The Introduction of QR Codes in Production Processes

Marat R. Khamidullin1*, Rifkat G. Mardanshin1, Alexander V. Prozorov2, Ramil I. Karimov2

1Kazan Federal University, Naberezhnye Chelny Branch, Naberezhnye Chelny, Russia
2Federal State Budgetary Educational Institution of Higher Education “Kazan National Research Technical University named after A.N. Tupolev – KAI”, Kazan, Russia

Received: 13/09/2019       Accepted: 22/11/2019       Published: 20/12/2019

Abstract
This paper is devoted to improving the ways of working with QR codes in enterprises. The approaches outlined in this paper are aimed at increasing the speed and accuracy of work in the field of accounting technology by reducing the number of man hours needed to draw accompanying documentation, and in the field of equipment related to a specific task. In this case, the goal will be to create a program for working with personal computers.

Keywords: QR code, Scanner, Database, Computer

1 Introduction
The concept of “QR code” (Quick Response code) arose for the first time in 1994 in Japan (6, 22). This standard for barcodes was developed and introduced by Denso-Wave Company to track the various stages of production of Toyota products in its enterprises. Soon, this standard has gained great popularity among advertising and marketing companies, as QR codes made the process of human interaction with an object more interactive (9, 11). The simplicity and popularity of QR codes made it possible to use these two-dimensional bar codes in all areas of human life, as well as in various industries. The QR code is a matrix (two-dimensional) barcode that allows much more information to encrypt than conventional one-dimensional barcodes (for example: “EAN / UPC”, “Code 128”), and also allows various types of data (numbers, symbols, hieroglyphs, as well as mixed data types) to store. This barcode standard has gained great popularity due to the fact that the recognition process does not require special scanning equipment, such as a special scanning beam, and can be recognized using any camera and decoding software installed on the device. Due to this, QR codes are used in almost all spheres of human life, such as trade, marketing, logistics, advertising, leisure, public services, tourism and many others (5, 12, 17, 18). In addition, the QR code standard has several levels of error correction, which makes their use even more reliable and in demand.

For more than 20 years, QR codes have begun to be actively used in our lives. Their extensions include product tracking, item identification, time tracking, document management, and general marketing (1, 7, 8). Thus, it could be not just a replacement for UPC, but also, given the speed of reading, for parts of handwritten text. This means that various companies could use it to facilitate work with their products. A computer and even a phone can create and read a QR code in seconds, and algorithms for these actions can be obtained from Open Source projects and repositories (2, 9).

2 Methods
In the past few decades, mankind has begun to strive to introduce new technologies faster and faster, which would ensure movement in step with the times. One of the best systems for development of applications in this case is a spiral pattern. It will give us an opportunity to quickly get an application prototype for integration. And by the time the implementation is complete, there will already be a release version of the product. Therefore, we need to look for the language and Open Source repositories, with the help of which it will be possible to conduct such development. Languages such as C # or Python will provide high development speed. The C ++ language would help us ensure a high speed of the application, but since our goal is to get a prototype in the shortest possible time, and also considering the high speed of computers with QR codes, using this language becomes meaningless (3, 21). Although Python allows us to reduce the time for developing a software product, a rather large amount of time will have to be devoted to creating a user interface, and in the case of C # we could use Windows Forms, which would minimize the time for developing an interface (4, 8, 19, 20).
3 Results and Discussion

In all existing programs that read and decode QR codes, a simple algorithm for detecting a QR code in the image received from the camera is implemented. Then, the standard procedure for decoding information from a QR code is implemented (3). However, this recognition algorithm requires a very clear positioning of the specially allocated area on the shooting device and a certain location of the QR code in space. After the specially defined area on the shooting device clearly matches the faces of the QR code, three positioning marks are searched for, which were discussed in the previous section. These marks are located strictly in certain places in the selected image area on the recording device. The disadvantage of this approach is that the QR code cannot be in any area on the image. A user himself / herself must first focus on the necessary QR code, and make sure that the area for shooting the QR code coincides with the QR code itself, which must be decoded (5).

In order to scan a QR code, the program must first "see" it. Since enterprises around the world are introducing Wi-Fi-connected points integrated in a network intended for communication between employees, this is an excellent way of wireless communication between machines for their joint work or for quick and remote data collection by employees from machines.

Given the above, it makes sense to broadcast an image to the local network. To do this, we need a device with a Wi-Fi communication module and a camera. It could be a smartphone, an IP camera, or even a barcode reader. In the case of the latter, it is necessary to purchase equipment, and in the case of smartphones, which nowadays are owned by the majority of the population of our planet, there is no such need. Even a HD quality picture in is suitable for reading a QR code. To start the broadcast, an employee only needs to download the free application, available, for example, on Google Play (10-13). The next step is to select the quality of the transmitted image. You should set the resolution to about 640 x 480 or 1280x 720, and the picture quality would be from 65 to 90 per cents (the resolution required for setting and picture quality varies depending on the quality of the camera). Then, by clicking the “Run” button, we can see the transmitted image and the IP address of the camera.


To do this, we just need to designate the class:

| and then display the image in a PictureBox (pictureBox1 in this case), obtained from some TextBox text field (textBox1 in this case): |
| pictureBox1.Image = brcode.Draw (textBox1.Text, 50); |

where 50 is the maximum height of the QR code.

We can save the image or print it using the Save () and Print () methods built into C#, respectively.

Since the image printed and glued onto the computer case should be read, it is necessary to use some kind of camera. Since a lot of people in everyday life are not to be without a smartphone, and enterprises actively install Wi-Fi points for communication between mills or personnel, it will be convenient to use a smartphone camera connecting to it via a local network. Since we will work through an IP camera, it is necessary to decide how the image will be transmitted. The main transmission methods are the following: the camera receives a picture at any interval and uploads it to the server (JPEGStream) or video is being broadcast (MJPEGStream).

Conventional code readers operate according to the first principle, but for phones it is better to use the second. It’s best to use the Open Source repository “AForge.Video”, which is also available for integration into the project in the “NuGet Package Manager”.

To obtain an image, it is necessary to designate its class, and also give it a link where the broadcast will be conducted, transfer a set of event handlers and, of course, start receiving the image:

| MJPEGStream stream = new MJPEGStream (“some url”); |
| streamNewFrame += new NewFrameEventHandler (video_NewFrame); |
| streamStart (); |

Some cameras transmit data through an HTTP header that is not compliant with the standard, which results in an exception in. NET. To avoid this exception, you must set the useunsafeheaderparsing httpWebRequest configuration parameter, which can be done using the application configuration file (7-16).

In order to display an image in a PictureBox, we must create a method by which the class will be indicated and the picture will be displayed:

| void stream_NewFrame (object sender, NewFrameEventArgs eventArgs) |
| { |
| Bitmap bitmap = (Bitmap) eventArgs.Frame.Clone (); |
| pictureBox1.Image = bitmap; |
| }

The last step remained is reading the QR code from the picture. To do this, the ZXing.Net repository, which is also available on NuGet, will be used. It supports such formats as UPC-A, UPC-E, EAN-8, EAN-13, Code 39, Code 93, Code 128, Codabar, ITF, QR Code, Data Matrix, Aztec, PDF 417, MaxiCode, RSS 14, RSS-Expanded (6-15).

To process the image, we should create a timer that receives the image once a second and works with it. For code recognition, it is necessary to designate classes and pass the picture to the handler:

| BarcodeReader Reader = new BarcodeReader (); |
| Result result = Reader.Decode ((Bitmap) pictureBox1.Image ); |

The “result” variable will contain the data encrypted in the QR code, which can then be, for example, written to a table or displayed in a file using the built-in functions and methods.
The last step left is to process the received data. Typically, such data is collected to filter out various kinds of reports, or the goal may be to fill out / modify the database. For all these cases, there are ready-made libraries that it makes sense to use.

In the case of working with documents of Word type, there is a namespace Microsoft.Office.Interop.Word. The task will be to write all received codes to a file that can be prepared in advance or created in the application itself. Since it is more convenient and faster to work with an already prepared document, we should review this method. Since the received codes are most likely in an array, it is necessary to combine these codes into one variable, or assign this data to the document directly. The second method is more convenient, since it will use the same amount of power, but will not occupy too much space in the computer's RAM. For this work, it is best to use the replacement method (4). To implement it, we must run the following code:

```
for (int i = 0; i < q; i++)
    if (i < q - 1) FindAndReplace(wordApp, "[id]",
                      database[i]+ "@[id]");
else FindAndReplace(wordApp, "[id]", database[i]+ 
                      ");
```

where q is the number of elements in the array, database [] is the array in which the codes are stored, and [id] is the element to replace.

Consider working with a database. The easiest way is to work with SQL queries. After all, it’s quite simple: we just need to perform regular SQL queries in the database associated with the project to add / modify / delete data in the table. An example of connecting a database and introducing the received data into it (1):

```
string connString = "% connection_data%;
using (SqlConnection conn = new SqlConnection
               (connString))
    for (int i = 0; i < q; i++)
        if (i < q - 1) FindAndReplace(wordApp, "[id]",
                       database[i]+ "@[id]");
else FindAndReplace(wordApp, "[id]", database[i]+ 
                       ");
```

4 Conclusions

Based on the work done, we can conclude that it is now very beneficial to start the widespread integration of QR codes. The reason is that the creation of applications for working with them takes an extremely short period of time, and since QR codes can only be read in a unique way, unlike handwritten text, this will reduce the number of errors and also ease the work load on staff. This will positively affect the emotional state of the staff and the speed of work in general. QR code recognition is not ideal today. In order for the QR codes to be correctly recognized, it is necessary that the recognized QR code image to be parallel to the camera lens. In addition, the user must place a special limited area (most often, limited by the corners drawn on the screen of the reader) directly above the recognizable QR code image. The problem is that QR codes can be recognized only with a certain camera location relative to the barcode itself and with a certain QR code location in space.

5 Acknowledgements

The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

References


