Part Codes and Serial Numbers
The good, the bad and the ugly

By Michael Bloor, Seratel Ltd.

Only about half of the electronics manufacturing plants that I deal with have serial numbers, or licence plates, reel IDs or whatever you prefer to call them, on component reels.

During recent discussions with two small EMS companies. I was surprised to find they did not have their own part codes for components, relying instead purely on the vendors' codes. "What if you source the equivalent part from multiple manufacturers", I asked. "That's a problem", was the answer in both cases.

For any company about to implement a scheme of part codes or serial numbers, or perhaps revamp a legacy system, there is a lot to be learnt from the efforts of others.

How many "things" do you have?

Think about how many items you actually deal with. I have seen one plant with 250,000 component part codes and others using 1.5 million serialised reels per year.

How many "things" could you have in the future?

Now think about how that could reasonably expand in the future. There are numerous examples of "ample" ranges being exhausted. In the USA, telephone area codes are split. In the UK, a similar thing happened to London telephone numbers, which were subsequently changed a second time.

IP addresses are running out, so are ISBNs (the unique IDs for books). GPS time wrapped around and, of course, no-one saw the year 2000 coming.

Eight days a week (Hums a Beatles tune)

Note that I said above, "reasonably expand". Many years ago, I was asked to look over the IT operations of a medical devices company. They had purchased an IBM mini-computer and one person was a combination of IT manager, programmer and operator. IBM had impressed upon the company the need to allow for expansion in any file layouts that were created. It worked – all the files allowed for an eighth day being added to the week.

Atoms in the solar system

I worked with an EMS company that was in the process of putting serial numbers on all reels at receiving. These were made up (I think) of the date, GRN number, item number within the batch and more – probably the operator's lucky lottery numbers.

The result was a serial number of up to 31 characters. One side effect was that the bar code containing the number overflowed from the reel label, so became unreadable. Think about then being forced to enter a 31 character code manually. It takes long enough in the supermarket when an item won't scan and the operator has to enter a 12 or 13 digit ID.

Using an alpha-numeric serial, 31 characters could give a separate ID to every atom in the solar system (mail me mike@seratel.ie for details if you want). I admire ambition, but this is overkill.

Will it go in a database?

Of course, it will. Everything is going to go into a database. It's the backbone of any ERP or warehouse management system.

Design of part codes and serial numbers can have a significant effect on database design, and consequently the speed and efficiency of operation. Part codes will appear not just in a parts master table, but in BOMs, purchase orders, inventory transactions and much more.

Just a number

Databases love numbers. A 32-bit integer is compact, easily sorted, compared and indexed. It can provide over 2 billion possibilities (4 billion if we work with unsigned integers). These numbers make ideal keys for records in a database, but numbers as long as 10 digits aren't that easy for humans to use without error. They may be best left to the parts of the system that humans don't interact with.

Making it human friendly

If you do decide to mix numbers and humans, then format the numbers to make them easier to use.

For many people, 23-0955-61 is easier to remember or transcribe than 23095561. Internally, software and databases can work with the pure number. When it is displayed to a human, add the separators. When a human enters the code, software can remove any separators included.

Keeping it compact

If we want a marriage of compact and human-friendly, there are some alternatives.

Let's say we use a code of two upper-case, alphabetic characters followed by two digits. For example: HY63. This gives us a range of 67,600 things and a code which a human can easily remember, type or quote on the telephone.

We know that the first two characters are always alphabetic and the next two numeric, so if someone mistakes letter O for digit 0, or digit 1 for letter I, software into which the code is entered can be written to correct this automatically.

If we now add an extra alphabetic character on to the end and make it a "check" character, we can catch most other transcription errors. Say the scheme we use generates a check character of X for the above code. We require the operator to enter HY63X. This is still simple to remember, but if any character is mis-entered, 25 times out of 26 this will be flagged immediately.

Expanding that a bit, or a lot

If a range of 67,600 is not enough, and it probably won't be for serial numbers, consider the scheme we have implemented in our reel serialisation software. This uses digits 0 to 9 and upper-case letters A to Z, excepting O and I. Omitting these two characters means there is never a confusion between the digit 0 and letter O, or between 1 and I.

This scheme gives extremely compact serial numbers covering as large a range as required. 4 characters gives 1.3 million possibilities. 6 characters gives 1.5 billion.

Again, we can create software to force alphabetic characters to upper case, and automatically translate O and I to 0 and 1 to coerce operator input into a valid form.

Another company using a similar scheme is the alamy.com stock photography agency. They allocate each picture an image number. They have over 200 million images and a recently uploaded image of mine was given the ID 2BKTA2P - better than a 9 digit number, but still not as compact as the scheme described above.

Keeping it unique

Perhaps a serial number being unique within your plant is not enough. You may want it to be unique within your corporation or maybe world-wide.

For products produced for sale and distributed outside your organisation, you may be looking at an International Article Number (also known as a European Article Number or EAN) for product codes.

If you want to use a common format of serial numbers within your organisation, without overlap between different manufacturing sites, the scheme described in the previous section will work for you. If the corporation as a whole could never make more than 2 billion things, tell plant A to start their number at the equivalent of, say, 50 million. Plant B starts at 100 million and so on.

Alternatively, assign each plant a suffix using the same 34 character coding scheme. If you use a 2 character suffix, then 1,156 plants can each have their own range of serial numbers. It must be a suffix, not a prefix, to avoid two plants generating the same string. Any plant can then produce 1.5 billion items without the complete serial number being longer than 8 characters.

Does it mean something?

If there is information in a code that means something on its own, then make use of it.

I have a personal account with the same bank that holds our company account. To login to either account I have to use a 10 digit ID. That's much too long. They do not need a range of over 1 billion users. Not only that, but all IDs seem to start with the same four digits.

I sometimes mistakenly try to log in to the company account using their consumer portal, or vice-versa. I get a message such as, "You have attempted to login to Business using a Personal number". If the number tells them that, why don't they just take me to the correct web page and enter the number into that page for me?

They are not receptive to criticism of their web site, so there is no point in suggesting this. Otherwise I might let them know that I accidentally found a way around their two factor authentication.

Will it be worked out?

In some cases, it makes sense to have a "calculated" part code. As an example, imagine a manufacturer of boiler suits or coveralls. They produce a number of styles, each with a three digit code. Each style is made in standard or tall fittings. They come in a small range of colours as well as a range of sizes.

If I want a standard fit, navy blue, size 112 boiler suit in style 505, then I can order item S505N112. Such a scheme is easy to understand and works well for both the manufacturer and their regular, trade customers.

Sales staff can be provided with a screen showing stock levels for all combinations of style 505 in a colour/size grid format. This can be supported by using separate fields in a database for fitting, style, colour and size, with the four fields together forming the record's primary key and part code.

Leaving room for manoeuvre

Room for expansion can be limited by schemes such as this. It allows for 26 different colours. What happens if you need more? This might happen if the company expanded its range of clothing types.

The solution is to use an “escape character”. Rather than have a colour in use with code Z, Z is designated as the escape character. If the system sees a Z, it expects it to be followed by a second character. The scheme now supports 50 colour codes: A to Y and ZA to ZY. Note that ZZ is not used as a colour code. Rather Z continues as an escape code, so that ZZA to ZZY are available for future use if needed.

Should the customer understand it?

Codes such as the one just described can be explained to customers, perhaps as part of a catalogue. Where a large range of variations are available for a single product line, this can simplify ordering.

I purchased a selection of weather-sealed electrical connectors for use on my boat. For each connector, there was a choice of plug or socket and the number of pins. I could also choose a free plug or socket or any of several variations of bulkhead or panel mounting. A logically constructed part code for every variant, explained in the manufacturer’s catalogue, made it simple to order the correct items from many on offer.

If not, then -

Don’t put information into a part code or serial number that can be looked up using that part code or serial number. Don’t make a part code over long by including information describing the part. Let the code simply be a key that leads you to the rest of the information.

Who owns this code?

Within the EMS industry, I often see codes with a prefix or suffix denoting the customer that the component is used for. This can take the form XYZ-98-9876-47 or 27-0002-33-888, where XYZ and 888 are customer codes. If some customers consign material, that may suggest using such a scheme.

Wherever multiple codes can exist for the same component, there should be some way of recognising and managing equivalences. As some components will be sourced from multiple manufacturers, each with their own, differing part codes, this means that we need a many to many code translation mechanism. This should be computer based, presenting an operator with a list of only the valid options. Systems with manual steps will inevitably result in mislabelling of material and consequent rework.

You may also want a method of subsequently moving components from one part code to an equivalent. Will you abandon a build if it seems at first sight that you have a component shortage, but the required part is actually in house under another part code?

Will it go in a bar code?

Your part codes, and probably your serial numbers, will need to be printed as bar codes. The choice of symbology used will have to suit the characters used in the codes.

Many companies use Code 39 bar codes. They are often suggested as the “default” symbology but may well not be the best choice. The characters that Code 39 can encode are limited – Upper case A to Z, 0 to 9 and seven special characters. Code 128 can encode all 128 ASCII characters and is much

more compact, so easier to fit on labels. Data Matrix 2D bar codes are even more compact and add a level of redundancy, allowing damaged bar codes to be read successfully.

Don't be tempted to use Extended Code 39. It can encode all 128 characters but takes up even more space and will produce an incorrect input from any scanner not programmed to expect it.

Using prefixes in bar coded data –the letter P before any part code is a common example – can help software verify that the correct bar code is being scanned. This can be important where a label contains multiple bar codes.

Suppliers' codes

Don't trust other people's bar codes without checking them. I have seen numerous examples of labels where the data in a bar code has not matched the human readable version above or below the code.

Standards and composite codes

Don't you love standards? There are so many to choose from. Two that you should consider are the ECIA and ISO15434 formats for (usually) 2D bar codes containing multiple fields. They provide the ability to scan a single bar code and read several fields - say part code, date code and serial number, in one operation.

Within the composite code separators delineate individual fields and each field starts with an ID (again perhaps P for part code) which shows its purpose.

In practice, we have found two problems with these bar codes. Firstly, different companies will use different IDs for the same field. Whereas many may use P for part code, we have seen one using 17P. Our solution to this has been to use a database table to translate all IDs to a common one used internally. This has the advantage that multiple variants, from different suppliers, can be handled automatically.

The second problem is with badly constructed codes. We have seen codes using incorrect or misplaced separators. Intelligent decoding software can cope with this.

Where software reads multiple fields at once through a composite bar code, it may be important to control the order in which the fields are processed, which may be different to the order in which they come in the bar code. For example, the software may need the part code first, to load relevant information from a database before other data can be sensibly handled.

Just need one field out of several?

A composite bar code can always be queried for just one of the fields it contains. For example, we created a web application including pages that allow the operator always to scan the same bar code, but with each page extracting only the specific information that it needs.

Will things change?

It is wise to maintain both forward and backwards compatibility with these composite codes. That way, if new software reads an old bar code, it can either supply a sensible default or ask the operator for any data it is missing. Conversely, if a newer format of bar code adds extra data, that can simply be ignored by older software.

Expect to have to deal with multiple standards. For example, International Standard Book Numbers were originally 9 digits in length, but from 2005 migrated to a 13 digit system compatible with EAN. Any software for cataloguing books based on the ISBN needs to accept either format.

Conclusion

The important conclusion is – part codes and serial numbers should be designed, not just happen. They have long term implications and good design can make a huge difference to the efficiency of any organisation.

Seratel Ltd.

Computer Systems for Electronics Manufacturers
Consultancy and Bespoke Software Development

www.seratel.ie
info@seratel.ie
+353 98 36244