

# Zebra TC-21 QR Scanner Set-Up

Johnjimy K. Som

June 8, 2021

Documentation to set-up a Zebra TC21 to scan QR Barcodes in hexadecimal, store in a .csv or .xlsx file, send the files into an SMB network, and parse in decimal to determine item codes from that scanned QR code. Note: *Italicized Text* means it can be seen in the Zebra Device.

## 1 Zebra TC-21 Configurations

Requirements to scan QR code:

- Zebra TC21 [Model:TC210K], Android Version 10
- Scanning Application
- File Explorer Application
- Server Message Block (SMB) configuration skills

### *Data Wedge* Application

*Data Wedge* is a built in application for Zebra devices, In order to have the scanning function to perform quickly and efficiently, the following changes needs to made:

- In the *Data Wedge* application, Profile:Profile0 (default) should be selected
- Scroll down to *Keystroke output* section, 'Inter character delay' is up to [10 ms]
- Scroll down and select *Basic Data formatting* section
- 'Send ENTER key' should be [✓], not unchecked

## 2 'Orca Scan' Android Package (APK)

*Orca Scan* is an application created by Cambridge App Lab Limited. *Orca Scan* will make the TC21 device: scan, export .csv, .xlsx to a file manager application that connects to an Server Message Block (SMB) service.

- In the *Data Wedge* application, Profile:Profile0 (default) should be selected
- Scroll down to *Data Wedge* section, 'Inter character delay' is up to [10 ms]
- Scroll down and select 'Basic Data formatting' section
- 'Send ENTER key' should be [✓], not unchecked

### 3 Parsing QR Bar-code

```
1  #include <iostream>
2  #include <string.h>
3  #include <string>
4  using namespace std;
5
6  string GetBinaryStringFromHexString(string);
7
8  int main(int argc, char *argv[]) // input is a 32
   character string
9
10 {
11     unsigned int encodeVer, printArea, itemCode, packingDiv,
        productionYear, quantity, serialNumber, checkSum;
12     string inHex, inBinary;
13
14
15     for (int i = 1; i < argc; ++i)
16     {
17         inHex = argv[i];
18         inBinary = GetBinaryStringFromHexString(inHex);
19         //cout << inHex << "\n";
20         //cout << inBinary << "\n";
21
22         ...
23         cout << "Serial Number: " << serialNumber << "\n";
24         cout << "Checksum: " << checkSum << "\n";
25     }
26
27     return 0;
28 }
29
30 string GetBinaryStringFromHexString(string sHex)
31 {
32     string sReturn = "";
33     for (unsigned int i = 0; i < sHex.length(); ++i)
34     {
35         switch (toupper(sHex[i]))
36         {
37             case '0': sReturn.append("0000"); break;
38             case '1': sReturn.append("0001"); break;
39             case '2': sReturn.append("0010"); break;
40             case '3': sReturn.append("0011"); break;
41             case '4': sReturn.append("0100"); break;
42             case '5': sReturn.append("0101"); break;
43             case '6': sReturn.append("0110"); break;
44             case '7': sReturn.append("0111"); break;
45             case '8': sReturn.append("1000"); break;
46             case '9': sReturn.append("1001"); break;
47             case 'A': sReturn.append("1010"); break;
48             case 'B': sReturn.append("1011"); break;
49             case 'C': sReturn.append("1100"); break;
50             case 'D': sReturn.append("1101"); break;
```

```

51         case 'E': sReturn.append("1110"); break;
52         case 'F': sReturn.append("1111"); break;
53     }
54 }
55 return sReturn;
56 }

```

Listing 1: Snippet of the C++ to decode the bar-code

## I. Decode Bar-Code: Visual Studio

*Data Wedge* is a built in application for Zebra devices, In order to have the scanning function to perform quickly and efficiently, the following changes needs to made:

- In the *Data Wedge* application, Profile:Profile0 (default) should be selected
- Scroll down to *Keystroke output* section, 'Inter character delay' is up to [10 ms]
- Scroll down and select *Basic Data formatting* section
- 'Send ENTER key' should be ☒, not unchecked

## II. Decode Bar-Code: Future Updates

Future updates regarding to the parser is a built in application for Zebra devices, In order to have the scanning function to perform quickly and efficiently, the following changes needs to made: