Implementing Simple Automation on Legacy Equipment without OEM Support

Dana Schwartz and Corey Anderson

TriQuint, 2300 NE Brookwood Parkway, Hillsboro, OR 97124
dana.schwartz@tqs.com (503)-615-9194

3

Keywords: Legacy Equipment, Automation, ARCNET, QNX

Abstract

Factory automation and legacy equipment are both common in the gallium arsenide/nitride industry. Here we describe an approach to integrate an older legacy etch platform without built-in external communication systems to our factory automation system.

Introduction

Legacy semiconductor manufacturing equipment is often used in the GaAs and GaN semiconductor industry.[1] The equipment has the advantage of low capital cost. One disadvantage is that some legacy equipment does not have the built in capability to interact with factory automation systems. Here we describe one approach to implement simple automation on legacy equipment that did not originally have any built-in ability to interact with factory automation.

A portion of TriQuint's etch tools are legacy tools from Plasma Materials Technology (PMT) and lack OEM support. These PMT tools do not have any external communication or external networking capability. This lack of communication left TriQuint exposed to a number of expensive failure modes. Failure modes included corrupt recipes, incorrect equipment configuration, incorrect lot set up and introduction. Since no data was recorded by the PMT and saved for later analysis it was often impossible to find root cause for low yielding lots. A lack of data made corrective action difficult, slow and frustrating. The PMT tools perform a portion of the process condition sensitive etches in the flow. Without data logging capability there were multiple re-occurrences of problems because the true root cause was difficult to find for intermittent problems.

Here we describe our approach to integrating some limited automation with the PMT tools. Our goal was to implement a robust yet simple automation scheme designed to eliminate TriQuint's exposure to damage caused by corrupted recipes and a number of intermittent equipment related failures. The automation scheme generates and transfers text files to an external and Ethernet capable computer. Enabling communication improves the quality and predictability of TriQuint's manufacturing process.

Description of Equipment

The PMT tool is a dual chamber high density plasma etch platform. The machine uses software and hardware designed by PMT to control the hardware for wafer etch applications. The PMT software runs on top of the QNX2 operating system. The QNX2 operating system uses Attached Resource Computer NETwork (ARCNET) to communicate between several node computers that make up the PMT tool's hardware control system.

The gateway computer shown schematically in figure 1 was built using Dell Pentium model 210L, and was used to interface between the PMT tool's ARCNET and TriQuint's internal Ethernet network. These older Dell computers have a chipset compatible with the QNX4 networking software. ARCNET cards compatible with the Dell 210L were purchased from Contemporary Controls (model number PCI20U-CXS). Each Dell computer could only support a single ARCNET card to interface with the PMT, and as a result every PMT needed its own gateway computer. The QNX4 operating system was used for the gateway computer because there was a known possibility of transferring files between QNX2 and QNX4. The necessary communication software was produced by a contract QNX2/4 programmer. The QNX2 operating system does not treat the gateway computer as a proper member of the internal ARCNET but instead treats it as a modem.

A Linux server using Perl scripts drives communication between the gateway computer and the PMT node 1 computer. All of the logic to gather and respond to data produced by the PMT was generated from the Linux server. Limiting the logic inside the gateway computer and the QNX2 computer made maintenance and updating of the system simpler.

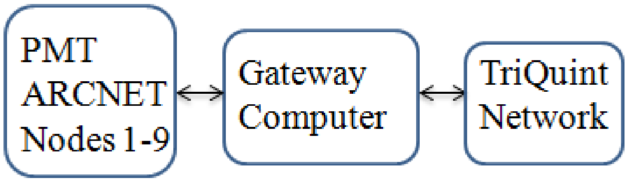

Figure 1: Simple schematic diagram of the connection between TriQuint's factory automation and the legacy plasma etch tool. The gateway computer was custom built to allow for limited automation functions.

Shown in figure 1 above is a simple schematic of the connections between TriQuint's factory automation, the gateway computer and finally the PMT's internal network. PMT node computer one controls the user interface and communication with the gateway computer. The other nodes control the individual sub-components of the equipment. The purpose of the gateway computers is to shuttle commands and files between factory automation and the QNX2 computer.

The PMT software has a built-in command line driven program called Send Message (SM). This program allows commands that return values of any tool variable. The SM program was then used as the basis for another computer program written by us to create a text file of selected PMT tool parameters upon demand called PMTSTATS. The selected parameters were process specific and different between tools. PMTSTATS is called by the gateway computer before every lot introduction to ensure that operators have set up the tool correctly before the lot starts processing.

SOFTWARE EVALUATION:

Attempting to install new software on production equipment is a task that must be performed carefully with vendor supported machines. In this case the risks are even higher since vendor support is no longer available and we do not have the original source code. The risks are compounded again because the computers running the QNX2 operating system are constrained by limited hard disk space and available Random Access Memory (RAM). Given these limitations, we designed our system to use the smallest possible QNX2 system resources. As a result the only capabilities exposed by our new software were to run command line statements at the QNX2 terminal and to stream text files from the QNX2 computer to the gateway computer.

Thorough testing was obviously necessary for each component in our new communication system. There is no built-in ability of QNX2 to stream text files. Custom software was written to stream the text files from QNX2 to the gateway computers. We started by building a simple network with a standalone QNX2 computer, a gateway computer and a script to drive the gateway to send and receive files from the QNX2 computer. Our success criterion for this test was to achieve 48 hours of continuous communication between the gateway and the QNX2 computer. During this test a ten megabyte text file was sent from the gateway to the QNX2 computer and then copied back from the QNX2 computer to the gateway. The same set of bytes was then used for the next round of transfer between the gateway and QNX2. Using this strategy the, file was manipulated by the gateway and the QNX2 machine for more than eight thousand iterations. At the end of this test the resulting text file was compared to the original file and found to be an exact match.

Testing of the new software on the PMT tools was taken slowly and step by step. Initially we used just one PMT tool to test the new software. The tool was selected because it was assigned a simple single step etch process that had an endpoint controller. The etch process itself was robust and the risk of misprocessing wafers was low. The pilot was expanded one tool at a time until all ten machines were running the new communication software.

Initially we simply tested the ability of the PMT to handle and etch wafers with our new communication software loaded into memory. Once we established that the PMT performed as expected with the new software we repeated the file transfer test described above while the tool was processing test wafers to verify normal function under stress from additional ARCNET traffic.

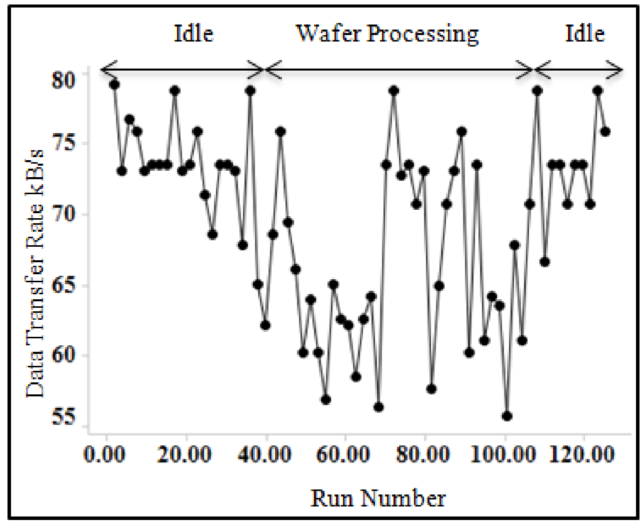

Figure 2: Data transfer rate of a 10 megabyte text file between the node 1 computer on the PMT tool and the gateway computer.

Figure 2 shows the results of a data transfer rate test performed while processing test wafers. The test was performed with a ten megabyte text file transferred back and forth from the gateway computer to the PMT computer in a continuous loop. Our 10 megabyte text file is 10 times larger than the largest data log file we anticipated downloading from the PMT tool. The purpose of the test was to stress the PMT node 1 computer beyond the maximum load expected during actual wafer processing. Before wafer processing begins the data transfer rate is stable at approximately 73 kB/second. During wafer processing the data transfer rate drops to a low of 55 kB/second. The data transfer speed drop is expected in this case. The QNX2 operating system assigns a priority number for each running process in the multitasking environment. The programs related to wafer processing have a higher priority number and have a lower que time when they request microprocessor resources. The new communication software is a lower priority than the PMT software and as a result the file transfer speed drops during wafer processing.

REDUCING RISK TO PRODUCTION WAFERS

The gateway computer has two purposes. The first purpose is to allow us to gather data about the internal operation of the PMT tool itself. This includes internally available memory resources, and the numerical values of tool parameters. The second purpose is to serve as a controlled storage location for the PMT recipe files.

PMT tools often have software related problems that require restarting the PMT node computers. During the computer restart some equipment related parameters end up re-set to default values. Maintenance technicians then have to set the correct equipment parameters up after every computer reboot. This represents a significant risk that the tool will not be configured correctly after a system restart. To protect against the risk of incorrect configuration, selected equipment constants are also written to a text file by the PMTSTATS program before every lot introduction, and checked against a preset value stored in an engineering accessible database. Operators are alerted with error messages that specify the correct action to take when equipment constants are found to be configured incorrectly.

Pre-production recipe verification was implemented as protection against recipe file corruption. Before every lot, introduction the selected recipe files on the PMT are verified against the files on the gateway computer. If any differences are found for the recipe the operator is alerted and introduction of the lot is prevented by presenting the operator with an alert message on a lot introduction web page. In addition recipe files stored on the gateway computers can be restored to the PMT tool as well if any are found to be corrupt or if the hard drive of the PMT needs to be re-formatted.

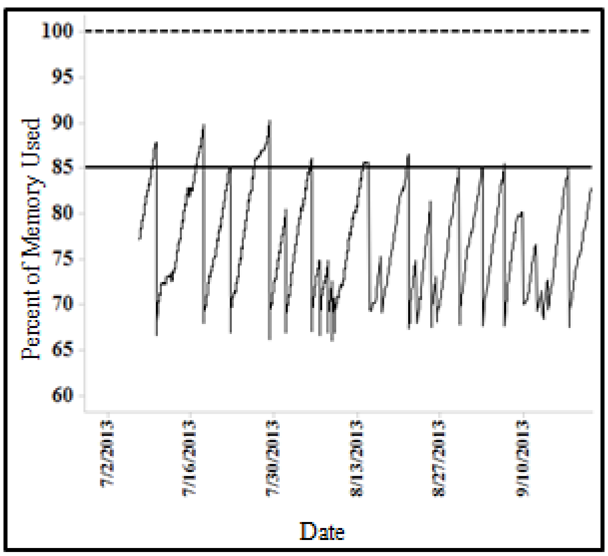

Figure 3: Percentage of available memory used as function date. The memory usage increases with each wafer processed until the machine uses all available memory.

The PMT software is especially prone to RAM memory leaks. Shown in figure 3 is the percentage of memory used as a function of time. During wafer processing computers that control the PMT reserve extra RAM, and do not release the RAM when processing completes. As a result the PMT has less available memory after processing a wafer. When memory usage reaches 100% the PMT ceases to function correctly. There are multiple node computers on each PMT that all have the same problem. Before every lot introduction the memory status of each node computer is written to a text file and compared against the pre-set limit of 85%. The limit was changed from 90% to 85% on July 28th 2013 to provide additional margin against software related wafer scrap events. The memory from the node computer shown in figure 3 does not always reach the limit of 85% because one of the other node computers occasionally reaches the limit first. As soon as any individual node computer reaches the pre-set memory limit then all of the PMT computers must be re-started. The node computer memory begins at approximately 68% post reboot and steadily climbs until the next reboot, and this creates the saw tooth pattern observed in figure 3.

Once the protection of recipe files and the usage of memory statistics were well established, the retrieval of wafer processing log files was the final task. The PMT has a limited ability to produce logs of some important chamber processing runtime parameters. The log files are stored on one of the node computers. The PMT will continue to create log files as long as hard disk space exists. Once the hard disk space is filled the PMT tool simply ceases to function in the middle of wafer processing. We found during testing that if the hard disk fills up, the QNX2 operating system may begin to overwrite existing data on the disk. The log files were normally disabled because of the risks to wafers, and the PMT.

With the above risks, enabling log files required us to implement an automated protocol of checking the machines for log files every two minutes. Once log files are located they were copied from the PMT to the gateway computer and then deleted from the PMT. The log files are then loaded into an accessible database for offline analysis. Additional protection against filling up the hard drive was implemented by systematically checking the available hard drive space before new lot introductions, and creating appropriate warning messages.

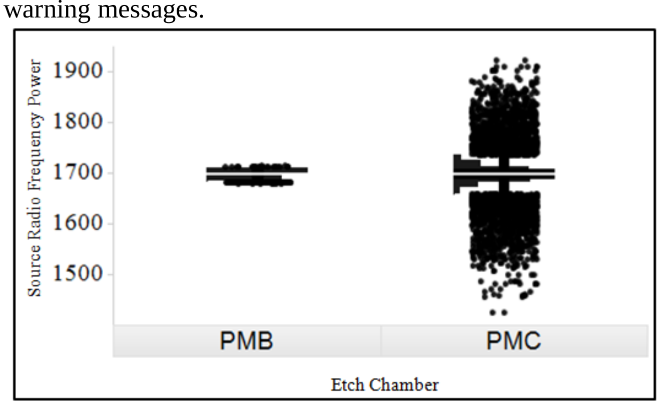

Figure 4: Box plot of the PMT Source radio frequency (RF) power as reported by the source power generator. Chamber PMC has a much higher standard deviation for source power than chamber PMB.

Figure 4 above shows the box plot of the PMT source RF power as collected and stored by the PMT data logging software. The data are presented to demonstrate the re-enabling of the PMT data logger and success in retrieving the data. In the example shown above, the chamber labeled "PMC" has a much higher distribution of delivered power. This particular problem only occurred on a single recipe at one step. The log files enabled the identification of the problem and allowed the engineers to confirm that the solution functioned as expected.

Conclusions

Legacy production equipment presents unique challenges for modern factory automation. We have shown here one method for integrating older equipment with factory automation. Older equipment is not as stable as newer production tools, but by integrating custom factory automation and a data management system, the risks associated with legacy equipment can be reduced.

ACKNOWLEDGEMENTS

The authors would like to thank Mitchell Schoenbrun for QNX2/QNX4 programming and hardware expertise. Frederick Pool, Bill Adams and David Pie supported the project with appropriate resources. Russell Hartzell and Robert Judd provided PMT related hardware and software support.

REFERENCES

[1] D. Brindza and T. Abshere, *Challenges of Equipment Support in a Factory With a Diverse Multi-generational Toolset*, 2013 GaAs MANTECH Technical Digest, pp. 351-354, May 2013.

ACRONYMS

PMT: Plasma Materials Technology
RF: Radio Frequency
ARCNET: Attached Resource Computer NETwork
RAM: Random Access Memory
SM: Send Message
OEM: Original Equipment Manufacturer
kB: kilobyte