

# Connecting Kinetis MCU with CMOS Sensor Interface through GPIO

## QQVGA image transfer to Kinetis internal SRAM

by: **Adi Shieber and Yan Vainter**

### Contents

## 1 Introduction

This application note explains how to connect a Kinetis MCU to CMOS Sensor through GPIO to receive image data in memory. The application can be implemented on Kinetis MCUs and CMOS sensors with CSI (CMOS sensor interface) gluelessly, without any external hardware. In this case the implementation was done by using Tower evaluation board for Kinetis K60 and CMOS image sensor. The compatibility of Kinetis MCUs allows this application to be implemented on different sub-families of Kinetis other than K60. The fact all Kinetis K series shares the same modules (IP blocks) enables easy reuse of the code across the entire K sub-families. Examples of target applications for this application note are security systems and other consumer products focusing on low cost and low power.

1	Introduction.....	1
2	Signals and Connections.....	1
3	Implementation Overview.....	4
4	Conclusions and further options.....	9
A	Hardware Connections.....	9
B	Software.....	11

## 2 Signals and Connections

There are four types of signals involved in creating the CMOS sensor interface between the MCU and the camera (CMOS sensor camera). The first type is the power supplies that supply power to the camera. The second type is the camera configuration signals that are used to configure the work mode by using I2C interface. The third type is the external clock for the camera produced by the MCU and the fourth is the camera image transfer control and data.

## 2.1 Power supplies

The Kinetis Tower board supplies the camera with VDD and GND connections.

## 2.2 Camera configuration signals

This type of connection to the camera is used to configure the CMOS sensor. For the specific camera used in this application note, the control interface is I2C. It configures the camera registers to the desired work mode. However, other cameras may use SPI interface which is also supported by Kinetis K series.

The following table shows the Kinetis and camera signals used in the camera configuration stage.

**Table 1. Camera configuration signals**

CMOS sensor signal	CMOS sensor I/O	Description	Kinetis signal	Kinetis pin type connection <sup>1</sup>	Kinetis I/O	Description
SIO_C	I	Control signal. Indicates each transmitted bit.	SCL	I2Cn_SCL	O	Serial clock line of the I2C system.
SIO_D	I/O	Bidirectional data signal.	SDA	I2Cn_SDA	I/O	Bidirectional serial data line of the I2C system.

1. I2Cn – n represents the number of the I2C module.

## 2.3 Camera external clock

Kinetis supplies the camera with an external clock to create all timing synchronizations for signals participating in image data transmission. The generated clock rate is according to the camera specification.

**Table 2. Camera external clock signal**

CMOS sensor signal	CMOS sensor I/O	Description	Kinetis signal	Kinetis signal	Kinetis I/O	Description
XVCLK	I	System clock input	PWM	FTMn_CHx <sup>1</sup>	O	Output clock according to PWM definitions

1. FTMn\_CHx – n represents the number of FTM module and x represents the number of channel.

## 2.4 Camera image transfer control and data

The Kinetis family does not have a dedicated CSI (CMOS sensor interface) therefore the camera was connected through a standard GPIO.

This connection includes all the signals involved in the image transmission from the camera to Kinetis; the data signals, which hold the image data, and the control signals, which manage the way the data is transmitted.

## 2.4.1 Image data signals

The camera transmits image data signals over eight data lines in parallel. These lines are connected to eight sequential Kinetis GPIO pins configured as inputs.

## 2.4.2 Image control signals

The camera control signals are VSYNC, HREF and PCLK. The VSYNC signal determines when a new frame begins. The HREF signal represents the period of data transfer of a row in the transmitted frame. When the HREF signal is active, there is valid data over the data lines every pixel clock (PCLK) cycle. The PCLK signal indicates a valid data byte over the data lines and it is used as transfer trigger for the DMA module.

## 2.4.3 Camera image transfer control and data connections

The following table shows the camera signals used in control and data transmission, Kinetis pins allocated to perform the transmission and the action each signal initiates.

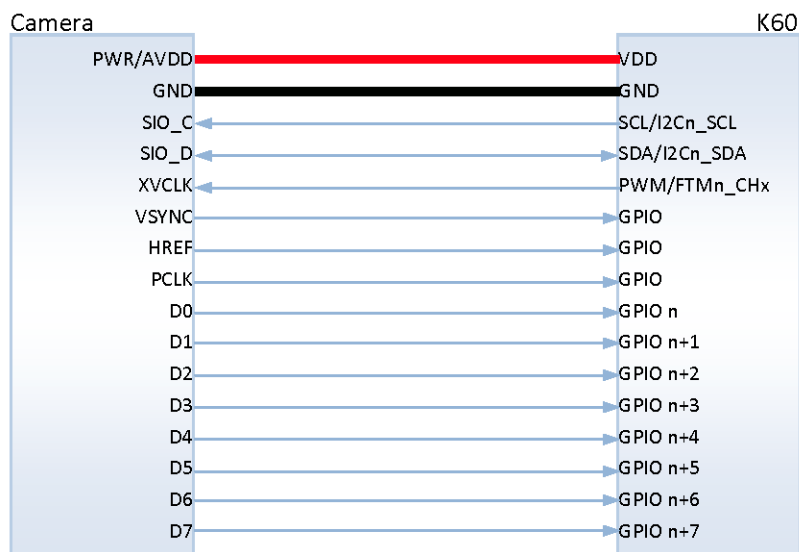
**Table 3. Image transfer control and data signals**

CMOS sensor Signal	CMOS sensor I/O	Description	Kinetis pin type connection	Kinetis I/O	Initiates
VSYNC	O	Vertical synchronization. Indicates the beginning of a frame.	GPIO pin	I	Interrupt routine
HREF	O	Horizontal synchronization. Indicates the period of data transfer of a row.	GPIO pin	I	Interrupt routine
PCLK	O	Pixel clock. Asserted when a byte is ready for transmission.	GPIO pin. DMA MUX source	I	DMA HW request
D0	O	Data line 0	GPIO pin n	I	
D1	O	Data line 1	GPIO pin n+1	I	
D2	O	Data line 2	GPIO pin n+2	I	
D3	O	Data line 3	GPIO pin n+3	I	
D4	O	Data line 4	GPIO pin n+4	I	
D5	O	Data line 5	GPIO pin n+5	I	
D6	O	Data line 6	GPIO pin n+6	I	
D7	O	Data line 7	GPIO pin n+7	I	

**NOTE**

It is significant for the eight GPIO port input pins to be sequential such that GPIO pin n is connected to data line [0] of the camera, and that GPIO pin n+7 is connected to data line [7].

## 2.5 Connections diagram



**Figure 1. Connections diagram**

## 3 Implementation Overview

### 3.1 Specification and features

Implementation is performed on MK60DN512VMD10 on TWR-K60D100M board and the camera used is OmniVision 1/9” CMOS VGA (640 x480) image sensor (OV7675).

The following configuration options are brought up as part of the specific camera used in this application.

Optional output formats can be RGB565 or YUV4:2:2 this depends on the conversion method and the display option in use. The destination of the image transfer is the internal SRAM of the K60 which is 128 KB and so the configured resolution for the camera is QQVGA (160 x 120). For the above output formats the camera transmits two bytes per pixel, therefore the frame buffer size in Kinetis is twice the number of pixels (38400 bytes per frame).

The input clock for the camera is provided by Kinetis. The K60 generates a 24 MHz output clock according to the typical value of the camera as specified in its Data Sheet. For more details about the camera please refer to the OmniVision website at [www.ovt.com](http://www.ovt.com).

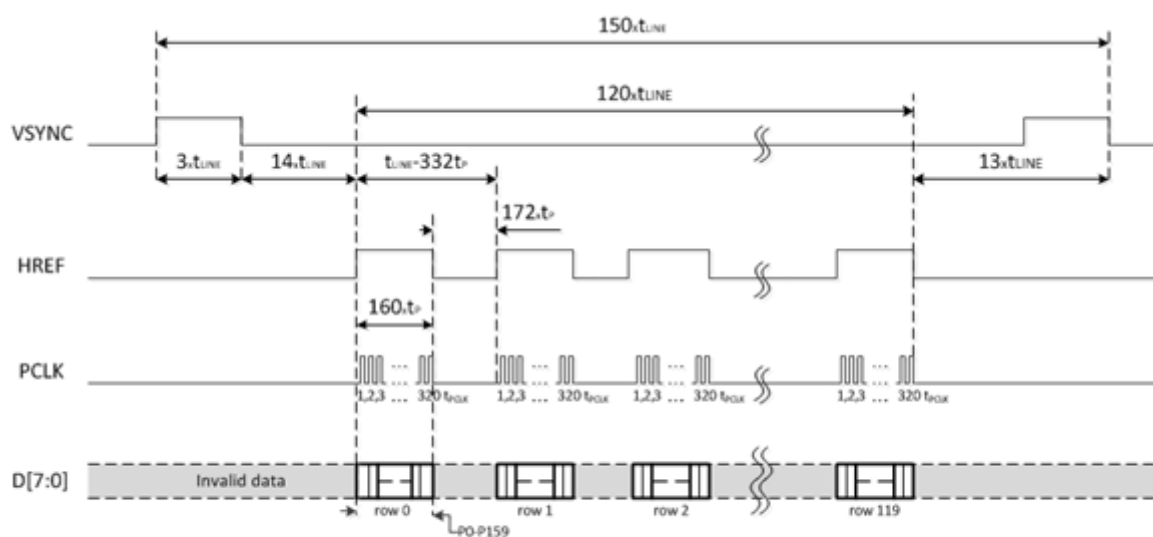
## 3.2 General description

Implementation of this application is divided into two major sections. The first is the camera configuration and the second is the image data transfer.

As mentioned before, the camera configuration is done through I2C communication protocol to configure the CMOS sensor to the desired work mode. The configuration is executed by writing values to the necessary registers. After the registers are configured, the camera produces image data accordingly. Data transmission to the Kinetis memory can now start. The camera configuration includes setting the PCLK signal to be active only during HREF active period. It also includes setting the frame rate and resolution.

The transmission procedure includes interrupts and DMA usage. A VSYNC signal generates an interrupt starting the transmission procedure. Following the VSYNC signal, the DMA hardware trigger is enabled. Each PCLK initiates a DMA transfer of eight data signals from the GPIO port pins (connected to the camera data lines) to Kinetis SRAM. The PCLK signal GPIO pin is the DMA source routed by the DMA Multiplexer. When HREF signal turns inactive, current row transmission is finished.

The following time diagram shows the signals during data transmission of a single QQVGA frame.



Note: For YUV/RGB,  $t_p = 2 \cdot t_{PCLK}$

**Figure 2. Time diagram of data transmission**

The following flow diagram displays the main stages of the process of acquiring a frame from the camera.

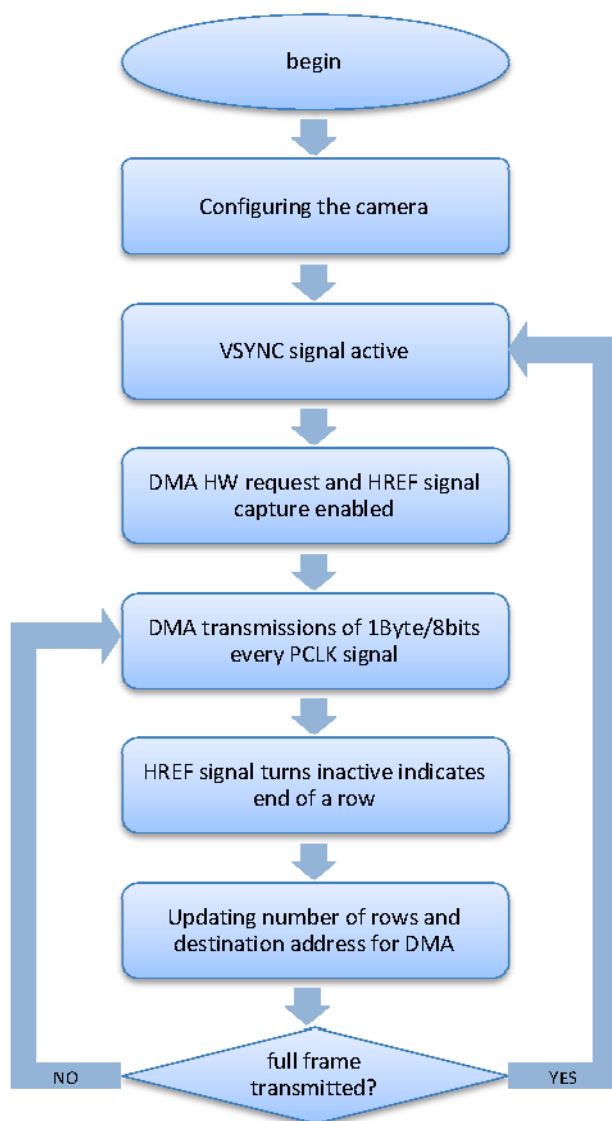


Figure 3. Flow diagram of implementation

### 3.3 Participating Kinetis modules

All the Kinetis K series share the same modules (IP blocks) that enable reuse of the code across the entire family. Selection of a different derivative of Kinetis K requires a new pin muxing to ensure all the necessary connections can be defined. The Kinetis modules that were used to implement this application are described below.

#### 3.3.1 Direct Memory Access Controller — eDMA

The data transfer is executed by the DMA to reduce CPU loading and allows it to work with other tasks at the same time (for example JPEG encoding). The PCLK signal indicates a new single byte that transmits over the data lines. It is used as a hardware request trigger for the DMA to transfer one byte per request.

Main topics to configure the DMA for hardware request triggered transfer include:

- Enabling the DMA module clock.
- Enabling the hardware request for the channel.
- The DMA Channel for transfer should be selected according to priority of the image data transfer over other DMA transfers.
- Destination address is the Kinetis internal SRAM.
- Source offset is set to 0 as the source address of the camera single byte data is static.
- Destination offset is 1 to increase the destination address after each DMA transfer by one byte.

A detailed DMA registers configuration is described in the code included in this application note.

#### NOTE

All eight GPIO port input pins of Kinetis must be sequential due to the fact they create the source address for the DMA.

### 3.3.2 Direct Memory Access Multiplexer — DMAMUX

The DMAMUX routes DMA sources (called slots) mapped to any of the DMA channels. Each DMA channel can be assigned to one of the possible peripheral DMA slots or to one of the always-on slots. There are three modes of operations for the DMAMUX; Disable, Normal, and Periodic trigger modes. Here, the DMA channel routes the specified source to the DMA channel in Normal mode.

The DMA source is the GPIO port that contains the PCLK signal input pin. This signal is the DMA hardware request pin. This GPIO port source number is listed along with all other DMA sources in the Chip Configuration chapter found in the Reference Manual, DMA Request Sources table.

Using the DMAMUX module in this configuration requires the following:

- Enabling the DMAMUX clock.
- Configuring the DMAMUX register according to the DMA channel in normal mode and to the GPIO port pin source.

### 3.3.3 Inter-Integrated Circuit — I2C

The I2C module provides a method of communication between several devices. In this case the interface allows communication between Kinetis (master) and the camera (slave). The implementation of the Kinetis-camera communication uses an I2C baud rate of ~400 kbit/s. Kinetis also supports other I2C baud rates.

There is a limitation on the camera I2C baud rate that must be taken into account when setting the Kinetis I2C baud rate. There is a relation between the input clock to the camera and the maximum possible I2C baud rate for the camera module in use. Please refer to the camera vendor for more details.

The I2C bus system uses the Serial Clock Line (SCL) and the Serial Data Line (SDA) for data transfers. Each camera has its own slave address. It can be found in the camera specification.

Using the I2C module in this configuration requires the following:

- Enabling the I2C clock.
- Setting the I2C baud rate.
- Enabling the I2C module.

I2C communication between Kinetis and the camera is composed of the following four parts:

- START signal.
- Slave address transmission - the I2C has a 7-bit address space followed by a R/W bit. The R/W bit defines the desired direction of data transfer for the slave.
- Data transfer.
- STOP/repeated START signal.

For more details of Kinetis I2C communication method please review the I2C chapter in the Reference Manual.

### 3.3.4 Port pins – GPIO

The GPIO is used for a DMA request trigger, interrupts source and data transfer with DMA to transfer image from camera without dedicated CSI port on Kinetis.

The camera control signals (VSYNC, HREF, PCLK) connected to Kinetis GPIO inputs are used to generate DMA request triggers that transfer image data from the camera without a dedicated CSI port on Kinetis. Users should define the pin muxing to avoid collisions between different modules and their required functionality.

All pins that are connected to the camera for control and data must be configured as input GPIO. The corresponding PORT module clocks need to be enabled. Each pin has edge sensitive (rising, falling, both) or level sensitive (low, high) support and can be configured according to signal behavior for interrupt and DMA request purposes. The VSYNC signal is configured for rising edge interrupt, HREF is configured for falling edge interrupt, and PCLK is configured for falling edge DMA request.

#### 3.3.4.1 Interrupt routines

Two interrupt routines are in use; one for the VSYNC signal and the other for the HREF signal. The interrupt routine for VSYNC signal enables a DMA request triggered by the PCLK signal and also enables an interrupt by the HREF signal. The camera can be configured such that PCLK signal does not toggle during horizontal blank. This implies that PCLK goes active only when HREF is active.

### 3.3.5 FlexTimer Module — FTM

The FTM is a timer that supports input capture, output compare, and generation of PWM signals. In this specific application the FTM is used as a PWM signal generator acting as an external clock to the camera. For more details of how to configure the FTM to work in PWM mode please refer to the FTM chapter in the Reference Manual.

## 3.4 Functional limitations

There are several limitations that must be considered during implementation. Solutions are suggested for some of these limitations.

### 3.4.1 Maximum DMA rate

The maximum transfer rate is limited for DMA hardware requests, especially for single byte transfers. Such a limitation requires a restricted pixel clock rate for the camera to avoid any data loss.

The maximum achievable rate for 100 MHz device is 8.7 MHz, according to the Performance section of the DMA in the Reference Manual. This limitation requires a careful configuration for the camera pixel clock rate and frame rate. As a result of this limitation this implementation uses 30 fps as frame rate and a 3 MHz PCLK rate, that is the DMA hardware trigger. Configuration of a 60 fps and 6 MHz PCLK was also tested successfully.



## 3.4.2 Memory

In this application note the internal SRAM memory was used to store the image data transferred from the camera. Kinetis K60 internal SRAM is limited to 128 KB, therefore the maximum resolution of the image is QQVGA (160 x 120) since there are 2 bytes per pixel. It is possible to use external memory interface of Kinetis such as DDR2 to increase the resolution of the image to QVGA (320 x 240) or other higher resolutions.

## 3.4.3 DMA major loop counter

The major loop counter of the DMA register `DMA_TCDn_CITER_ELINKNO` is limited to 15 bits, in the case where the channel linking feature of the DMA is not in use (refer to DMA chapter in the Reference Manual). Therefore, there is a limitation of 32K major loops equal to 32K bytes transferred. To overcome this limitation of transferring a full frame of QQVGA resolution and above, you must update the pointer of the DMA destination address after each row transferred. It is implemented by interrupt from the falling edge of the HREF signal.

## 3.4.4 Camera XVCLK input

To reduce the overshoot and undershoot of the XVCLK signal, serial resistor of 330Ω was used.

# 4 Conclusions and further options

This application note provides the basic concepts for implementing a glueless connection of CMOS sensors to the Kinetis MCU. In this application note the solution offered is implemented on the K60. For a cost-effective solution, it is possible to use the entry level of the Kinetis K series - the K10 and others. This is possible to implement because the Kinetis K series is scalable and uses the same IP blocks across the sub-families.

Whereas the application for K60 is implemented using the internal SRAM (128 KB) to store the image data, there are further options to use external memory. A use of external memory allows working with a higher resolution than the QQVGA implemented in this application note. Such an option is DDR2 or external SRAM. To work with external SRAM, use the Kinetis FlexBus interface. In case of DDR2 usage, you must choose a Kinetis device that supports a DDR controller.

An advanced option for this application suggests a JPEG compression. A JPEG compression algorithm can easily be adapted on the Kinetis MCU due to the DSP capabilities of the ARM Cortex-M4 core.

To process the image data on a different endpoint it is possible to transfer the data using different communication features the Kinetis MCU offers. There are either wired or wireless options possible to use:

- Ethernet port 10/100 that exists on the derivatives of the K60 and higher.
- WiFi - Freescale works with 3rd parties that offer a variety of Tower wireless peripheral boards with WiFi connectivity including 802.11n. For more information go to [www.freescale.com/tower](http://www.freescale.com/tower).
- ZigBee - Freescale offers a variety of solutions of sub GHz and 2.4 GHz Tower System Radio Peripheral modules. For more information go to [www.freescale.com/tower](http://www.freescale.com/tower).

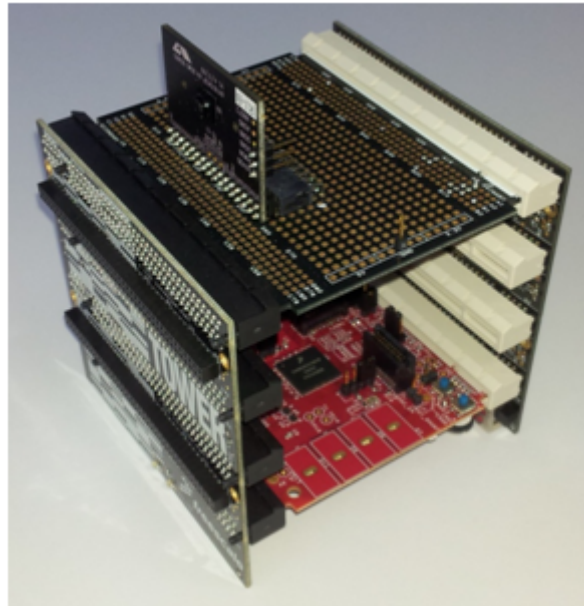
## Appendix A Hardware Connections

The following table shows the physical connections between the K60 and the camera as they are implemented. It specifies the names of the signal involved, pin name on the camera connector, pin name on the TOWER elevator, pin name on the CPU module TWR-K60D100M, and the muxing option of the pin. Refer to the schematics of the TWR-K60D100M and TWR-ELEV. Note that physical connection can be performed using the TWR-PROTO board.

**Table A-1. Kinetis-CMOS sensor connections**

Signal	CMOS sensor Connector - J2	TWR-ELEV primary connector	TWR-K60D100M – Primary J7A/TWR-PROTO Primary J2	Muxing Option
VDD	21 — PWR	J9 3	B3	—
GND	32 — GND	J8 — GND	A65 — GND_15	—
Camera configuration clock (I2C clock)	14 — SIOC	J8 7 — SCL0	A7 — SCL0	ALT2 — I2C0_SCL
Camera configuration data line (I2C data)	12 — SIOD	J8 8 — SDA0	A8 — SDA0	ALT2 — I2C0_SDA
Camera input clock <sup>1</sup>	20 — XCLK1	J8 40 — PWM0	A40 — PWM0	ALT4 — FTM0_CH0
VSYNC	15 — VSYNC	J8 22 — SSI_BCLK	A22 — SSI_BCLK	ALT1 — PTE12
HREF	13 — HREF	J9 80 — FB_D0	B80 — EBI_D0	ALT1 — PTC15
PCLK	17 — PCLK	J9 66 — FB_AD15	B66 — EBI_AD15	ALT1 — PTB18
D0 -Data line 0	2 — D2	J9 22 — GPIO2/OPEN	B22 — GPIO2/ SDHC_D1	ALT1 — PTE0
D1- Data line 1	1 — D3	J9 10 — SPI1_MOSI	B11 — SDHC_D0/ SPI1_MISO	ALT1 — PTE1
D2- Data line 2	4 — D4	J9 7 — SPI1_CLK	B7 — SDHC_CLK/ SPI1_CLK	ALT1 — PTE2
D3- Data line 3	3 — D5	J9 11 — SPI1_MISO	B10 — SDHC_CMD/ SPI1_MOSI	ALT1 — PTE3
D4 - Data line 4	6 — D6	J9 9 — SPI1_CS0	B9 — SDHC_D3/ SPI1_CS0	ALT1 — PTE4
D5 - Data line 5	5 — D7	J8 10 —GPIO8/OPEN	A10 — GPIO8/ SDHC_D2	ALT1 — PTE5
D6 - Data line 6	8 — D8	J8 21 — SSI_MCLK	A21 — SSI_MCLK	ALT1 — PTE6
D7 -Data line 7	7 — D9	J8 24 — SSI_RXD	A24 — SSI_RXD	ALT1 — PTE7

1. A 330Ω resistor is connected in series between the J2 connector pin 20 and the J8 connector pin 40 adjacent to J2



**Figure A-1. Kinetis-CMOS Sensor Hardware**

## Appendix B Software

An example software project targeted at the Kinetis K60 MCU has been created to demonstrate the transfer of the camera image data to the internal SRAM. The Freescale Tower system based on the TWR-K60D100M and the TWR-PROTO boards have been used for demonstration software development and testing.

The software project runs from the MCU flash memory and does not run under an operating system. The code footprint is approximately 7 KB. The example sample project was implemented by using IAR tools version 6.40.2. The frame buffer that is transferred to the internal SRAM can be saved and reviewed for debugging and analysis purposes on a PC by IAR IDE.

The code is comprised of the following basic stages:

- Initialization of the participating K60 modules and the required interrupts. This part includes the init functions and the clock enable phase.
- Camera setting by I2C — This is where the OV7675 is configured to the preferable mode of work to produce the QQVGA image resolution in YUV422 format.
- The periodic loop of image data transfer managed by the interrupts that are triggered by the control signals of the camera.



**How to Reach Us:**

**Home Page:**

[freescale.com](http://freescale.com)

**Web Support:**

[freescale.com/support](http://freescale.com/support)

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [freescale.com/SalesTermsandConditions](http://freescale.com/SalesTermsandConditions).

Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, ColdFire+, C-Ware, Energy Efficient Solutions logo, Kinetis, mobileGT, PowerQUICC, Processor Expert, QorIQ, Qorivva, StarCore, Symphony, and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Airfast, BeeKit, BeeStack, CoreNet, Flexis, Layerscape, MagniV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SafeAssure, SafeAssure logo, SMARTMOS, Tower, TurboLink, Vybrid, and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Original Copyright–Latest Copyright Freescale Semiconductor, Inc.

Document Number AN4627  
Rev 0, 5/2013  
29 May 2013

