A207A and A207B. One was explain the purpose of the ATC family of standards. Two was to identify the basic components and operations of transportation field cabinet systems. Three was to identify the features of the ATC 5201 standard. And in four we described the ATC 5201 architecture. And in learning objective five, we describe how the ATC 5201 standard works with other ITS standards. And six we helped you specify ATC equipment for system and equipment procurements. And as I said previously that we have covered the first couple of these learning objectives in A207A to lay a foundation for what we're going to talk about today. And we'll be concentrating on learning objectives three through six. Learning objective three identify the features of the ATC 5201 advanced transportation controller standard. We'll talk about the problem with traditional transportation controller architectures. We'll talk about the structure of the ATC 5201 standard and we'll talk about features for ATC controller units. If you took a look at our history and if you went, if you recall, the evolution of ATC equipment-- sorry-- of transportation control equipment from A207A you'll see that there's really been a very modest advance relative to other technologies used in our everyday life. And so there's just modest advancement in the use of these technologies in a traffic management world. So we call that the traditional capability curve. But as we now we have NTCIP communications, those of you who have taken some of those classes. We're also talking about adaptive control. Adaptive control has been around for a little while but it continues to advance in what we're trying to do. And now we're talking about things like connected vehicles where the vehicles are connected to the infrastructure. We call that V to I. And so we have an application demand curve that is very steep so that there's pressure, if you will, because of this technology gap that exists. We have this low kind of traditional capability that's lower and an application demand curve that's kind of skyrocketing here. And now we're talking about things like big data and data sharing and all of these things and again, that all puts pressure on the infrastructure. Now, if you were to take a look at just the comparison of performance to illustrate what I just said in the previous slide, you

took a look at a model 170 controller deployed in 1982 it would be a 0.2 MIP processor. And MIP stands for millions of instructions per second. If we take a look at the model 27E standard that was back in 1998 we see that it is a 4.5 MIPS processor. And you might say wow that's pretty good and that's about 26 times more powerful than a 170. However, when we're talking about the ATC standards both version 5 and 6, we're talking about a specification that says that minimum is going to be a 16 MIPS processor. But commonly available today, out there, vendors are selling 500 MIPS processors. And so we're talking something that's 2500 times more powerful than model 170, and most of the model 170s in use today. There have been many uses proposed for ATC controller units. Of course, there's traffic signal control and traffic management and transit and light rail priority and we're also talking about things like emergency management and lane use and access control and data collection systems and data distribution. And then, of course, I all ready mentioned the connected vehicle applications that have to do with safety and even eco driving. So the purpose of the ATC family of standards is really to provide a general purpose field computing platform for transportation applications. And we want it to be open architecture. That means any manufacturer; developer should be able to build products for the standards. It's modular. That means it has an internal structure so that there's separation of functions and this reduces time for configurations and it helps with maintenance and things like that. It must be multi process and multi-application. Multi process means it can do multiple things at the same time. And multiple applications means it can be used for different purposes so we have something you can do different jobs at the same time. We want to be able to grow with technology. Traditional standards for transportation control equipment have sometimes been obsolete by the time they come out, by the time they're approved and gone through the entire process. So we don't want that to happen. What we want to be able to do is have this standard come out and then be able to grow with technology without necessarily having to go back and immediately start changing it. We want to be able to upgrade legacy transportation field cabinet systems or TFCSs. We want to be able to have a contemporary performance and capabilities but still using maybe the cabinet infrastructure that's all ready in place like our 332 type cabinets from the 170s and the NEMA TS1 and TS2 cabinets, et cetera. This is structure of the ATC 5201 standard. You can go down through here, through this section. This course will focus a lot on the engine board details, that's aspects of section four. And then also we'll be focusing in on appendix A, we'll be talking about the Linux operating system and we'll also be talking about device driver interfaces, et cetera. So you can go through the standard but the key components here that we're focusing in on this class will be the sections I just mentioned. The remainder of the standard is really all about the environment and being able to connect to the cabinet systems, et cetera. So those physical types of things or environmental types of things. So here's just a reminder of our TFCS's that are out there. We have NEMA TS1. We have the model 33X type of cabinets or maybe some people say 3XX type cabinets. Both of these were parallel. Recall those are parallel communications. They have discrete wiring throughout the cabinet. And then we have

NEMA TS2 type 1 cabinets and the ITS cabinet version 1. And these cabinets used serial communications to communicate to the different components of the cabinet. These cabinets tend to have more capability in being able to work with more sophisticated single monitoring systems. So let's talk about the features for ATC controller units. The major operational features are manage or configure applications. Secondly is manage external devices. Thirdly is facilitate ease of maintenance in future hardware, software updates. All of the rest of the operational features listed in the standard are organized into these three groups. Let's talk about managing, configuring controller applications. The first one is install/update application software quickly and efficiently. And we want that to be able to be done by users, the end users. We want to be able to install the operating system quickly and efficiently. Now, this may be-- we're talking about the Linux operating system. So in the first one we're talking about application programs and in the second one we're talking about the operating system. We need the ability to manage the calendar and clock functions. Our business here in the transportation world is all about timing. And so those capabilities need to be there. And then we need to be able to configure and verify parameters for particular local applications. The intent here, this need, if you will, or this set of features has to do with being able to input things on a screen or a keyboard and have communications access to the controller from maybe remote places. The next one here is upload/download data blocks as needed to transfer files and accommodate bulk transfers of new application databases. Many times the applications have a database that needs to be uploaded to the controller so they can operate properly. Or sometimes they're configured at a remote site and then uploaded or changed. And so that's there to accommodate that need. And we talked monitoring and verifying present application status. This is to view remotely or locally what's going on with the controller system or the applications running on them. We want to be able to allow operator control of application execution. And this feature allows the operator to manage the starting, stopping, scheduling of one or more applications on the ATC. And then the last one here is facilitate the long-term storage of data for logging and other data storage applications. So let's talk about managing external devices. So we're talking about managing control of variety of external field devices. There's different kinds of equipment out in the field, most obviously are signal heads. But we want to be able to do this remotely. We want to be able to do it locally. And we want the controller to be able to operate these external devices in an unattended fashion. In other words, there's nobody there. It's acting independently from another system or human. And we also want to monitor the output and status of a variety of external field devices. We're not trying to-- we don't try to restrict what that might be but it allows the controller to monitor these device outputs and status and then to use that status for local configuration failure diagnostics, logging et cetera. So the most pressing need for this would be to monitor the power that's going out to the traffic displays or traffic signal heads. But they could be used in other fashions also. This last major area for features, for ATC controller units is to facilitate the ease and maintenance of future hardware and software updates. So when we're talking about the first one here maintaining and updating controller hardware, the

main problem in this case has been processor obsolescence. It hasn't been the method in which the controller interacts with the rest of the cabinet system, but the processor itself has become obsolete. And so that's really what we're talking about there. And we're talking about maintaining, updating, controller software. By maintenance here, we mean the standards referring to the maintenance performed by a technician or engineer in keeping application software operating and keeping the operating system up to date. And then we want to be able to support diagnostics. We want to be able to do things like accessing the controller and taking a look at operating system parameters via some specific serial port or through an Ethernet port. So now we have an activity. This is your opportunity to see what you've learned. Which of the following is not a purpose of the ATC standards program? Your choices are A, general purpose field computing platform. B, grow with technology. C, open architecture. Or D, compact. Please make your selection. Let's review our answers. If you said, D, compact you are correct. Being compact is not a purpose of the ATC standards program. However, the module architecture of the ATC 5201 standard is conducive to creating units of various sizes. If you said, A, your answer was incorrect because defining a general purpose field computing platform is a goal for the ATC standards program. And the same is true for being able to grow with technology. That is a goal for the program. And being open architecture that is a goal of the program. We want manufacturers and developers to be able to create-- we want to invite them to create solutions for the transportation industry by creating open standards that anybody can bill to. Some of the other goals that haven't been listed here were multi process and multi-application, be modular and to be able to upgrade legacy TFCSs. We have another question here for you, which of the following is not a major feature of ATC controller units? Previously, we talked about the program, now we're talking about the features of the ATC controller units. Your answer choices are A, manage/configure applications. B, Windows operating system. C, manage external devices. Or D, facilitate ease of maintenance in future hardware, software updates. Please make your selection. Let's review our answers, which of the following is not a major feature of ATC controller units. The correct answer was B, a Windows operating system is not a major feature of ATC controller units. A Linux operating system is used. If you said, manage/configure application this is incorrect because this is a major feature of the ATC controller. Manage external devices, this is also incorrect because as we discussed we want to be able to manage other devices in the field. And then if you said D, this was incorrect because we want to be able to maintain controller units and make updates and this a major feature of ATC controllers. To summarize our learning objective here in identifying the features of the ATC 5201 standard we discussed the problem with traditional transportation controller architectures. We discussed the structure of the 5201 standard. And we talked about the features for ATC controller units. Now, questions have come in previous iterations of this class and I'd just like to add a question here. This question goes to that bar chart that we discussed the capabilities of the various types of traditional versus the ATC based controllers. And the question was, aren't there model 2070 controllers with faster processors? And yes, there are. There's been various

vendors and some specs that have specified some of these ATC concepts but none of those specs or 2070s with the faster processor actually conformed to ATC 5201. Now, you can build controllers that are ATC 5201 that will work in the same kinds of field cabinet systems that 2070s work in and we'll discuss that in a minute. Now, we will take a look at learning objective four describe the ATC 5201 architecture. We'll discuss key elements of the ATC 5201 architecture. We'll describe how the standard and the ATC 5401 standard work together. And then we'll discuss ATC software portability, compatibility and interchangeability. Key elements of the 5201 architecture is that it's based on an engine board concept. This concept allows the computational capability of the controller to grow with technology without removing necessarily the field cabinet system or even possibly the controller to do that. It uses a Linux operating system which goes along with the open architecture concept. It's open source. It's multi process, it's multi-application. And it can be downloaded from the Internet. And it can be run without runtime licenses. Other elements of the architecture are that the mechanical requirements in it unlike some controller specifications, the mechanical requirements are for the physical interfaces only. There aren't specifics where there has to be a thumb screw or a particular size of the unit, et cetera. Another feature is that it works with all of the major transportation field cabinet systems. This is very important because of the expense of taking out that equipment. We want to be able to get this modern capability and using our traditional infrastructure. And then we want source code portability for application software. And we're going to talk about what we call the application programming interface standard, ATC 5401 in a minute. If you don't have that application programming interface we can still achieve portability meaning that we can get a program running on one vendor's machine, running on another vendor's machine. But typically it would only be one application program at a time. This is a graphic representation of the ATC engine board concept. The dark bars at the top and the bottom represent connectors and then there's various parts. We have two Ethernets. We have a USB capability. We have a real time clock. Memory. We have the CPU running the Linux operating system. And we have numerous serial I/O capabilities. Here, we'll talk specifically about the I/O on the engine board. They are the interfaces, I should say. So there's a console port and that's used for diagnostic and maintenance purposes and access to the Linux operating system. Typically your operational users won't use that port. There are serial ports. There are general purpose, serial port one, two and three. There's a serial port built in that's dedicated for use of the front panel and keyboard. Again, these interfaces are all built into the engine board. There's field I/O ports that are dedicated for communicating to field I/O devices and buses, et cetera, so we can communicate with the-- when I say field I/O the inputs and the outputs of the transportation field cabinet system. There's a USB memory device. This is not your typical USB usage where it can just run about anything out there in consumer electronics. This one is dedicated for memory devices only. There's a data key which is a Flash memory type of memory device and some applications that uses devices to have certain parameters stored there. Then we have two Ethernet ports and these have been used

and added switches to them, Ethernet switches so that they can actually come out as multiple ports out on the physical device. This is actually a picture of one vendor's engine board. And we'll just identify the parts, again. Every engine board will look differently but in this case, here's the CPU. There's three kinds of RAM. There's a DRAM or dynamic random access memory and that's typically the kind of RAM you would think you would have on your computer. We have a Flash memory and that's memory that retains data without power. Think of your memory cards, USB Flash drives and install straight drives. And then we have SRAM which stands for static random access memory. And that's fast access memory that retains data as long as power supplied. And you can think of the cache memory of a personal computer. Here we have the real time clock. And here's an Ethernet chip added in. And then here's a serial IO, the CPU, serial I/O built into it but then we have an additional chip here for the extra interfaces that we have. Then we have the USB. We have chips here controlling the USB interface. We have a layered architecture when we're talking about the ATC. And we call the-- and I'll just go through this here. If you still look at the bottom, that's our engine board and it's turned flat. So we're looking at side few of this. And we call that the hardware layer. And then on top of that is the definition of the Linux operating system and the device drivers that's connecting the operating system and interfaces to the board itself. And we call that the ATC board support package layer or BSP. So the engine board and the BSP that's defined by the ATC 5201 standard. So what that does then that allows application software from various vendors to operate on this ATC engine board and allows operational users then to interact with it. Now, this is something that is really exciting. We can take this engine board that we just described and we can actually make different architectures that use this engine board. So you can get an engine board from various vendors and create, let's say, a host module for a 2070 controller shown here. You could build one there and we could also create a host module that's compatible with a NEMA controller and this is just an illustration of that. So you think of this hardware interchangeability and we're going to go later into the software interchangeability. This hardware upgrade capability is done because the engine boards have identical pinout. Because all of the engine boards have the same pinout, remember I showed you those broad set of pins on our graphic previously, future boards may be able to plug into existing host boards. It's really important that this is defined and is fixed so that we can have this upgrade capability in the future. This allows a computational capability that can grow with technology, in other words new engine boards can have more memory on them. They can have faster processors and still work in our controllers. And then, again, by using this it's possible to upgrade without changes to the rest of the controller. And I want to emphasize that we're talking about a hardware upgrade capability. When you upgrade software there may be other things involved, for instance, timing and we'll discuss some of that going forward. And we'll discuss that even further when we talk about when we go to the next module A208. Here are examples of ATC conforming controllers to ATC 5201 and from various vendors. These work in both NEMA TS1 TS2 and would work in an ITS cabinet, in a 3X cabinet. And this is exactly what we were

trying to accomplish. All of these have the engine board have Linux running inside them. There's other companies coming with their version of ATCs. And, again, they're available for all of the TFCS architecture out there. Let's have another activity, test of our knowledge. So which of the following is critical to being able to replace an engine board with a more powerful engine board in the future? Your answer choices are A, identical pinout. B, new host module. C, same processor family. Or D, same engine board manufacture. Please make your selection. Let's review our answers, which of the following is critical to being able to replace an engine board with a more powerful engine board in the future? If you said A, identical pinout you were correct. This is essential to allow new boards to replace the previous ones. If you said new host modules that's incorrect because generally the host module of the existing controller can remain the same for a particular controller. But over time, any board needs updating, so it's possible a manufacturer might want to update a host module as well. But generally that's not the critical point for using the engine board. If you said, C, incorrect and generally the Linux board support package allows any qualified processor to be used on an engine board. In other words, companies that come out with new chips, one of the first things they want to do is make sure Linux runs on their chips because that's how they sell their chips to various computer manufacturers. So that job is being done. And so we reap the benefits of that. So the point here is we don't want to restrict ourselves to a particular processor or even a family of processors. And if you said D, the same engine board and manufacture that's incorrect. Again, we have an open architecture and we want to be able to allow other vendors to create engine boards that could be replacements for the ones we have bought in the past. So let's describe how ATC 5201 and ATC 5401 standards work together. And I said previously the ATC 5401 is the application programming interface or API. The API provides a window-like interface to support the use of I'd say interfaces will be a window like user, interface to support the use of multiple application programs running concurrently on your controller. In other words, just like you have your browser on your computer running at the same time you're working on an Excel program and at the same time you're working on a Word document, you want to be able to use a mouse or key application to flip through those windows so that you can work with them. But they're actually all running simultaneously. And the other thing that the standard allows is that application software can now be written to operate on any ATC controller unit in any major field cabinet system. Third, we want to be able to have multiple application programs share the resources of the controller and the TFCS. Now, we all ready talked about sharing the resource being the front panel but this also means being able to share the resources of the input devices and also share the resources of the outputs and also to be able to share access to the real time clock. These are very important. So we're going to illustrate this a little bit here, this API concept. If you take a look at the left here, this is actually a block diagram or an ITS cabinet version one and we're not going to go through the parts of this. It's just if you recall there's a controller, there's inputs, there's outputs and there's monitoring capabilities and that's what's represented here. And there's a power distribution capability. Okay. So the controller has the front panel. And in

this front panel we show five programs running. We have a ramp meter program running. We have a signal program running. We have an emergency management program running. We have a data distribution program running and some sort of system checking type of program. So here this controller is running five different programs. In this example here, the star next to a single program says that's the default program. And what that is the program that is displayed when the controller is first started up and that is selectable by the user. So let's say we're trying to share the various output devices out in the field. Now, if we were running a traditional architecture, only one program would have access to these field devices, for instance, if you recall the outputs are controlled by what's called low switches and that's represented by these brave blocks with the examples showing the red, yellow, blue colors, although they're black and white so to speak. So what happens is ramp meter program will start up and it will reserve access to, in this case, two of the low switches, the single program-- the emergency management program when it comes up. It'll reserve access to a particular low switch maybe that controls access to an emergency gate or something like that. And then the signal program will take the remainder of the load switches. And in this way, this reserved access keeps the programs from stepping on each other so that you don't have two different application programs trying to control the same signal head which would be disastrous. But yet, at the same time these programs don't have to communicate with each other. The API controls its access. So now if we take a look at our layered architecture we see a similar picture with the engine board and the Linux board support package layer that we discussed. But now we have this layer of software. The ATC 5401 standard defines software that runs on ATC engine board. And it's likely that you will receive the software when you receive your controller and it will all ready reside on the engine board. So ATC 5201 defines the engine board and the board support package, and the ATC 5401 standard defines the API software. And now we have application software that will use the API when we're talking about the shared resources and the operational user will use the application programs to do their jobs as well as things like the front panel manager to interact with the various programs running on the control. Let's talk about the "ilities" [ph?], we call them portability, compatibility and interchangeability. So portability is the ease with which application software and data can be transferred from one application platform to another. It's also the ability to move with minimal changes, application software between computers. With the ATC controller we're really talking about source code portability. So what that means is that application writers will write their source code knowing that it's going to be ported at different devices and so they will use the Linux operating system calls, they will use the API software libraries. And then depending on the target processor of the engine board, they'll compile it for that process. For instance, if ATC engine board manufacturers is using a particular processor, then the application software is compiled for them and the same way for the other vendors. Now many times, the vendors themselves will be selling their own software, but we want to be able to open up this market so that third parties can do this. And you can imagine just think about something like Microsoft Word. You can buy a version of Word that operates on a

Macintosh and you can buy a version of Word that operates off a PC. And so that's done through compile and linking for the particular processors. Let's define compatibility, that's two or more systems or components performing their required functions while sharing their same environment. Then also the two components do not need to communicate with each other. So this is the environment that we just described. So we want to be able to run multiple applications concurrently. And here we show how that's done because of the ATC operating system in the API software. Now, let's talk about interchangeability. It's a little harder. The same functional and physical characteristics, so as to be equivalent in performance and durability. And you can think of that that might be subjective, right. People will have their opinions on whether it's equivalent in performance and durability. And then also it's the ability to exchange devices of the same type without alteration to the device or adjoining items. And adjustments are permitted, for instance, you change controller software where you may need to configure it, so that's okay. But, again, this is subjective. Interchangeability is in the eye of the beholder. An example of this might be that someone runs Internet Explorer and they also run Firefox and they also run Google Chrome. These are three Internet browsers on that you can get on a PC. And to certain people they're effectively interchangeable. They all do their main function is all to be able to have access to the Internet. However, if you talk to fans of any one of those products they'll say absolutely their product is better. It's none interchangeable with the others. So, again, this is just an illustration of interchangeability being a subjective matter. This is a graphical representation of just what we discussed. And we see here that application three is interchangeable with application four as long as they're doing the same type of application. Here we have another activity. Which of the following ATC controller resources is not shared or managed by API software? Your choices are A, the real time clock. B, the front panel. C, data key. D, field input and output. Please make your selection. Let's review our answers. Which of the following ATC controller resources is not shared or managed by the API software? If you answered C, you are correct. A data key is a memory device controlled by the BSP and not the API software. You recall the API software allows multiple programs to set the real time clock of the controller so A was an incorrect answer. If you said, B front panel, this is incorrect because the front panel is managed by the API so that application programs can share the front panel, allow users to switch between programs. And if you said, D, this was field I/O, this is incorrect because the field I/O is managed by the API software which allows programs to share the field I/O points simultaneously. Or I should say they should be reserved and used without problems. Summarizing our learning objective where we described the ATC 5201 architecture. We discussed the key elements of the architecture. We talked about how the 5201 and 5401 API standards work together. And we talked about software portability, compatibility and interchangeability. Now, we will discuss learning objective five, describe how the ATC 5201 standard works with other ITS standards. We'll discuss the various transportation field cabinet systems. And well also talk about ITS communications standards. As we've said ATC 5201 works with all of the major TFCSs out there including the model 3XX type cabinets, NEMA TS1, NEMA TS2,

NEMA TS2 type 2 and 1 cabinets I should say and then also ITS cabinets. Now, this is really important. The ATC 5201 standard has generally more rigorous environmental and testing requirements than the TFCS standards. And that's important because when you're wanting the unit to be able to operate within these other cabinet systems it needs to standup to the same environment that those cabinet standards have. So this is important. So the ATC 5201 generally has more rigorous environmental and testing requirements. Here's just an example on the left there's a controller on the second shelf there. That's an ATC unit operating in a NEMA TS2 type one cabinet. On the right, this is a model 332 cabinet and we have an ATC controller unit operating this cabinet on top there as a shelf mount. So ATC 5201 provides computational power and interfaces for ITS communications. It's been all ready shown that the typical 170 controller units require replacement hardware in order to utilize NTCIP ASC communications and that's covered in another module in this program. And then also that the ATC 5201 provides the capability to support multiple applications using different types of NTCIP communications simultaneously. In other words, the traffic controller is not just a ramp meter. It can be a ramp meter running the NTCIP ramp metering interface. At the same time, it's an actuated signal control running the ASC software and potentially running other applications such as the signal control and priority or other standards being used at the same time. And then ATC 5201 provides the interfaces in computational power for applications such as adaptive control and connected vehicle applications. Again, when some of these have all ready been identified as needing replacement or some sort of augmentation to be able to perform the connected vehicle signal phase and timing messaging where the signal phases and timing is being sent out to the vehicles that are coming down the boulevard. I want to point out that the API does not provide NTCIP software itself. That comes along with the application level depending on what the application needs to do. So it just provides this capability that applies to the Linux operating system and the API allow these multiple programs to run NTCIP simultaneously. So which of the following is a true statement. Your answer choices are A, API software provides NTCIP communication software for multiple applications. B, ATC 5201 allows multiple applications to use NTCIP communications simultaneously. C, ATC environmental requirements are generally the same as TFCS standards and specifications. Or D, most 170 controllers are suitable for NTCIP ASC communications. Please make your selection. All right, let's review our answers. Which of the following is a true statement? If you said, B, you were correct. The ATC 5201 standard and the Linux operating system supports multiple network addressing allowing multiple applications to use NTCIP communications at the same time. If you said A, this was incorrect. The API software does not provide NTCIP communications directly. Remember, we said it allows for it but those applications provide their NTCIP communications themselves. They're set to be at the application level. If you said C, that the environmental requirements are generally the same as the TFCS standards and specifications, this answer was incorrect because generally the ATC environmental requirements are generally more robust than the other standards and specs. And if you said, D, most 170 controllers are suitable for

ASC communications that was incorrect. It's been shown that for controllers to perform full functioning NTCIP ASC communications they will need some sort of augmentation. A summary of our learning objective here, describe how the ATC 5201 standard works with other ITS Standards. We discussed the transportation field cabinet systems. We talked about support for other ITS communications standards. Now, we'll move on to our last learning objective in this module, specify ATC equipment for system and equipment procurements. Here we'll discuss how the ATC 5201 standard fits in the system's lifecycle. We'll talk about developing migration strategies and mitigating deployment issues. And third, we'll create an example specification based on the ATC 5201 standard. We'll not get into all of the nitty gritty details but give you the information that you need to create a spec. Now, this particular learning objective will be a little bit longer, a little more complicated. So please bear with me and hopefully we'll bring everybody along here. So from our other PCB modules, you'll recognize the system's life cycle represented here by the V diagram. And generally speaking we have the concept side here, beginning of the concept of operations. And then we developed system requirements. And we have high level and detail design. Then we go into the software and hardware development and possibly field installations. And we talk about unit testing and subsystem verification and deployment and validation. And we see the left side of the V and the right side of the V work together because of the foundation we laid on the left side we can now do this verification validation on the right. And it goes on to operations and maintenance and exchanges and upgrades. And then eventually there's retirement and you can go on to replace the system. Well, how does ATC 5201 fit into this? So as we said many times ATC 5201 defines a general purpose computing platform for TFCS's. ATC controller units can be deployed as part of a center to field system project. The V diagram described a system and we're usually talking about a central system there. And so these units can be deployed as you get a new center to field system. The ATC controller units may be used for multiple C2F systems, center to field systems. For instance, maybe you have a different center to field system for emergency management. And you have another one for your traditional traffic management usage and there's just different applications used on the ATC unit that runs simultaneously. Or you may also have generations of center to field systems maybe after several years you want to change systems, central systems and you do that and the ATC unit is still there and operational. So also the ATC units may extend the life of a system. You can add maybe some features to the ATC unit that makes the central system more functional and preserves its use. And then procurements may be made in relation to a system deployment but it's more of a strategic system to use the ATC standards. And what I'm saying there is that generally you're not saying let's just try a unit here and let's try a unit there. I mean that happens in terms of when you're first deploying these things. But once you see the benefits of the ATC concept, then it becomes more of a strategic decision to use the ATC standards and you're not comparing one intersection to another in terms of, for instance, something like price or et cetera. So let's just illustrate what we just walked about here. So as part of a system an ATC controller may be tied to some sort of system

deployment. You might see them being deployed in the implementation and integration stages of the system. ATC units might be deployed to extend the life of a system. They might be in place for multiple center to field systems. But as we said it's more of a strategic plan for your agency or part of a regional architecture in that ATC's are deployed on a continual basis at any point of any of your system projects. Let's talk about migration strategies. We really want to emphasize this as to have a written plan and policy for deploying ATC equipment as part of a strategic plan. And then factors that may influence that plan will be hey I'm working with the existing TFCS's. I don't want to leap to new cabinet architecture so that may affect your plan. You choose to move to new architectures, that will affect your plan. You want to make the deployment coincide with the new projects that are coming up. You want to do it just part as a constant replacement, just part of your scheduled maintenance for your intersections. Here's a thought: many of the manufacturers or developers of software for traditional transportation controllers have a version of their software that will run on an ATC. So you can think about hey, I could possibly upgrade the hardware and still have the look and feel of the software I had previously. So that may be-- that may help you in migrating to ATC's. Or maybe you just want a whole new set of features and capabilities and things and so you'll upgrade your operational software. You may need to worry about compatibility with the existing or plan system software and you need to consider that they'll be some training for your operational staff. The idea is that they'll still be able to see some of their programs that they've always seen with the ATC. But there will be different things you have to learn such as using the front panel manager, et cetera. I just want to state that the front panel manager is part of the API software so it wouldn't necessarily come on an ATC you would buy today and we'll talk about that. So here's an update, ATC 5201 version 6 is currently being developed with publication projected for second quarter of 2014. And although the API standard, ATC 5401 version 2 has been published complete API software meeting the standard is not yet available. And starting this month, the USDOT, that's October of 2013, is starting a project to create a reference implementation of API software for public use. And it's projected in 2014 and 2015 at the latest for that-- for access to that. And the idea here is that then vendors and application developers can all, for instance, download from a specified website the APIs software that has been supported and developed by the industry. And that way we have the most reproducible effects of the API software and the application providers, you can have more confidence that their software will run on other ATC platforms. And this API reference _____ includes a validation suite that can be used to test API software on an engine board. We suggest that you reference ATC 5201 version 6.10 or later. This is because version six has many corrections to the predecessor ATC 5.2B. However, if you have to at this moment, if you have to reference a published standard you may still have to use ATC 5.2B. We suggest you establish a contractual commitment from engine board suppliers to provide operational API software for their engine board once the API, ROI is available. The reason we're saying this is you may be looking at a hardware procurement right now but the API reference limitation isn't available. All of what you're

doing is trying to establish that once it's available that your manufacturers will provide that. Establish a contractual commitment from application software providers to supply software compatible with the API once the API RI is available. And then you need to identify the tool chain used to create the application programs for the engine board. Now, what I'm-- actually, this should refer to as the board support package because it's important to know to be able to tell other application developers how do you compile and link something that will run on a particular engine board. So you will be working typically with the manufacturers on getting that information. You'll want to determine access to source code for the Linux environment. This may be something you negotiate with your manufacturers. You should require a test program from the manufacturers demonstrating that each of the interfaces is operational on the controller. Have a certification from a manufacturer that the ATC has been purchased and passed with testing under a load. And you might consider these kinds of testing to be done by a third party and have them show you a certificate that it was done by a third party. Prior to any purchase of application software, make sure that those application providers show you that their software runs on your controller. That makes sense but it's not like the PC world where you can go to a store and you're 99.9 percent sure that that software that you picked off the shelf will run on your PC. You want to make sure since we're a much smaller market, you want to make sure that the application providers prove it to you. And then we suggest that you buy support services with software in the case that the software becomes interoperable due to the changes in the configuration of the controller. Now, that you're running multiple applications these kinds of interactions are going to be-- may have unforeseen interactions. We're hoping to mitigate that through this API RI but your best bet is to buy support services for the software that you get. In all of these things we suggest that these mitigating points that you add them to your procurement spec. So here we have an activity, what is the best way to migrate to ATC equipment? Your choices are A, use existing TFCS's and replace older controllers with ATC controller units. B, use existing operational software that has a version suitable for ATC controller units. C, replace controller units as part of a regular scheduled maintenance. D, all of these items are equally good and it depends on the needs of the agency. Please select the best answer. So let's review our answers. What's the best way to migrate to ATC equipment? And if you said D, you were correct. All of the items we listed can be part of a migration strategy. And an agency must define their needs and goals to assess what is best. If you said, A use existing TFCS's to replace older controllers this is a good option for migrating and mitigating equipment, mitigating deployment issues. But there's multiple ones, other answers. Using existing operational software that has a version suitable for ATC controller units. This is a good option also. If you said C, replace the controller units as part of the regular scheduled maintenance this is a good option for migrating to ATC equipment but agencies must define their needs and goals to assess what is best. Okay. Creating a specification based on the ATC 5201 standard. We need to keep in mind that agency specifications are written based on how the agency will be purchasing the equipment. So there's no one way to do this. It depends on the agency. And one of the

methods is the agencies create standing specifications. What I mean by standing is that they're intended not for a particular job, but they are created for their agency and there for a long period of time to ensure that they get consistent units and expedite their procuring of equipment. So agencies may develop these kinds of these standing specs. And then they may develop a prequalified vendor's list. Or they may have some bid process where a single vendor is selected and used for a period of time. There's a couple of ways of doing this. We recommend that the agency separate the specifications for the controller equipment from that of the application programs. They don't necessarily have to be completely separate specs but the idea is that you want to be able to use the hardware with other application programs in the future and you don't necessarily want-- you won't necessarily in the future be running even the same traffic management software on that ATC controller unit. So the idea is to set up some method and separation where you make that clear and you're not necessarily bundling the software together with the hardware. We recommend that you develop a concept of operations based on your strategic plan. You're describing the characteristics of the TFC equipment from a user's perspective. You're covering the operational scenarios in which the equipment will be used. And this ConOps even though you do it it may or may not be part of the specification. That's up to you. Here is an example concept of operations outlined from the FHWA systems engineering guidebook version three. Really emphasize that you do this concept for the proposed system and you have a user oriented operational description of what you want to do. And also your operational needs, you need to establish those and formalize those, have kind of a system overview. And another important thing is the operational scenarios, what happens if this or that. And that's especially important if you're talking about running different application programs that may, in fact, be used under certain maybe larger events or catastrophic events. So you need those scenarios of what happens when these events occur and then probably less important ones also. Let's talk about requirements for specifications that use ATC 5201. So ATC 5201 standard generally describes a minimum level of capability for ATC, for controller units to conform to the standard. In other words, control manufacturers may go beyond those things. And also to remember when you're talking about an ATC 5201 specification it's going to end up being a composite of standards and/or other specifications. So you're going to have reference to occupants there. And the composite may consist of explicit requirements developed by the agency. They may have requirements gleaned from other reference documents. And you may actually reference requirements or sections in those other documents. You may have inherited requirements from the reference documents with an established document precedence. So here's a proposed organization of requirements. This is just a suggestion. You have your reference documents and precedents. You establish that, your general requirements, your field cabinet system requirements, interface requirements et cetera. And we're going to step down through this in the subsequent slides. Now, remember that since an ATC 5201 based specification is going to be a combination of standards being used you want to establish this precedence. For example, here might be a statement that

you have in your standard. "In the case there is a conflict between the requirements in reference documents that is not directly addressed by a stated requirement, the precedence of reference documents shall be as follows: Agency specific requirements, ATC 5201 standard version 6 requirements, your TFCS requirements" depending on what you're using NEMA TS2 or model 3XX or an ITS cabinet. Generally, your requirements in the standards will be complementary but you'll have specific agency requirements to clarify the cases where the reference documents may differ. And there may be, occasionally, a conflict that exists between reference documents. So general requirements they cover some examples of these, material requirements, electronic requirements, mechanical quality, things like that. Material like aluminum the types and thickness of aluminum electronic might be pin locations. This gets into very gory details and it depends on your agency how willing and able you go into this. And you might talk about different fasteners in your mechanical requirements. And you might talk about component sampling et cetera. This is used by some agencies but most don't go into this kind of detail. You have TFCS requirements. You need to identify what kinds of cabinet architectures that you'll be using for the controller unit. And those, depending on the architecture that'll influence possibly the user interface, the serial and parallel interfaces and the operational voltages of the controller. So let's talk about the user interfaces. Here we're going to talk about what the minimum is in according to the ATC 5201 standard. At a minimum there should be a CPU active LED indicator on the front panel of the controller. There should be an Ethernet port and a USB port and that's a removal memory device and not a common USB port for running a printer of something like that. But the idea is to be able to use your memory devices there. In addition to that, you need to have a connector in this case either a EIA-574 9-pin D or an RJ45 connector that looks like an Ethernet connector but in this case it's used in a serial fashion. But the idea for this is is for a console port. And this is typically used by developers and manufacturers who are accessing the Linux operating system and doing other industrial type things. But the minimum user interfaces has what's in the first box as well as now here what's in the second box and also in the third box. So the third box really covers the user interface for the operational users. And so that user interface will be either a keyboard-- you know, your keyboard, your liquid crystal display and Bell. But it may also have a 574 9-pin D connector so that an external front panel could be used. So this describes the minimum requirements. It also does not limit that other more advanced keyboards may be used. It doesn't restrict the use of potentially graphics and things on the LCD. But every ATC needs to have these parts. So there are some optional features that are included. Some users may want their ATCs to have a data key memory device. They may want it to have an AUX switch which comes from the model 2070 world where there's a switch on the front. And that's only available, the AUX switch is only an optional feature if the controller has a keyboard, LCD and Bell which most of the controllers except in a few cases. Then an optional feature is to have a power switch. You know, some agencies are fine with just plugging in, you know, using the plug as the power of the controller. And then also you may have power LEDs and those are used to help the user and some agencies want that

and some others don't. Okay. Continuing with our user interface requirements, again, the type of cabinet architecture or TFCS architecture you're using that may have an interface on the-- it may have restrictions on the kind of user interface that's included on the controller. And as I had said previously the standard allows expansion of the user interface requirements to include things like high resolution graphics and additional front panel keys. But these same units still need to be able to operate in a fashion that supports the minimal requirements. Now, this little caution is that I want to caution agencies to verify these types of interfaces will also operate in a mode that supports ATC 5201 and the ATC 5401 standards. So you may see something very flashy but you need to verify that with your provider that this is true, that they will work in those other modes. And now we're talking about serial and parallel I/O requirements. The TFCS architecture may have serial and parallel I/O all ready defined. In other words, there's all ready certain requirements if you say you're going to be running a NEMA TS1 cabinet or a NEMA TS2 cabinet you're all ready defining certain serial and parallel I/O requirements right there. So as we discussed earlier in this presentation ATC 5201 offers a numerous additional general purpose ports. And agencies should include the requirements for extra ports based on their concept of operations. And here this is another warning is that we caution you not to unnecessarily require communications port as can drive up costs and limit choices of vendors. In other words various vendors will package their products differently for either a manufacturing or target customers, but we just want to caution you to use your concept of operations to help you determine what is really needed. So it's kind of a similar thing here. We have a CPU and memory requirements, there's a minimum of 60 MIPS and a minimum of 64 megabytes of DRAM and 1 megabyte of SRAM and 16 megabytes of Flash memory. But we allow in this standard for that all to be increased. And, in fact, as we said previously it's difficult to even buy a 60 MIP processor at this time. So we're talking about hundreds of MIPS on your typical contemporary processor. But agencies are cautioned not to arbitrarily require the highest number of MIPS and highest memory available as they can drive up costs and limit choices of vendors. Again, the same thing we're talking about previously is that your concept of operations drive this and be careful that you just don't spec yourself into a particular vendor. The environmental and testing requirements those will vary depending on the type of cabinet system you have. Again, we said that the ATC 5201 based controllers are generally more robust than the TSC architectures. Operational voltages for the controller, though, must be indicated so that they're consistent with the architecture being used. As far as warranty is concerned, get as much warranty as possible and get software support for your Linux environment. Sometimes those are separated especially when you're looking at getting software from another vendor that you'll need to get the software support for the Linux environment on the controller itself. Other requirements will include the tool chain so that you can have other people create applications for that engine board. You'll want to negotiate the access to the source code of the Linux environment that's used on the engine board. That may be done in a special way. It may be some sort of escrow kind of thing or the companies may just offer that if you ask for it. Ask engine board suppliers

to provide operational API software for their engine boards once the API RI is available. And that might be something that you have to negotiate also if there's any fees involved with that. Or maybe if the fact that you have software support that will just be included. You may have other things like packaging requirements, delivery of requirements, manual requirements. These are things typical of any procurement done by an agency. Let's go in to testing our knowledge, again. Which of the following is a good practice when preparing a specification using ATC 5201 v06? A, establish a precedence of reference standards and specifications. B, always specify the fastest CPU available. C, always specify several extra serial point ports than you need. And D, never exceed the minimum user interface requirements of the ATC 5201 standard. Please make your selection. Let's review our answers. Which of the following is a good practice when preparing a specification using ATC 5201 version 6? If you said establish a precedence of reference standards and specifications you were correct. This is important since an ATC 5201 specification is a composite of other documents. If you said B, specify the fastest CPU, this is incorrect because we need to consider-- agencies need to consider leaving themselves out of options if they only specify the highest amount of CPU, the highest speed CPU available. It needs to be based on your concept of operations. If you said always specify several more serial ports than you need that sounds like a good idea but you need to also consider that you may be limiting your choices of controller units and may increase costs by doing that. Again, base this on your concept of operations. If you said D, this was incorrect. Never exceed the minimum user interface requirements. Use of graphics and other things those will actually be supported by the ATC API. And so the graphic user interfaces are completely consistent with the standard. However, they need to be able to also perform the basic operation that is defined in the standard. So let's summarize with a summary of our learning objective six. Specify ATC equipment for system and equipment procurement. We discussed how ATC 5201 fits in the system's lifecycle. We discussed the development of migration strategies and how to mitigate deployment issues. We talked about creating a specification based on ATC 5201 standard. So what have we learned today? We learned that the ATC 5201 standard provides for control units that can grow with technology using an engine board concept. Secondly, we learned that the ATC 5201 standard can be used to specify ATC control units that can operate in any major TFCS architecture. We learned that the ATC 5201 standard works with the ATC 5401 standard to provide for portable compatible and interchange application programs. Lastly, we discussed agencies who will reference other ITS standards or specifications in preparing an ATC 5201 specification. Here are some resources the first 405 here listed referred to the TFCS architectures that are in use today. And the ATC standards can be found with the institute of transportation engineers. And we have several modules in this line regarding the ATC standards at the ITS PCB training website. This is the second one as part of its ATC 207B and there is subsequent module A208 that will be coming on the application programming interface. Let's review some of the questions that have come in in previous iterations of this class. Someone asked what happens when you have a mission critical application running and

a data collection application running, who gets priority? Well, when a software is loaded and run on an ATC unit the application priority can actually be set by the user developer or the integrator so that you establish a priority there for the applications if there's ever a demand at the same time. But it is estimated that your typical traffic management program will use less than one percent of the CPU power to operate. So the fact that we're using these contemporary processors kind of mitigates that concern all by itself. Someone else asked, could you explain further the difference between portable and portability and source code portability? Okay. Portability might be what you have when you have X86 based application programs. In other words, my version of Microsoft Office will run on a Dell. It'll run on various other brand name PC's like HP et cetera. And you get that portability of that object code. But if you wanted to go to a different family of processors as those used in a Macintosh you'd actually have to get a different product that looks and feels the same but operates on the other machine. So that's done because the company, in this case, Microsoft has compiled a version of their software for that other family of processors. So that's the same kind of thing you will do when you get application programs. You will ask the provider of the application program to give you something compatible with your engine board. It's been said that the ATC controllers and API software work together, should I just wait for the API RI project to be completed before I buy an ATC? Well, that really depends on you. If you are in need of getting new controllers, I suggest that you do go ahead and buy an ATC but make provisions in you procurements so that to use the API reference implementation once it's available. So it all depends on what your window procurement is. Someone else asked, if there's a reference is there a lab that they can reference that certifies that a controller meets the ATC 5201 standards? And there is no national lab like there is for other industries such as UL or other things. But some state agencies and a few local agencies actually have labs where they test the products and create an approved products list for their agency. In any case, your best protection for an agency is a good specification and warranties and a formal procurement process that gives the agency confidence that they're getting what they asked for. Okay. Let's take a look at our next module will be A208 using the ATC 5401 application program in the interface standard to leverage ITS infrastructures. We have four learning objectives there. We'll identify the features of the ATC 5401 API standard. We'll describe the ATC 5401 architecture. We'll describe how the ATC 5401 standard works with other ITS standards. And then talk about specifying API software for system and equipment procurements. And this concludes our module. Thank you.

End of A207b_Final.mp4