Introduction to the SEMI Standards: Implementing GEM and PV2

Table of Contents

- NOTICE -

The information contained in this document is subject to change without notice. Every effort has been made to supply complete and accurate information. However, Cimetrix Inc. makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Cimetrix shall not be liable for errors contained herein or direct, indirect, special, incidental, or consequential damages in connection with the furnishing, performance, or use of this document.

- COPYRIGHT -

- TRADEMARKS -

Cimetrix®
6979 High Tech Dr
Midvale, UT 84047
Phone: 801.256.6500
www.cimetrix.com

1 Overview of GEM and PV2 Standards

Officially titled the "Generic Model for Communication and Control of Manufacturing Equipment," the SEMI E30 standard (a.k.a. "GEM") is a popular industry standard that SEMI (Semiconductor Equipment and Materials International) originally published in 1992. After being used extensively in 100mm and 200mm front end semiconductor factories, GEM was also adopted as the underlying standard for the "GEM300" collection of SEMI equipment software standards used for manufacturing automation in 300mm wafer fabs.

The GEM standard defines messages, state machines and scenarios to enable factory software to monitor and control manufacturing equipment. The GEM standard is formally designated and referred to as SEMI standard E30, but frequently simply referred to as the GEM or SECS/GEM standard.

GEM was developed to produce economic benefits for both device manufacturers and equipment suppliers by defining a common set of equipment behavior and communications capabilities to support manufacturing automation programs of semiconductor device manufacturers. Because the SECS/GEM standard was written with very few semiconductor-specific features, it can be applied to virtually any manufacturing equipment in any industry.

Many types of manufacturing equipment in semiconductor (front end and back end), surface mount technology, electronics assembly, photovoltaic, LED, flat panel display, and other industries worldwide provide a GEM interface so that the factory host software can communicate with the equipment for monitoring and/or controlling purposes.

In September 2007, the European SEMI Photovoltaic Equipment Interface Specification (PV-EIS) task force started work to standardize photovoltaic (PV) equipment communication interfaces by leveraging the most appropriate existing technology. In 2008, the task force selected the SEMI E30 GEM standard as the underlying technology and, then proposed the new PV Equipment Communication Interface (PVECI) standard to specify how the GEM standard should be implemented on PV equipment. In June 2009, SEMI members officially approved the PV2 document as SEMI standard PV2.

1.1 Industry Usage of the SECS/GEM Standard

All GEM equipment can communicate with a GEM-capable host using either TCP/IP (using the HSMS standard, SEMI E37) or RS-232-based protocol (using the SECS-I standard, SEMI E4, or both. Each piece of equipment can be monitored and controlled using a common set of SECS-II messages specified by SECS GEM.

When equipment has a GEM interface, it takes just minutes (or even seconds) for factory GEM host software to establish communications and begin monitoring the equipment's activity. By using an industry-standard GEM interface, equipment manufacturers can focus on their core competency, instead of spending scarce engineering resources on implementing connectivity. It also means that factories can spend more time and money improving production operations and process capabilities, rather than setting up communication to the equipment.

There are additional SEMI standards and factory specifications that are either industry-specific or equipment-type specific. A number of these are covered in the following sections.

1.1.1. Semiconductor Front End

The semiconductor front end segment of the industry defined a series of standards known as the GEM300 standards that includes SEMI standards E40, E87, E90, E94, E116, E148, and E157, and also reference the E39 standard.

Each standard provides additional features to the GEM interface, yet build upon the features in GEM E30 standard. 300mm factories worldwide use the underlying GEM standard's data collection features in order to monitor specific equipment activity such as wafer movement and process job execution. The SECS/GEM standard and the additional GEM300 standards are required on almost all 300mm wafer manufacturing tools in order to implement manufacturing automation. This industry segment has been the strongest supporter of the GEM and related SEMI standards.

You can read more about the GEM300 standards in the Cimetrix white paper ***Introduction to the SEMI Standards: GEM300.***

1.1.1 Semiconductor Back End

Numerous types of equipment in the semiconductor back end segment of the industry implement the GEM standard. Additional standards have been implemented for some of the specific equipment types, such as:

- SEMI E88: Specification for AMHS Storage Specific Equipment Model (Stocker SEM)
- SEMI E122: Standard for Tester Specific Equipment Model (TSEM)
- SEMI E123: Standard for Handler Equipment Specific Equipment Model (HSEM)
- SEMI E130: Specification for Prober Specific Equipment Model for 300mm Environment (PSEM300)
- … and a number of others

1.1.2 Surface Mount Technology

Many equipment in the Surface Mount Technology industry support the SECS/GEM standard, including chip placement, solder paste, oven and inspection equipment. The GEM standard has been used on these equipment types for over 15 years.

1.1.3 Flat Panel Display

The flat panel display industry has largely been a long-time user of SEMI standards for connecting manufacturing equipment to their factory information and control systems, but the interfaces are typically user specific and incorporate custom SECS messages. Other FPD factories have also deployed custom integration approaches.

Although many of the equipment suppliers to the FPD industry also offer semiconductor equipment, we don't expect the FPD industry to adopt the newer generations of advanced data collection and process control standards in the short term. This is due primarily to the fact that the FPD manufacturing process was not as complex as semiconductor production (more inline), and also because FPD manufacturing was highly automated from the outset.

However, as panel sizes and feature counts continue to increase (consider the large LED-based high-definition televisions), the FPD industry will undoubtedly make use of more and more manufacturing data to maintain product quality and manufacturing efficiency—and the SEMI standards will be there to support them.

1.1.4 Surface Mount Technology

Many types of equipment in the surface mount technology industry support the GEM standard, including chip placement, solder paste, oven and inspection equipment. The GEM standard has been used on these equipment types for over 15 years.

1.1.5 Photovoltaic

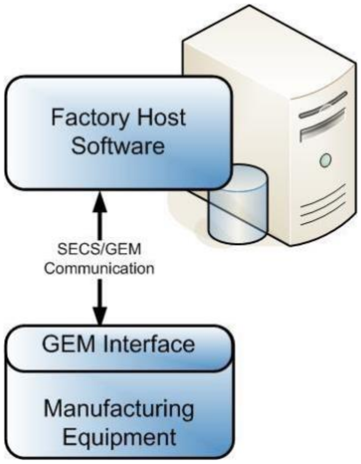
In 2008, the Photovoltaic industry officially decided to adopt the SECS/GEM standard and submitted a proposal for a new SEMI standard, ballot 4557. Even prior to adopting the GEM standard, several photovoltaic equipment suppliers were already capable of supporting the GEM standard. The standard is called PV2, and defines a framework that utilizes the SEMI E37 (HSMS), SEMI E5 (SECS-II), SEMI 30 (GEM), SEMI E148 and SEMI E10 (OEE) standards.

1.1.6 High-Brightness LEDs

The high-brightness LED industry is currently working with SEMI to define needed standards through the HBLED Task Force. The adoption of GEM has been accepted and further investigation is taking place concerning the GEM300 and EDA (Equipment Data Acquisition) standards.

2 Technology Overview

A GEM (E30) interface is implemented by the equipment manufacturer to enable the equipment and factory software (a.k.a. "host") to communicate using SECS-II (E5) messages over a standard Ethernet network. The purpose and format of each message is well defined within the SEMI E5 and E30 standards. A SECS-II message is identified by the combination of a stream and function number, S#F#. For example, S1F13 is used to establish communications between the factory host software and the equipment software.



The GEM standard specifies approximately 80 SECS-II message definitions that can be implemented, but equipment manufacturers typically implement only a subset of these messages. In particular, the PV2 standard identifies the specific subset of SECS-II messages that must be supported by photovoltaic equipment.

Some SECS-II messages can only be initiated by the factory host software, such as a request by the factory for equipment data or to send remote control commands. Some messages are initiated only by the equipment, such as alarm and event notifications like "process completed." A few messages can be initiated by either party, such as messages for exchanging production recipes (referred to as process recipe management). Although uncommon, some factories and equipment might require the use of SECS-II messages that are outside the scope of GEM or PV2.

GEM standard compliance is divided into two sections - Fundamental Requirements and Additional Capabilities. GEM defines compliance only for the equipment interface, not for the factory host software. Implementations of the GEM standard can be scaled to match the complexity of the equipment and the needs of the factory host software. The GEM Fundamental Requirements should be implemented by all equipment. These include establishing communication with the factory host software, implementing a processing state machine, event notification, protocol error messages, and a GEM implementation document.

While complex equipment might provide thousands of different GEM events for notification, very simple equipment might provide only a few. The GEM Additional Capabilities are optional features that should only be implemented when the factory plans to use these features. These features include alarm reporting, dynamic data collection features, remote control, process recipe management, terminal messages, spooling, and limits monitoring.

GEM COMPLIANCE STATEMENT		
FUNDAMENTAL GEM REQUIREMENTS	IMPLEMENTED	GEM COMPLIANT
State Models	□Yes □ No	□ Yes (see#1) □ No

Equipment Processing States	□Yes □No	
Host-Initiated S1,F13/F14 Scenario	□Yes □No	
Event Notification	□Yes □No	
On-Line Identification	□Yes □ No	
Error Messages	□Yes □ No	
Documentation	□Yes □ No	
Control (Operator Initiated)	□Yes □ No	
ADDITIONAL CAPABILITIES	IMPLEMENTED	GEM COMPLIANT (see #2)
Establish Communications	□Yes □ No	□Yes □ No
Dynamic Event Report Configuration	□Yes □ No	□Yes □ No
Variable Data Collection	□Yes □ No	□Yes □ No
Trace Data Collection	□Yes □ No	□Yes □ No
Status Data Collection	□Yes □ No	□Yes □ No
Alarm Management	□Yes □ No	□Yes □ No
Remote Control	□Yes □ No	□Yes □ No
Equipment Constants	□Yes □ No	□Yes □ No
Process Recipe Management	□Yes □ No	Process Programs: □Yes □ No
		E42Recipes: □Yes □No E139Recipes: □Yes □No
Material Movement	□Yes □No	□Yes □ No
Equipment Terminal Services	□Yes □No	□Yes □ No
Clock	□Yes □No	□Yes □ No
Limits Monitoring	□Yes □No	□Yes □ No
Spooling	□Yes □No	□Yes □ No

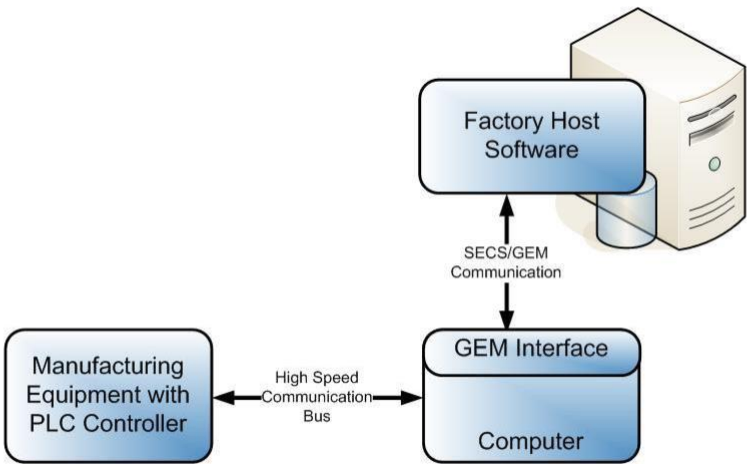
Much like how the GEM standard is a subset of the SECS-II standard with additional required features, the PV2 standard is a subset of the GEM standard with additional required features. The following is a list of specific PV2 features:

- The required format to use for data items in the SECS-II messages
- A specific list of variables, equipment constants, and collection events
- A subset of SECS-II messages
- An implementation of SEMI E10 to report equipment states related to reliability, availability, and maintainability (RAM)
- An implementation of the Network Time Protocol (NTP)
- A statement of PV2 compliance

These PV2 requirements should make PV2-compliant equipment even easier than GEM to integrate with the factory host software.

3 System Architecture

A GEM or PV2 interface can be added to any manufacturing equipment. Most GEM interface software runs on an industrial computer; therefore, if the equipment already has a computer, the GEM interface software can be added as a new module on that system. In the case where the manufacturing equipment is controlled entirely by one or more PLCs with no industrial computer for the GEM interface software, a computer must be added. This is a common requirement for photovoltaic equipment. This architecture requires a high-speed duplex communication bus between the PLC and the computer in order to keep the GEM interface well synchronized with the equipment. Many types of manufacturing equipment have a distributed architecture with multiple computers that must be synchronized with the GEM computer.



4 Impact on Existing Equipment

A PV2 or GEM software interface cannot be easily added to existing equipment without impacting the design of the control system. GEM is not an "add on," nor is it trivial. Equipment suppliers need to be aware that proper GEM implementations are "invasive" and will likely require changes to existing software; they may also require changes to the computing hardware.

GEM not only provides data collection functions, but also remote control and process configuration capabilities. Here are some examples of GEM requirements that are likely to impact existing equipment designs:

- For GEM data collection, the GEM interface needs access to data as fast as 10 Hz from most, if not all, modules in the equipment.
- The GEM interface needs access into the primary control system to support requirements for remote control and process recipe management.
- A GEM-compliant equipment must implement a processing state machine which requires frequent information and interaction with the equipment control system.
- In order to be GEM compliant, certain GEM features must be added to the operator interface. Typically, these features are added to the graphical user interface, but can also be implemented using switches and LEDs.

In order to minimize the impact of these changes, the GEM interface should be designed into the equipment architecture as early as possible in the equipment development cycle. Waiting until the late stages of development makes it more difficult to integrate, and can be detrimental to the equipment and GEM interface quality.

5 Ensuring GEM or PV2 Interface Quality

As semiconductor and solar (photovoltaic) factories increase automation and further refine manufacturing processes, they will require more information from the equipment, this increasing their reliance on the GEM interface. The factory users need and expect high-quality, reliable GEM software on the equipment. Listed below are a few historical trouble spots for equipment suppliers implementing a PV2 or GEM interface. Equipment suppliers should run tests to confirm the equipment they develop demonstrates the appropriate functionality.

5.1 Collection Event Data Quality

GEM equipment can send messages to notify the host of certain key "events." GEM defines these as "collection events" because these events present a good opportunity for the host to collect context data associated with the event, such as the lot ID, recipe name, and/or process parameter values. GEM provides this data collection capability by allowing variables on the equipment to be linked to or associated with one or more events.

When sending event messages to the host, it is critical for the GEM software to accurately report the values that those variables had at the time the event occurred. A common problem in some GEM interfaces is that the reported variable value is wrong when multiple occurrences of the same event are nearly simultaneous, or when the same variable is associated with different events. Collection event data quality can only be ensured when designed into the GEM software, correctly implemented, and verified though explicit testing.

5.2 Equipment GEM Documentation

It is disheartening to look at a map and at the road and see that there is a clear, impeding discrepancy, such as a "road closed" sign in the direction of one's planned travel. Although a GEM interface is somewhat self-describing, most of the useful information about a GEM interface is available only by means of the GEM interface documentation. The GEM interface document is created and distributed by the equipment supplier along with the equipment.

Equipment suppliers need to ensure that the GEM documentation is well maintained and consistent with the corresponding GEM interface software.

Each time a new GEM interface revision is installed, equipment suppliers should provide an updated GEM interface specification for the users. GEM requires specific data items and messages for the host to query information about the equipment's GEM software revision. These data items should be updated to match the new revision information for the GEM interface specification.

5.3 The "Spooling" Challenge

The GEM standard includes a feature called "Spooling," which is an optional feature in the PV2 standard yet required by some PV factories. The purpose of the spooling feature is to enable the equipment to save important messages, like events and alarms, when the host communication interface has failed, and then send those saved messages when the host communication is working again. Once the equipment begins spooling messages, it will continue to do so until the host connects again and requests the spooled messages.

Most recent implementations of GEM allow the spooling feature to be disabled, as allowed by the GEM standard. This is important because if a factory does not use the feature, it can cause some problems with the factory automation scenarios if the equipment suddenly begins spooling messages and the host cannot tell it to stop spooling and start transmitting again.

Spooling is one of the more complex GEM functions, and it is common for new implementations to have spooling-related defects. Implementation of GEM Spooling requires a thorough analysis of the requirements and use cases, careful and robust designs, and a lot of testing. This is certainly one aspect of GEM where using proven, commercial solutions can save a lot of time and effort, providing a much higher quality implementation.

5.4 Complying with Factory Scenarios

Like any standard, the PV2 and GEM standards are subject to interpretation. Ultimately, the interface must interact with the factory host software to satisfy its SECS-II message scenarios (specific, required sequences of SECS-II messages). While compliance to the PV2 and GEM standard is important, this does not ensure a successful integration with the factory host software at every customer site. Implementation details, use cases, and performance requirements will vary from equipment to equipment and factory to factory based on manufacturing needs. For example, if an equipment supplier implements the GEM trace data collection feature (time-based data sampling) expecting a 2-second sample interval, but the factory host software expects a 200 ms sample interval, then the equipment's GEM software is not going to meet the user's requirements. As another example, while one factory might wish to download process recipes before processing material arrives, another factory might wish to download process recipes after material arrives. Both are valid, but the use cases are quite different, so the equipment must support both scenarios if its customers require it.

In the semiconductor industry, it is common practice for the factories to provide equipment suppliers with SECS/GEM scenario documents and additional factory conformance requirements. These artifacts serve as the detailed GEM interface requirements, and likewise determine testing expectations. Additionally, it is common practice for equipment suppliers to provide the GEM Interface Manual to the factory for review and approval before the equipment is delivered. This gives the factory an opportunity to approve or request changes in the GEM interface before the system is delivered when the supplier can more easily accommodate the change requests. The PV industry is still developing solutions for these software requirements and integration issues.

5.5 Maintaining Continuous Operation

Both the equipment supplier and the factory expect the GEM interface to operate 24 hours a day, 7 days a week, and 52 weeks a year without any failures attributable to the GEM software. Software quality is paramount. Developing such quality and reliability into software requires an investment and software expertise. Many equipment suppliers are engineering companies that are forced into developing software in order to sell and put their engineering products into production. It can be difficult for some managers to recognize and justify the need for this software development investment.

5.6 Integrating Process Recipe Management and Remote Commands

As mentioned earlier, the PV2 or GEM interface impacts the entire equipment design. In particular, the process recipe management and remote command features are particularly challenging to implement. Process recipes are the equipment processing instructions. As in the GEM standard, they are also optional in the PV2 standard. When implemented, they must be integrated in a safe, reliable manner.

On complex equipment, the recipe information must be distributed to multiple, disparate destinations. It can be difficult to develop a single recipe format that configures all of the subsystems which can also be enhanced in the future as new equipment features are added. To avoid some of these problems, equipment manufacturers should format the recipe files using an XML schema. XML is a natural format to organize even complex recipe information and is extensible. Using XML also facilitates the creation of off-line process recipe editors.

It is also common for equipment to fail to perform recipe validation and verification. The validation logic is often embedded in GUI controls which manage data entry to avoid potential recipe problems. However, the recipe downloaded from the host cannot be easily validated through the GUI controls. For these reasons, one should separate the recipe validation and verification software from the GUI so that the same logic is applied to the GUI and GEM interface.

Perhaps one of the most prevalent issues in the semiconductor industry is the management of process recipes. It can be difficult to guarantee that the equipment is using correct recipe version, thereby putting the manufacturing quality at risk. The PV industry can avoid some of these version control issues by changing the process recipe version every time it is edited (whether locally at the equipment, offline, or through the factory host software), and by reporting the current recipe version accurately through the GEM interface. If using XML format, the version can be embedded in the process recipe itself.

6 Optimizing Equipment Performance

When implementing PV2 or GEM, it is important to have a clear understanding of the performance requirements for the interface. The exact requirements vary from factory to factory and equipment type to equipment type. Equipment suppliers should request specific performance requirements from their customer's factory. The technology used for the GEM standard is designed or intended to be used as a real-time communication interface. Still, good performance of the communication software—or perhaps more importantly, predictable and reliable performance—is critical to successful manufacturing automation.

The most important consideration for system performance is that the GEM software must not have a negative impact on the performance, throughput, or reliability of the equipment's primary function of processing material. Poorly designed GEM software can cause degradation in equipment performance or increase cycle times. The degradation might be constant or increase over time or be correlated to GEM message activity. The equipment must be capable of running as efficiently with the GEM communication as it does without GEM.

6.1 Message Processing Performance

SECS-II messages can only be sent one at a time. It is acceptable and even necessary to queue up SECS-II messages before they are sent. With modern software development practices using multi-threading, multiple processes, and queues, it is possible to maintain equipment performance while reporting accurate information through the GEM interface. Many of the SECS-II message transactions are initiated by factory host software request. The equipment should respond promptly to most host requests, ideally in less than a second.

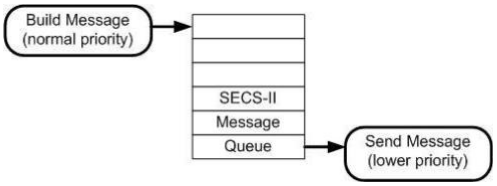


Figure 3: SECS-II Message Priority

6.2 Communication Load

The factory host software determines network bandwidth requirements, regardless of the complexity of the GEM interface. The factory host software determines which collection events are enabled (if any), which alarms are enabled (if any), which data collection reports are defined and enabled (if any), and all other aspects. Without factory host software assertion, a GEM interface is idle. In contrast, a very simple GEM interface from a single equipment could overload a network if the factory host software sets up extensive data collection. Therefore, it is very important for factories to create and publish GEM interface use case scenarios so that the equipment suppliers can verify expectations.

6.3 Synchronization with the Data Sources

While everyone agrees that the GEM interface must report current data, the precise requirements and tolerances associated with data latency vary greatly. In the context of a GEM interface, latency is the time between a variable value changing at the data source and the time that the factory host software receives the value from the GEM interface. The latency requirement for some data might be 100 milliseconds or less, while for other data the latency requirement might be 5 seconds or more. The function of the equipment, the type of data being collected, and the use of the data by the factory are all factors in determining these requirements.

In a GEM interface, poor data quality resulting from data latency is a common problem, especially for older GEM implementations. In order to create a well-performing GEM interface, the equipment supplier must carefully evaluate the equipment's hardware and software architecture to ensure that data can be transmitted to the GEM interface software with sufficient performance. The equipment software and hardware might need to be redesigned or upgraded to meet these requirements. For example, if the GEM interface software and other equipment systems are running on separate computers or PLCs, the system will require a high-speed communication bus.

The GEM standard provides a few different features for sampling data. For time-based data sampling, it is common to see requirements for data sampling rates for 1 to 100 variables at a frequency of up to 10 Hz. Data collection frequencies higher than 10 Hz can be accomplished by some equipment; however, rates faster than 40 Hz are not practical for most GEM interfaces.

The SEMI E151 Data Quality Guideline defines terminology for discussing data quality topics; please refer to this standard for more information.

6.4 Equipment Cycle Time Degradation and CPU Utilization

In a poor GEM interface implementation, turning on the GEM interface data collection causes the equipment performance to degrade. The obvious side effect is an increase in cycle time. For example, initially the performance might be satisfactory, but then performance degrades once all collection events are enabled or when data collection reports are defined. Once the performance problem is detected, it is never acceptable to the factory. Such problems can be avoided by using modern software practices—including distributed processing, multiple processes, multi-threading, and operating system priority management—to ensure the most important processing tasks are not interrupted.

Timestamp and process data values must be accurate in the SECS-II messages, but the transmission of the data can lag a little, as the GEM software works to deliver queued messages. Many equipment types have CPU utilization "surges" that can restrict the GEM software's message processing performance. However, when the CPU utilization returns to lower, sustained levels, the GEM interface can catch up with the queued work. If there are situations in which the GEM software on the equipment gets behind and cannot catch up, then the software and hardware designs might need to be modified to provide a higher level of performance.

7 Implementation Tips

7.1 Disciplined Software Development Practices

Implementing the PV2 or GEM standard is a software development project; therefore, the development should follow a proven software development methodology. All software development methodologies require the team to perform a requirements analysis. In a PV2 interface, the requirements come from the GEM standard, the PV2 standard, the SECS-II standard, the equipment itself, and most important, the customer. The equipment developers must work hard to determine how the factory intends to use the PV2 or GEM interface. Factory engineers should create SECS-II messages scenarios to establish clear equipment interface expectations.

Again as in all software development methodologies, the PV2 and GEM development needs quality project management oversight. Software developers tend to underestimate the effort required to deliver a reliable, quality interface. An experienced project manager can help the team to break down the project into clear, attainable milestones and ensure quality procedures are followed at each development stage.

7.2 SECS-II Message Logging

The best way to diagnose GEM communication problems is by analyzing the communication between the GEM interface and the factory host software. This can be captured by the equipment GEM interface software, factory host software, or by a third-party product that can tap into the communication called a "Network Sniffer" or "TCP/IP Sniffer." Most, but not all, third-party Network Sniffer software packages do not support SECS-II messaging, and are therefore only of limited use. Because the SECS-II messages are transferred on the network in a standard binary format, the SECS-II messages must be converted into human-readable text before interpreting the problem. A good GEM interface will provide an advanced feature in the GEM section of the equipment's user interface to enable and disable the capture of SECS-II message logging. Also, the captured logging files should be readily available to the equipment operator for diagnosis by the GEM experts.

7.3 Test Customer Scenarios before Shipping

Each time an equipment supplier receives a SECS/GEM scenario from the customer factory, this scenario should be incorporated into the equipment's software release test plan. Every hardware release and software upgrade release should include executing the known factory host software scenarios. Do not install any software at the factory unless it has been certified to pass the various scenarios. Following this simple rule can avoid many problems.

7.4 GEM Equipment Simulator

Few equipment suppliers keep spare equipment on hand for testing and experimentation, since each unit of equipment is a significant investment and typically sold as soon as possible. This presents a challenge for the equipment engineers and software developers who need access the equipment hardware in order to continue development. Access to fully assembled and functioning equipment hardware is limited to short windows of opportunity.

Factories have similar issues. In order to develop the factory host software and integrate it with each piece of equipment, the factory software developers need access to the equipment. This usually cannot be done while the equipment is running in production. In order to facilitate equipment and factory host software development, equipment suppliers should invest in the development of a software equipment simulator. Even in a simple form, an equipment simulator can be very effective at increasing engineering productivity. With more investment, the equipment simulator can imitate the real equipment with great fidelity. Much of the GEM interface can be developed and tested using an equipment simulator. In the semiconductor industry, some factories require GEM simulators. In time, GEM simulators may become a requirement in the photovoltaic industry as well.

7.5 Early Integration

As mentioned earlier, implementing PV2 or GEM on an equipment can have a big impact on the equipment's entire hardware and software system. Therefore, plan to integrate as early as possible to understand and minimize this impact. Features in the GEM interface can help to clarify the equipment processing design. For example, GEM requires the equipment to publish a processing state machine. The details of the processing state machine are designed by the equipment supplier and organizing the equipment operation into a proper state machine helps to clarify the equipment processing design and implementation. It helps to determine what the operator can and cannot do in the processing, diagnostics, and service operation modes. Integrating GEM early in the equipment design lowers the cost of GEM development, improves the GEM interface performance, and increases the GEM interface quality.

7.6 Arrays vs. Lists

GEM uses "variables" to report data to the factory host software. The variable can contain a single unit value, an array of values, or a list which can contain a structure of values. When a variable needs to contain multiple values of a single data type (e.g., floating point values), implementing the variable as a single array data item is more

efficient than implementing it as a list data item. This is because an array is a single SECS-II data item, whereas a list contains multiple SECS-II data items, and therefore, more data. For example, a list of 1000 four-byte integer values uses 6003 bytes in the message, whereas an array of 1000 integer values uses only 4003 bytes.

Creating a SECS-II list in the GEM software requires more CPU and memory than creating a SECS-II array. In addition, sending a list across the network uses more network bandwidth than an array. Parsing a list of data in the factory host software likewise requires more CPU and memory than an array. This is not to say that lists are not useful, but arrays should be used when possible.

The list structure can be a simple list of values or a complex set of lists within lists. A complex list is more difficult to create for the equipment engineers, and more difficult to parse for the factory engineers. Although complex lists can be appropriate in some circumstances, often it is best to split up the data into multiple variables with simpler structures. This not only simplifies the engineering effort to handle the data, but also facilitates adding more data in the future, because once a GEM interface has been published, the variable list structures should not be modified.

7.6.1 Testing

There are multiple ways to test a GEM interface and each technique or method requires special software that can emulate the host scenarios. There are commercial GEM testing software packages that can emulate a host as well as provide generic software test tools that can be useful. It is also common for equipment suppliers to develop and maintain custom software for testing and building test automation. This "library" of test software will grow over time to incorporate additional requirements from new customers and lessons learned from the existing installations. When planning the GEM implementation project, remember to include the development of this test environment to emulate the factory host.

Testing must be initially performed by the equipment supplier before the GEM interface is delivered to the factory. Most factories will perform additional acceptance testing to confirm compliance to the factory requirements and the GEM standard itself. Additionally, the equipment supplier and factory can contract a third party to perform an "equipment characterization," to perform GEM compliance testing, and to evaluate the GEM implementation.

8 Conclusion

Understanding the SEMI SECS/GEM and PV2 standards, and the impact on their product roadmaps, might seem a little daunting for many equipment suppliers. This paper highlights key elements and issues associated with GEM software projects to help guide users toward a successful, high-quality, and hopefully profitable implementation.