Working with MICR Readers

Tanmay K. Mohapatra

What is MICR

MICR stands for "Magnetic Ink Character Recognition" - the technology behind the production and usage of the string of special characters most commonly seen printed below bank checks. Before we delve deeper to understand the details of MICR, it would help to trace through the history of the now ubiquitous banking check.

History

In the early days of money and banking, trading and exchange of money was usually limited to small localities [1]. People mostly dealt with cash. On rare occasions where the transacting parties were at relatively far away localities money orders were used. With time population grew, towns grew into cities, industrialization took over and not every place could be self sufficient in all respects. The need to trade with places that were much farther also grew. As it was not very convenient to arrange for money orders for every trade, banks needed to come up with new ways of dealing with money.

The checking account thus came into being. Initially this facility was probably available for only select people. But with time, this was needed by more and more people as a necessity rather than convenience. As the volume of check transactions increased, the banks felt the need to automate processing of checks.

Initial check sorting machines were mechanical [2,3]. Two commonly used machines were called the “Sort-A-Matic” and “Top Tab Key Sort”. However, mechanical sorters were slow and often resulted in damage to the check itself. With the advent of cheaper electronics and computers, people began to look at their use in sorting of bank checks was only logical. Thus was born the MICR technology, a result of the attempts of Stanford University and Bank of America.

Technology

MICR characters are printed using an ink laden with iron oxide particles. Iron oxide has magnetic properties and can retain magnetic fields when it is applied on it. The working of a MICR reader is essentially based on the concept of moving characters printed with this magnetic ink over two magnetic heads, one that charges the characters and the second one that immediately follows the first and reads the magnetic charge. The pattern of the electrical field is what determines the character being read. The

characteristic shape of the MICR font is designed to give a unique electrical signature pattern to each character which can be easily recognized by the machine with minimum ambiguity and maximum tolerance.

> Another related topic of interest is the very common bar code we so often see on items on the shelves of a supermarket. Bar coding utilizes a technique is similar to Morse code – a series of narrow and wide bars make up for one character [4]. The reader contains a photo diode and a light/laser source. The photo diode measures the intensity of light as the light source is moved across the bar code. The waveform of reflected light thus produced is decoded to read the contents.

There are multiple MICR fonts that are in use today – E13B which is used in US, and Asia, and CMC7 which is used in Europe and parts of Latin America. Standards and specifications have been developed that specify how a check should be printed and the exact dimensions of the MICR fonts [5,6].

Interfacing a MICR reader to an application

MICR readers are typically supplied with different kinds of interfaces to connect to a PC – keyboard wedge, RS232 or Ethernet network.

The *keyboard wedge* connects to the keyboard port of the PC and sends the MICR data directly to the application as if the same was typed through the keyboard. Therefore, if the application is designed to accept MICR code lines directly as input, it becomes extremely to use this kind of interface. The drawback of this kind of interface is that the data can not be processed before being sent to the application. Moreover the user has to make sure to place the cursor on the correct text field, otherwise the data will get inserted into the wrong text field. The *RS232 or serial interface* gives more flexibility. However, it is slightly more complex as one has to develop a separate program to perform the RS232 communication with the device. Modern devices are capable of interfacing over an *Ethernet network*. They can connect directly to the LAN, obtain an IP address through DHCP and communicate with their counterpart that runs along with the application on the PC.

MICR readers are capable of taking an image of the check for archival. They can also be interfaced with a character recognition engine that can provide additional verification of the MICR line through OCR as well. Character recognition engines can also be used to recognize handwritten amounts on the checks [7].

To simplify programming, device vendors usually supply a *Windows ActiveX component* that performs the RS232 or Ethernet communication internally and provides a simpler interface to the application developer. Sample programs provided along with this article demonstrate how to interface MICR devices from MagTek [8] and Unisys [9] to a .NET

based application using both the direct RS232 communication and the ActiveX mechanism.

Standards in use today

J/XFS (Java extensions for financial devices) standard [10, 11] promises to provide an encapsulation of most kinds of financial devices for the Java environment. However, J/XFS is not widely supported among vendors as of today. The device vendors usually provide specific API sets for their devices. In recent years however, many device vendors have unified the API used across their own devices and have come up with unified API. For example, NCR provides a API set called WiseIP that can be used to all MICR devices provided by NCR. Similarly, Unisys provides unified C APIs for interfacing with Unisys MICR readers. Magtek, Unisys and NCR also provide unified interface through a common ActiveX control. More details on these APIs can be found at their website.

References

1. http://www.asapchecks.com/micr/a-checkitout.htm
2. http://www.sullivanpress.com/online_museum.htm
3. http://www.rich.frb.org/econed/museum/
4. http://www.taltech.com/TALtech_web/resources/intro_to_bc/bcpwork.htm
5. http://www.apca.com.au/Public/apca01_live.nsf/WebPageDisplay/publications_and_products?OpenDocument
6. http://www.cdnpay.ca/rules/acss_standards.asp
7. http://www.unisys.co.uk/products/payment__systems/amount__recognition/
8. http://www.magtek.com/
9. http://www.unisys.com/products/payment__systems/
10. http://www-106.ibm.com/developerworks/java/library/j-jxfs1/
11. http://www.jxfs.com/

Tanmay K. Mohapatra *is a Technical Architect based at Bangalore, India.*