



Selecting the Right “Tool” for Barcode Data Collection

What tool would you use to drive a nail into a piece of wood? A hammer of course! Ever see someone try to drive a nail with a pair of pliers or a rock? How’d that work out? Probably not as well as if they had access to the right tool. They probably didn’t have a hammer and just used whatever they could find. The results of any endeavor depend on the resources, or tools, available to do the job. The same can be said when selecting barcode data collection equipment. Before we get caught up in reading data sheets and comparing the display resolution and memory capacity of the myriad of handheld devices available today, we need to stop and assess the true needs. Look at the business process requirements and the technology infrastructure, and pick a device that will get the job done with the desired results but at the lowest investment cost and with as little disruption as possible. Minimize the risk of business disruption and maximize the ROI of the purchase.

Questions we should ask:

- 1** Once my data is collected, do I need to transfer it “real-time” to my host system? Or can it be transferred at a later time in a “batch” mode?
- 2** If a “real-time” data transfer is preferred, do I have a wireless LAN to facilitate communicating the data instantly from the handheld to the host system?

3

If I choose “real-time”, and I have the wireless infrastructure, what type of interface is required between the handheld device and the host software system? Does it require simple terminal emulation? Maybe a web browser? Maybe I need to load a piece of software, a “thick-client”, directly onto the handheld in order to communicate with my host software.

Once these questions are answered, we’re ready to embark on a discovery process to decide exactly what type of device would be best. The operative word here is “type”, because not all handheld devices are the same. It’s very easy to “overbuy” when it comes to technology because people tend to “future-proof” because it gets outdated so fast. Buy the biggest, fanciest widget with all the bells and whistles! But is that the wise thing to do? Spend money on what we *think* we might need five years from now?



TE

Windows®

Batch

Fundamentally speaking, there are three types of barcode data collection devices that should be considered when trying to select the optimal device. We’ll take a look at all three and consider the pros and cons of each.



AML M5900

Batch Data Collection

A batch data collection device stores data directly on the device. As the data is scanned or keyed in, it stays in the memory of the device until such time it can be connected to a computer and uploaded. Afterwards the data can be erased, freeing the memory of the device for its next use.

The upside to this approach is simplicity and cost. Because the information is stored on the device and then uploaded or “batched” to a computer over a wired connection, such as USB, there is no need for a wireless infrastructure (WLAN) in the facility. It can also be used in remote areas where wireless connectivity may not be an option at all. The device itself is lower cost than a Wi-Fi equipped device because there is no radio. For the same reason, battery life tends to be better as well.

One downside is that a small software program or application must be produced and loaded onto the device in order to generate the screen prompts telling the operator to scan or enter data. This can usually be accomplished with a program generation tool that can be used by non-programmers, but still takes some time and thought to properly execute.

Another downside to batch data collection is that the information is not available in real-time. This means that it is not available to be processed by the host software application until it has been uploaded to the computer, and then imported into the host package. Until the data is uploaded, it is still somewhat vulnerable should the device get damaged or destroyed.

Overall, batch devices are an affordable, easy way of collecting data for relatively simple applications such as physical inventory counts. However, they may not be the best choice for a receiving application in a high-volume, high-velocity distribution center where the information is needed on a more real-time basis.



AML M7220

Terminal Emulation

The use of wireless data collection (Wi-Fi) is the real-time alternative to batch data collection. The handheld devices can be almost identical, except that the wireless device is obviously equipped with a radio for communicating over the wireless LAN. Once we move into the wireless world, not all handhelds are alike.

Terminal Emulation is a methodology by which handheld computers can actually emulate “dumb terminals” when connected to a server over the WLAN. It is an extremely simple way of sending text-based screens down to handheld devices in order to provide information to the user and prompt them for scanned or keyed data. The only software application running on the handheld device is the software used for emulating the “dumb terminal”, hence the term “terminal emulation”. For the past three decades, this has been the method of sending and receiving data to handheld devices. It’s simple and reliable.

Most handheld computers are capable of running terminal emulation software, or TE software. In fact, some come equipped with the TE software already loaded. However, it’s important to know that not all handheld computers are alike. Some are specifically designed to be terminal emulation devices only, while others are more powerful Windows® devices that are largely overkill when running TE software.

A typical Windows®-based handheld computer will have a microprocessor speed of anywhere from 400 to 800 MHz and a minimum memory configuration of 128MB of RAM and 128MB of ROM. TE software can actually run efficiently on slower processors that require less memory and consume less energy (battery). Because the screens on TE devices are almost always text-based, a color display with touch panel is rarely necessary, further reducing the cost and energy requirements.

In short, handheld devices designed specifically to run TE software will always be less expensive and will also have better battery life. Dedicated TE devices are inherently simpler because typically they will only run the TE client, connect to a server, and login to the host application. The user is presented with no other options.

The downside to TE handheld devices is that it can only be a TE device. It can never run a true browser or any software developed for a Windows® environment, which means it can't run third party packages like device management software. If the chance exists that the host software application might change in the near future requiring a web browser or some other Windows®-based client, then a dedicated TE device might not be the best option for the long term.

The upside is their cost, simplicity, reliability, and battery life. If terminal emulation is likely to be the preferred method for connectivity for the foreseeable future, a dedicated TE device would be the simplest, most cost-effective solution.



AML M7225

Windows® Architecture

The vast majority of handheld computers available today come equipped with a variant of either Windows® CE or Windows® Mobile (now Windows® Embedded Handheld). These devices are quite literally computers in every sense of the word in that they can run software programs that were designed for their particular operating system and hardware configuration like screen size, keypad, etc. This software might be a “canned” Windows® application like Internet Explorer®, or third party applications like terminal emulation clients. It might also be custom software written to connect and communicate with a specific host software package running on a server. The point is that a handheld computer becomes whatever its application software makes it, be it a web browser, a TE device, a route accounting tool, etc.

The obvious upside is versatility. As long as the software exists or can be written, the Windows®-based handheld computer can become a powerful tool. They are generally available in a wide range of form factors from semi-rugged PDAs to super-tough handhelds and tablets designed to withstand even the worst conditions, including hazardous or even explosive environments.

The downside can be cost, and to some degree complexity. The software may or may not exist. So, it may have to be purchased or even written. It is a very common practice to run TE software on a Windows® device, but in most cases, the computing power is wasted. The device is in fact “dumbed down” when making it a TE device. Also, a Windows® device is not designed to be dedicated to one specific purpose. It may have to be “locked down” to remove options that the user doesn’t need to see or access. Taking a tool that is designed to be versatile, and then deploying it in such a manner that it is dedicated to a single task, is sometimes harder than it first appears.

Summary

The most important part of any discovery process is to first understand what options exist. Then, selecting the option that is the most ideal for meeting the specific needs of the application. Whether or not it makes sense to “future-proof” by purchasing equipment that might have more features and capabilities than what is currently necessary, is a business decision that has to be made on a case-by-case basis. It’s important to remember that any capital purchase will expect to produce a return on that investment. A lower investment equates to a faster return. It’s also important to remember that although technology advances rapidly and it’s hard to keep up with, sometimes business environments and processes can change just as rapidly. Sometimes the right answer is to buy what you need today, and worry about tomorrow, tomorrow.