NXP Semiconductors

User's Guide

Document Number: QRDEMOUG

Rev. 1, 07/2016

QR Demo User's Guide

1. Introduction

The QR decoder demo shows how to decode a bar code or QR code. This example is based on the TWR-K80 and TWR-LCD module.

The QR decoder demo acts as a bar code and QR code scanner which recognizes symbologies from an image, then analyzes and displays the decode result.

This document describes the functionality of the QR decoder demo and how to run demo applications.

Contents

2. Features

1. Embedded decoding algorithm for many popular symbologies
2. Real time camera image display
3. Project based on IAR® Embedded Workbench®
4. Kinetis Tower System based
5. Minimum decode algorithm RAM/ROM cost: 84K/110K (QR decode only)
6. Camera and LCD display resolution: 320x240.

3. Overview

3.1. Description

This demo application is based on the TWR-K80 and TWR-LCD, and is intended to provide the proof-to-concept QR decoder by using Kinetis K8x series.

At power up, the demo displays the initialization screens. The application then begins the continuous CMOS sensors acquisition task. The user puts an example QR image like the one featured in this document in front of the CMOS sensor. After successful detection, the program begins decode processing and then displays the result and information.

The software performs the following tasks:

1. CMOS Sensors image capture task, managing the acquisition of image data
2. Image processing
3. QR and 1D code decoding
4. LCD display.

3.2. Supported symbologies

- Supported 1D code:
  - EAN8, EAN13, EAN128
  - UPC-A, UPC-E
  - ITF-6, ITF-14, Interleaved 2 to 5
  - CODE39, CODE128
  - CodaBar
- Supported 2D code:
  - QR code

4. Hardware Configuration

4.1. Hardware requirements

The following boards will be used in this demo:

1. TWR-K80
2. TWR-ELEV
3. TWR-LCD
4. A reworked TWR-PROTOTYPE or a custom PCB board which can connect to OV7670 or Hi708 CMSOS module.

4.2. Hardware setup

This section list all pin assignments in this demo, some of these assignments may need rework in order to route the signal to the TWR-ELEV and TWR-LCD board. For camera connection, it is recommended to use a tower prototype board to connect to OV76770 or Hi708 module as shown in *Figure 1*.

The document top-level information is the *metadata* about the document. The metadata information defines the document identification number (DocID), revision number, revision date, title, document type, and copyright date. The document's headers, footers, table of contents, and the back page are also categorized as metadata and the instructions to complete these are provided in this section.

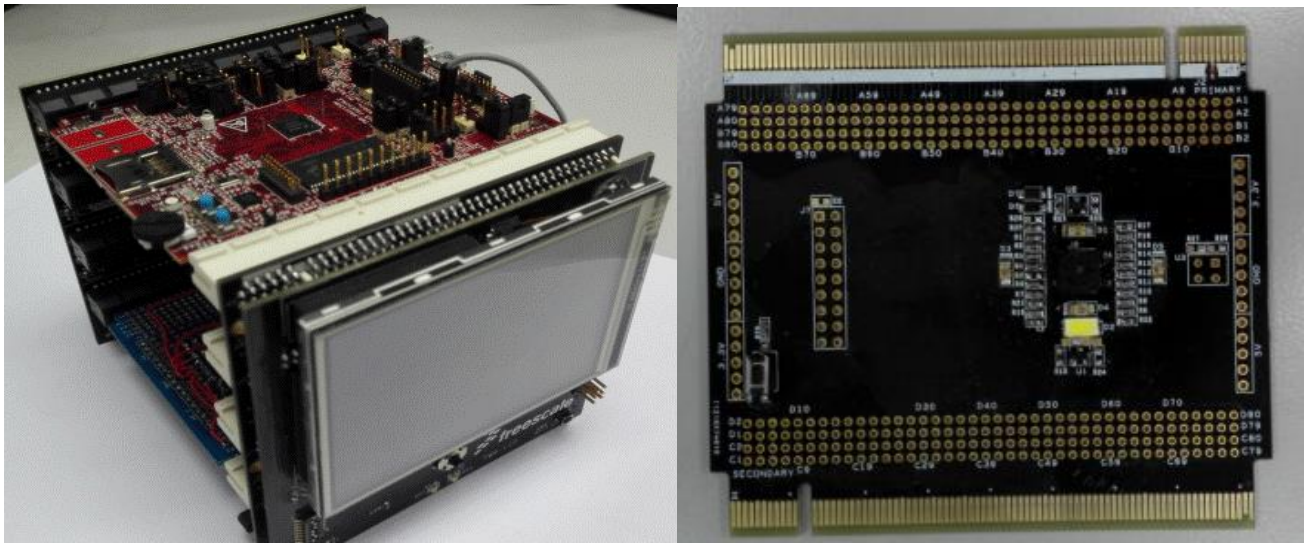

Figure 1. QR decoder demo hardware

All pin assignments are listed in *Table 1*:

Table 1. **Pin assignments**

Function	MCU pin	Mux	TWR-ELEV	OV7670	Comments & Reworks
UART	PTA14	LPUART0_TX	N/A	N/A	Connect to an TTL-UART terminal
Key	PTA21	GPIO	N/A	N/A	1D decoder trigger button
CAMERA	PTD8	FLXIO_D24	A7	D0	—
CAMERA	PTD9	FLXIO_D25	A8	D1	—
CAMERA	PTD10	FLXIO_D26	C38	D2	R295 populated
CAMERA	PTD11	FLXIO_D27	C37	D3	R299 populated
CAMERA	PTD12	FLXIO_D28	D40	D4	R113 populated
CAMERA	PTD13	FLXIO_D29	D39	D5	R112 populated
CAMERA	PTD14	FLXIO_D30	D38	D6	R111 populated
CAMERA	PTD15	FLXIO_D31	D37	D7	R110 populated
CAMERA	PTA12	FLXIO_D18	B62	HREF	—
CAMERA	PTA1	FLXIO_D11	B37	PCLK	Pixel clock signal, R34 populated
CAMERA	PTA10	I2C2_SDA	C8	SIOD	Camera SCCB interface
CAMERA	PTA11	I2C2_SCL	C7	SIOC	Camera SCCB interface
CAMERA	PTA13	FLEXIO_D19	B61	VSYNC	—
CAMERA	PTC3	CLKOUT	A64	XCLK	R143 populated
LCD	PTB18	FB_AD16	B67	N/A	—
LCD	PTB17	FB_AD15	B66	N/A	—
LCD	PTC0	FB_AD14	A66	N/A	—
LCD	PTC1	FB_AD13	A67	N/A	—
LCD	PTC2	FB_AD12	A68	N/A	—
LCD	PTC4	FB_AD11	A69	N/A	—
LCD	PTC5	FB_AD10	A70	N/A	—
LCD	PTC6	FB_AD9	A71	N/A	—
LCD	PTC7	FB_AD8	A72	N/A	R30 removed
LCD	PTC8	FB_AD7	A73	N/A	—
LCD	PTC9	FB_AD6	A74	N/A	—
LCD	PTC10	FB_AD5	A75	N/A	—
LCD	PTD2	FB_AD4	A76	N/A	—
LCD	PTD3	FB_AD3	A77	N/A	—
LCD	PTD4	FB_AD2	A78	N/A	—
LCD	PTD5	FB_AD1	A79	N/A	—
LCD	PTD6	FB_AD0	A80	N/A	—
LCD	PTC11	FB_RW	B71	N/A	—
LCD	PTD0	FB_ALE	B63	N/A	—
LCD	PTD1	FB_CS0	B64	N/A	—
SDRAM	—	—	—	—	SDRAM pin assignment is same as schematic of TWR-K80 (RevB), no rework is needed.

The TWR-K80's jumper settings are as follows:

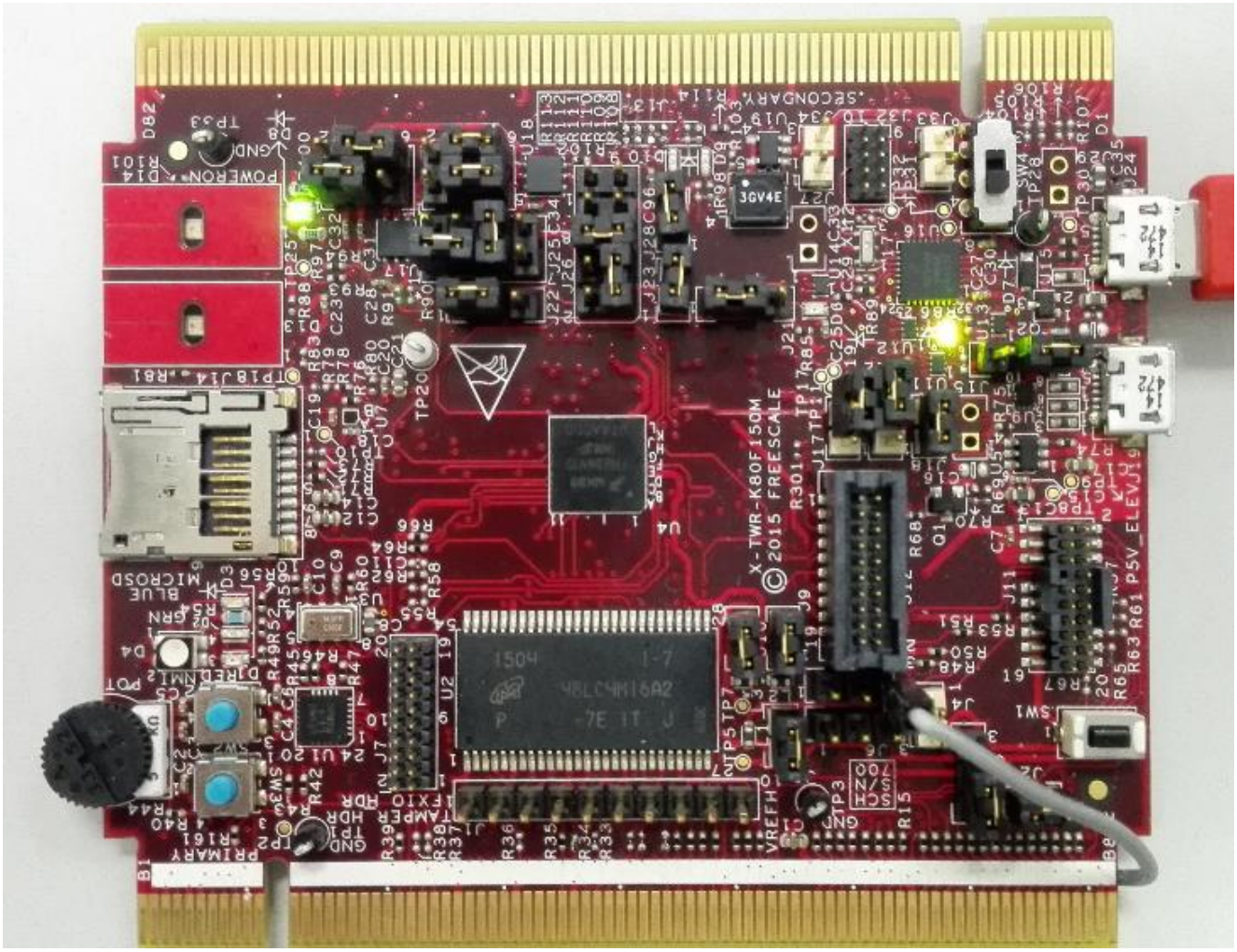

Figure 2. TWR-K80 jumper settings

5. Running the Demo

5.1. Installing the demo

This section features steps required to install and test the demo application. The following steps help you to install, build, and run the QR decoder demo:

1. This demo is developed based on KSDK1.3 for K80. Download the latest version of KSDK V1.3:

   http://www.nxp.com/products/software-and-tools/run-time-software/kinetis-software-and-tools/development-platforms-with-mbed/software-development-kit-for-kinetis-mcus:KINETIS-SDK?tid=redKINETIS_SDK

2. Launch the downloaded executable and click through the installation dialogues for the SDK. The SDK should install to an <Install_dir> similar to: "C:\Freescale\KSDK_1.3.0\"
3. The C source code for this demo can be licensed from NXP for use with any NXP products. Contact your NXP sales representative for the license agreement. After you have obtained this source code package, unzip and put in parallel with KSDK install folder as shown in *Figure 3*.

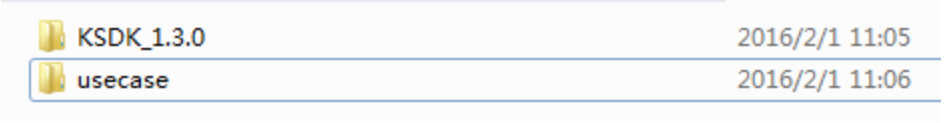

Figure 3. QR decoder demo hardware

4. Within the "usecase" folder, the pre-built project folders for the supported tool can be found (IAR). The IAR workspace file can be found in the qrcode_zxing\usecase\qrcode_zxing\iar folder.

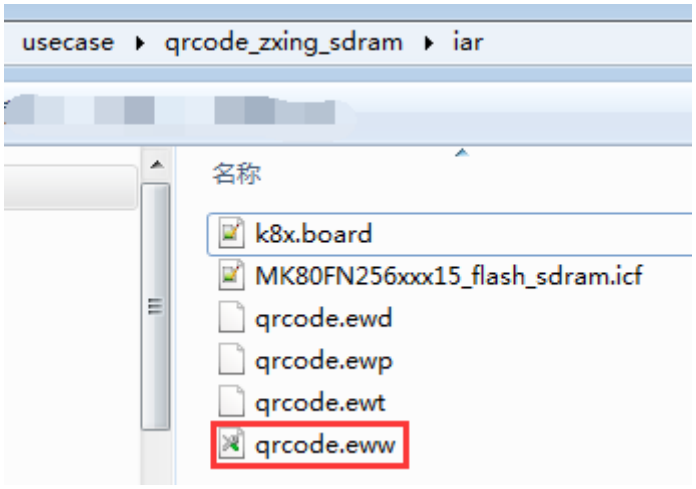

Figure 4. QR decoder demo hardware

5. Build the project and download the target board.

5.2. Download the binary

The demo binary image is attached in the same folder with this documentation. All provided binaries were compiled with the IAR tool chain and with flash release configuration.

The binary file name is as follows: qrcode.bin

The binary files can be downloaded directly into the Flash memory by using CMSIS-DAP. Use a USB cable to connect between TWR-K80's J24 and PC host, after the USB enumeration is completed. An MSD disk will appear on your PC, then, drag qrcode.bin into MSD disk to finish programming.

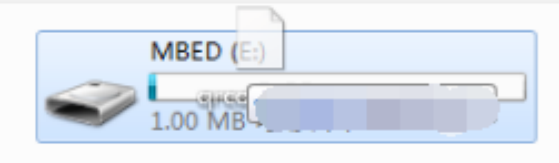

Figure 5. QR download binary image into the target

After downloading is complete, press the reset button to run the demo.

5.3. Decoding

The board acts like a camera. Once the valid image is captured, the information will be displayed. The demo program will try to find the pattern and decode it atomically. If the decode is successful, the result prints on the screen and the program stops temporarily to display the decode result.

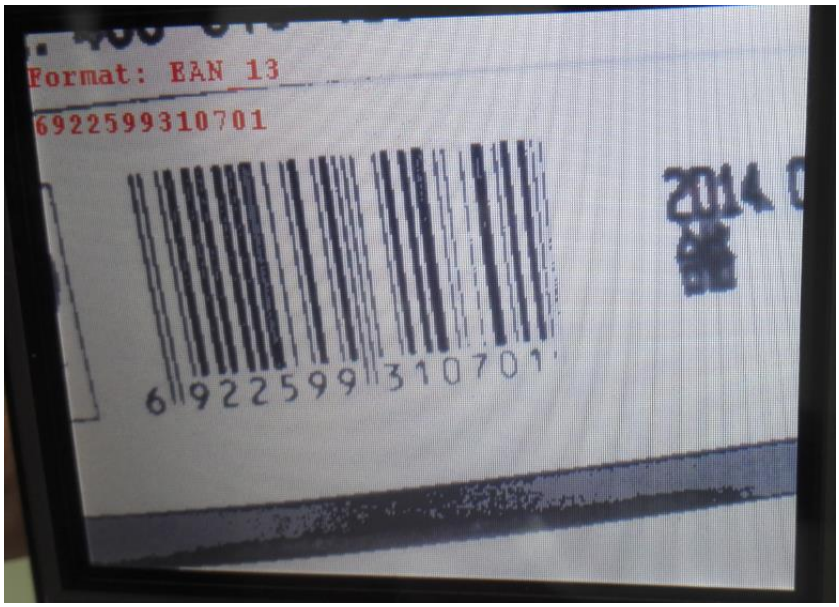

Figure 6. Bar decoding

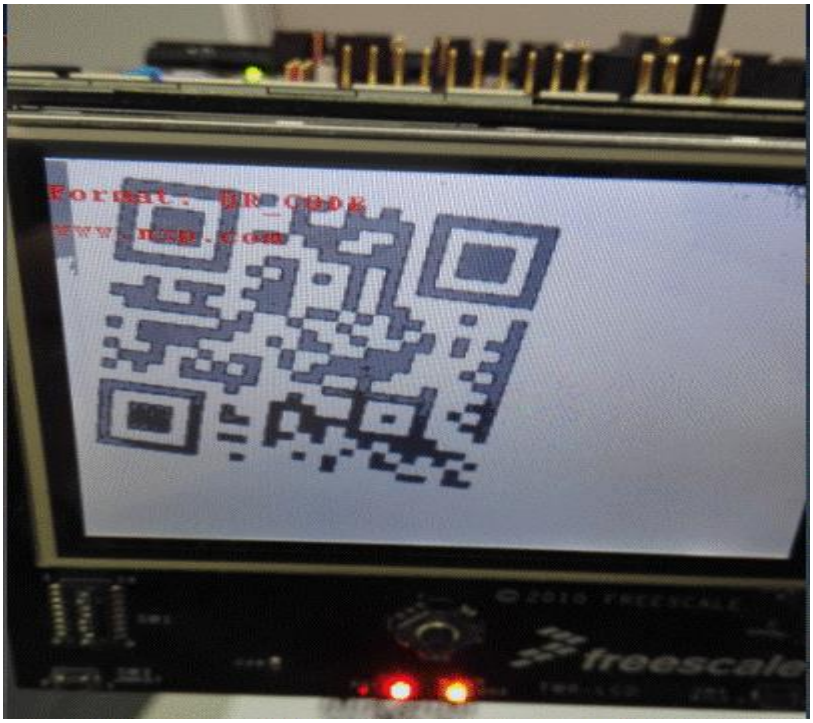

Figure 7. QR code decoding

6. Examples of QR Code and Barcode

The following figures show examples of an EAN-13 barcode and a QR code:

Figure 8. EAN-13 barcode

Figure 9. QR code

7. Known Issues

1. The Hi708 sometimes cannot initialize successfully if the power supply is inefficient.
2. Decode may fail after running for long periods of time due to memory leakage.

8. Revision History

Table 2. **Sample revision history**

Revision number	Date	Substantive changes
0	03/2016	Initial release
1	07/2016	*Section 2* point 1 updated *Figure 1* updated Note in *Section 5.1* removed *Section 5.2.1* and *section 5.2.2* removed and replaced with *section 5.3* *Section 7* point 1: “OV7670s” replaced by “Hi708”

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: QRDEMOUG
Rev. 1
07/2016