Freescale Semiconductor
Application Note

Document Number:AN4762
Rev 0, 10/2013

Phase Voltage Reconstruction Based on the Qorivva Devices

by: Petr Konvicny

Contents

1 Introduction

The Pulse-Width Modulation (PWM) is a modulation technique that conforms the width of the pulse based on modulator signal. It allows the control of the power, supplied to electrical devices, such as motors. The average value of the voltage (current) fed to the load is controlled by turning the switch between supply and load on and off. In the motor control area, are many reasons why it is useful to know the value of the real phase voltages, which are applied by the inverter to a load. Based on these values, the motor control application can then reconstruct the real inverter phase voltages and use them in the advanced motor control algorithms such as observers, dead-time compensation, etc.. This document describes the Freescale Qorivva family features/approaches that can be efficiently used for motor control algorithms.

2 Reading real PWM duty-cycles

Freescale's Qorriva devices offer two possible ways to collect the real PWM duty-cycle applied to a load/motor. One possible solution is utilization of the PWMX inputs from flexPWM module, and the other is to use the eTIMER module.

3 External hardware requirements

To be able to read a real duty-cycle, the converter has to provide the real phase voltages as feedback signals connected to MCU inputs. Naturally, they cannot be linked directly (which can cause MCU damage) but have to be conditioned to device input levels. Figure 1 shows the simplified circuit schematic. A signal conditioning circuit (an external hardware circuit) can be implemented, for example, as a voltage divider, or for a low-voltage application the customer can put to good use the Freescale MOSFET pre-driver (MC33937A) features as shown in Figure 2. The pre-driver internal phase comparators allow the user to read through the PHASEx output's digital image of the real phase voltages compared to the half of the DC-bus voltage. The MCU can process these feedback signals, which reflect the real duty-cycle applied by the voltage converter, for real load phase voltage reconstruction. The precision of the signal conditioning can have a significant impact, for example, on the dead-time compensation function results or precision, especially at high DC link voltages.

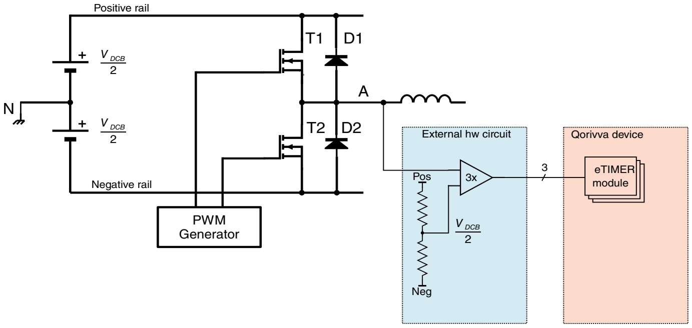

Figure 1. Real phase duty-cycle reading diagram

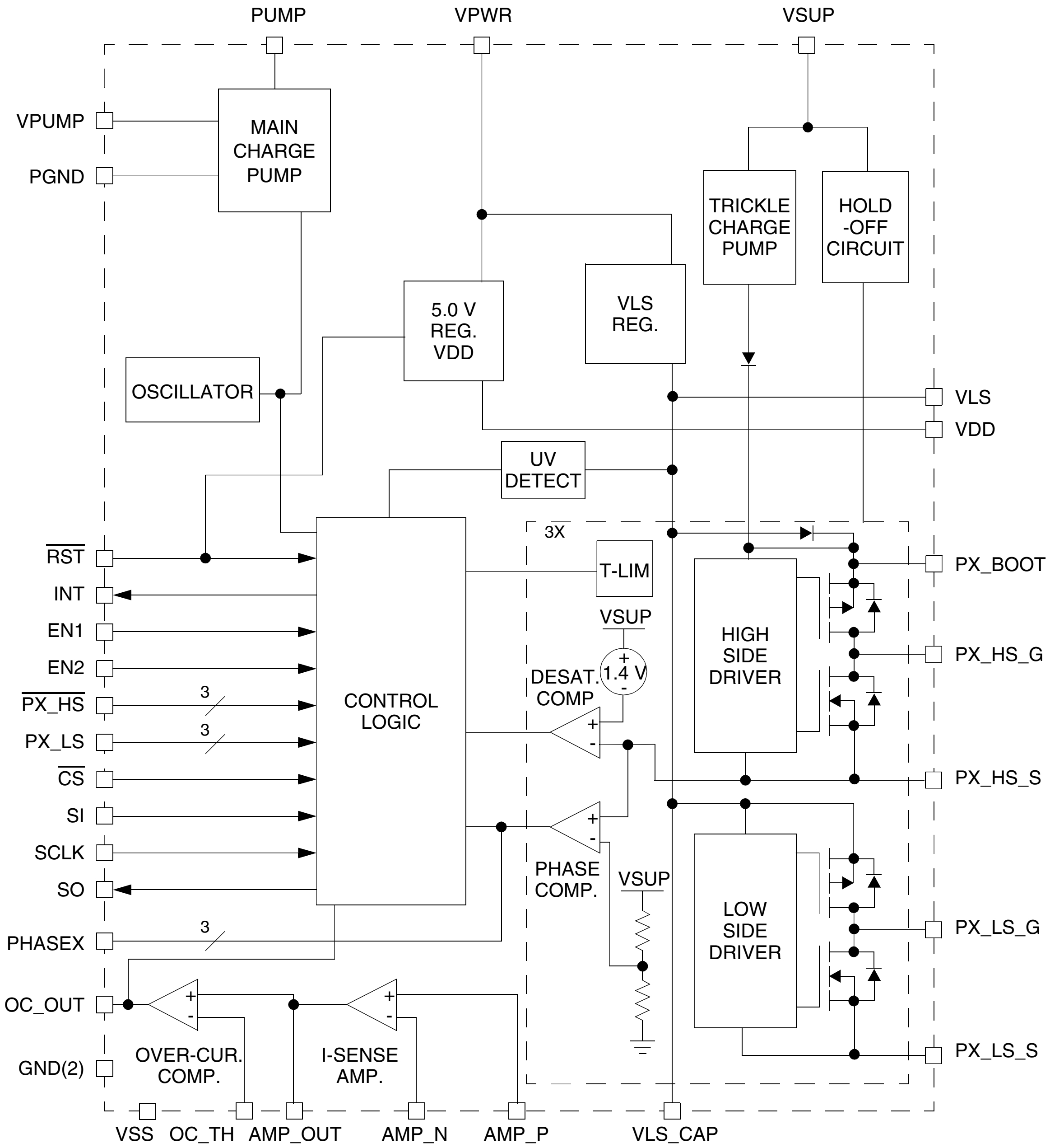

Figure 2. MC33937A internal block diagram

4 PWMX inputs method

Each FlexPWM module contains four submodules with two PWM outputs (PWMA and PWMB) and one input (PWMX). A PWMX inputs enables sensing a duty-cycle of the real phase voltage (it is proportional to the applied duty-cycle). If the associated pads (related PCR registers) are configured as PWMX inputs in the SIUL module, then the PWMX inputs are directly connected to the FlexPWM submodule capture circuits. Each submodule has one capture block with two capture circuitries 0 and 1 for capturing the time (submodule timer) of the rising, falling or both edges, as is shown in Figure 3. These circuits capture the submodule internal timer values of the PWMX signal at the rising and falling edges. These values can be used, for example, for calculation of a real voltage applied to a load.

Each FlexPWM submodule contains one capture control X register (CAPTCTRLX) dedicated for setting the capture circuit properties. Figure 4 shows a capture control X register. Bits EDGX1, EDGX0, ONESHOTX are set in order to correctly capture the time of the PWMX rising and falling edges. The ONESHOTX bit is set to one. The one shot mode is selected. The capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed and capture circuit 1 is armed. After capture circuit 1 captures a value, it is disarmed and the ARMX bit is cleared. No further captures will be performed until the ARMX bit is set again.

The EDGX0 field is set to 01.

This field determines which input edge triggers a capture event. The value 01 captures the falling edges.

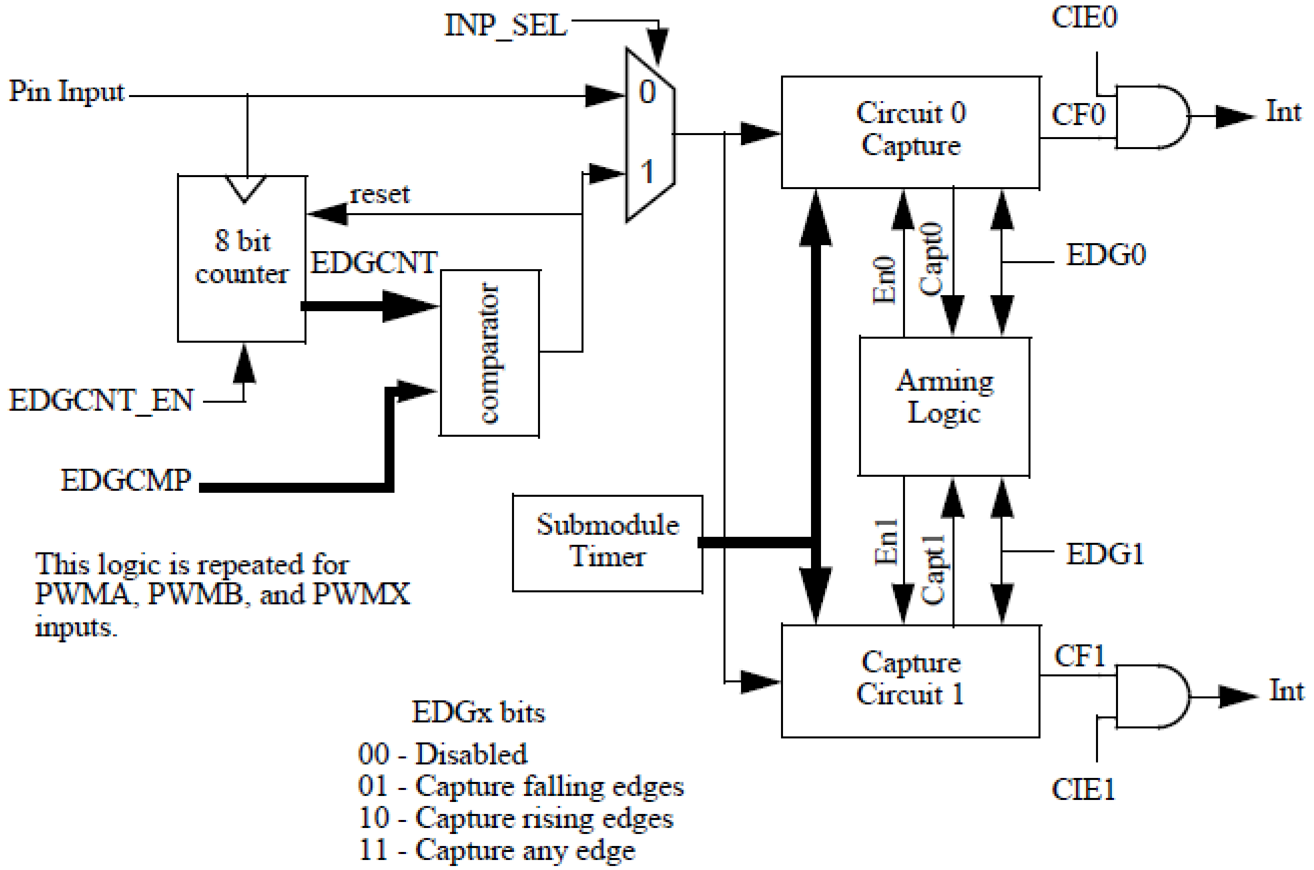

Figure 3. Capture logic diagram

The EDGX1 bit field is set to 10.

These bits determine which input edge triggers a capture event. The value 10 capture rising edges.

Address: Base + 0x0030 (Submodule 0)
Base + 0x0080 (Submodule 1)
Base + 0x00D0 (Submodule 2)
Base + 0x0120 (Submodule 3)

Access: User read/write

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	CX1CNT			CX0CNT			CFXWM		EDG CNTX _EN	INP_ SELX	EDGX1		EDGX0		ONE SHOT X	ARM X
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4. Capture Control X register

The results of the respectively captured values of the rising and falling edges are stored in the CVAL0 and CVAL1 registers and could be used for calculation of a real phase duty-cycle respective voltage applied to a load.

5 eTIMER method

The second option for sensing a real PWM duty-cycle is to connect the conditioned feedback phase signals to the input pin of eTIMER channels, as shown in Figure 6. All motor control dedicated modules (flexPWM, CTU, eTIMER) work in the same clock domain. Each of the six channel counters consists of two independent inputs/sources (primary, secondary). Each channel has two basic modes of operation: it can count internal or external events, or it can count an internal clock source while an external input signal is asserted, thus timing the width of the external input signal. This second mode of operation can be an advantage when used to read the real phase duty-cycle. The counter mode is selected by the CNTMODE field in the CTRL1 register. To read the feedback signal width, the CNTMODE field value has to be set to 3 (count the rising edges of the primary source while the secondary input is high active). From the CNTR or HOLD registers, the user can read a real duty-cycle value.

6 Signal interconnection

The Freescale Qorriva based motor control controllers and the 3-phase low-voltage power stage boards offer the possibility of implementing both the above described methods. The schematic of the boards can be found on the Freescale web page. The real phase voltages applied to the 3-phase load (motor) are measured by using MC33937A internal phase comparators, see Figure 2. The outputs from these inner phase comparators (MC33937A pins PHASEX) are connected to UNI-3 interface connector (J3) as signals ZCx (A, B, C, pins 34, 35, 36) on the 3-phase power stage board side.

7 FlexPWM approach

On the controller board side, the mentioned signals are connected to the submodules PMWX0, PMWX1, and PMWX2 inputs as shown in Figure 5. These inputs are directly connected to the FlexPWM internal capture circuits. The captured values correspond to the times of the rising and falling edges of the phase voltage. From these values the real phase duty-cycle (voltage) applied to the load could be calculated.

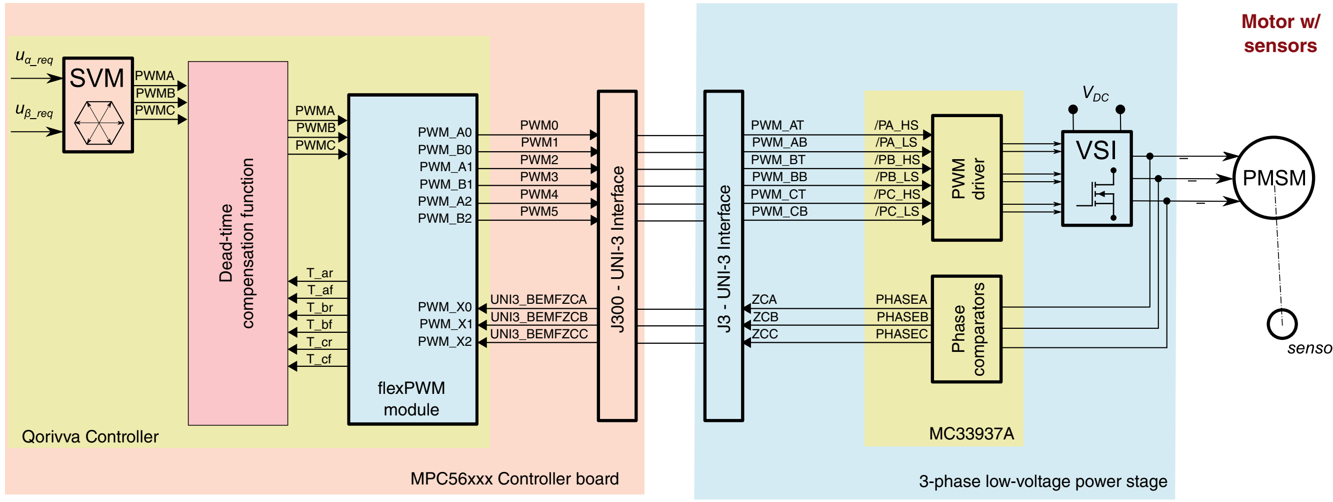

Figure 5. FlexPWM-based real duty-cycle sensing block diagram

8 eTIMER approach

On the controller board side, the mentioned signals (ZCA, B, C, pins 34, 35, 36) are connected to the eTIMER inputs called PHASEA0, PHASEB0, and INDEX0 as shown in Figure 6. These inputs are then directly connected to the eTIMER channels 0, 1, and 2. The module channels are then configured to read the time of a positive signal value, which corresponds to the inverter real phase x duty-cycle. These values can be read by the dead-time compensation function to compensate for it.

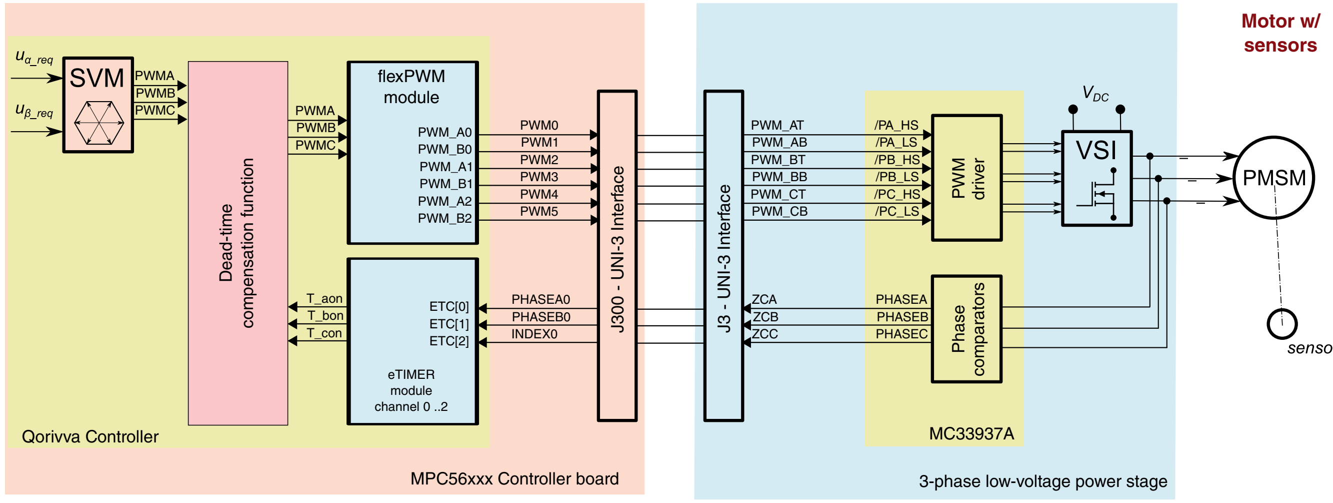

Figure 6. eTIMER-based real duty-cycle sensing block diagram

9 Comparison of methods

As is shown in Figure 7, the real system introduces some delays on the signal path which can lead in particular situations to a wrong reading of the converter phase voltage feedback signal. Let's suppose that the situation in Figure 7 is evaluated by the FlexPWM module, as is described in FlexPWM approach. The falling edge of the feedback signal comes after the MRS signal (Master Reload Signal - start of a new PWM period). This can be caused by the system delays. The FlexPWM capture unit triggers only one capture event (rising edge). However, the second event (falling edge) is not recognized in this period, and the system then reads the wrong value. This can happen when the control calculation period equals the PWM period. For systems with a calculation period of n-times (two and more) higher than the PWM period, this drawback will not occur.

When the eTIMER approach is employed this situation will never arise. The eTIMER count module system clocks when a secondary input is high without any impact on the MRS signal. This method also provides a real duty-cycle with no additional calculations. On the other hand, it requires more on-chip resources (an additional three eTIMER channels), but the user can read the real pulse width without any CPU load.

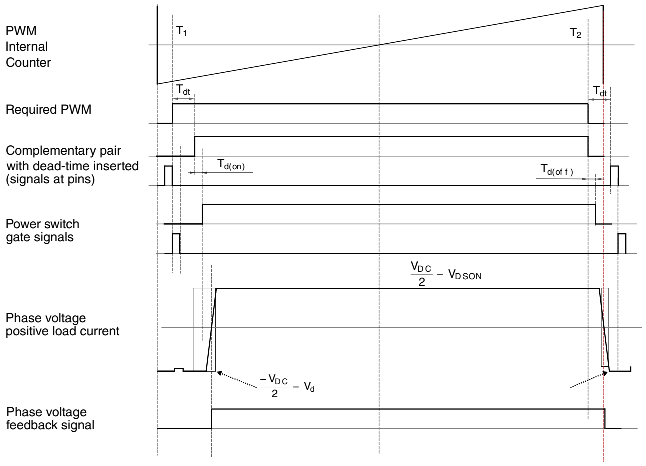

Figure 7. PWM signals delay

10 Phase voltage calculation

The measured real duty-cycle value can be used to calculate the voltage applied per one PWM period to a load. Using a volt-second balance approach, the average phase voltage can be calculated. A volt-second balance (average voltage) per one PWM period can be derived from equation 1.

$$\overline{V_{AN}} = \frac{1}{T_S}\int_{0}^{T_S} v_{an}(t)\mathrm{d}(\mathrm{t}) \qquad (1)$$

This equation does not take into account the voltage drop on the switches, and this might introduce an inaccuracy in the magnitude of the phase voltage. The phase voltage is proportional to the duty-cycle and DC-bus voltage, as shown in equation 2.

$$\begin{aligned}\overline{V_{AN}} &= \frac{1}{T_S}\left(\frac{V_{DC}}{2}t_{ON} - \frac{V_{DC}}{2}t_{OFF}\right) = \frac{V_{DC}}{2}(2 * DC - 1) \\ \frac{T_{ON}}{T_S} &= DC \cdots \text{measured real phase duty-cycle}\end{aligned} \qquad (2)$$

11 Conclusion

Both approaches can be useful when the user wants to investigate the real PWM duty-cycles introduced by a VSI converter to a load. Neither data collection method overloads the CPU.

The Freescale Qorriva family members dedicated to the motor control area usually contain four key internal modules which may be advantageously used for a deterministic analog data collection of the system variables, rotor angle and speed sensing, and the generation of the PWM signals for the electrical motor. Figure 8, Figure 9, Figure 10, and Figure 11 show the motor control modules interconnection block diagrams for a selected family members. A flexPWM module allows the user to generate up to eight PWM signals, and usually for the motor control we use six of them to drive the MOSFET/IGBT converter circuit. A CTU (Cross Triggering Unit) provides synchronization and event generation independently from the CPU. This can be synchronized with the start of the PWM period (MRS event) and can carry out up to eight trigger events. One eTIMER module consists of six independent timers with capture, which are able to count the internal clock or external input edges. The ADC module converts analog quantities such as phase current, DC-bus voltage, etc. The CTU module can directly control ADC module without any CPU load. This feature allow users to collect any analog quantities without any CPU load. Usually, after the end of a conversion, an interrupt arises where the motor control algorithm is calculated.

12 Appendix

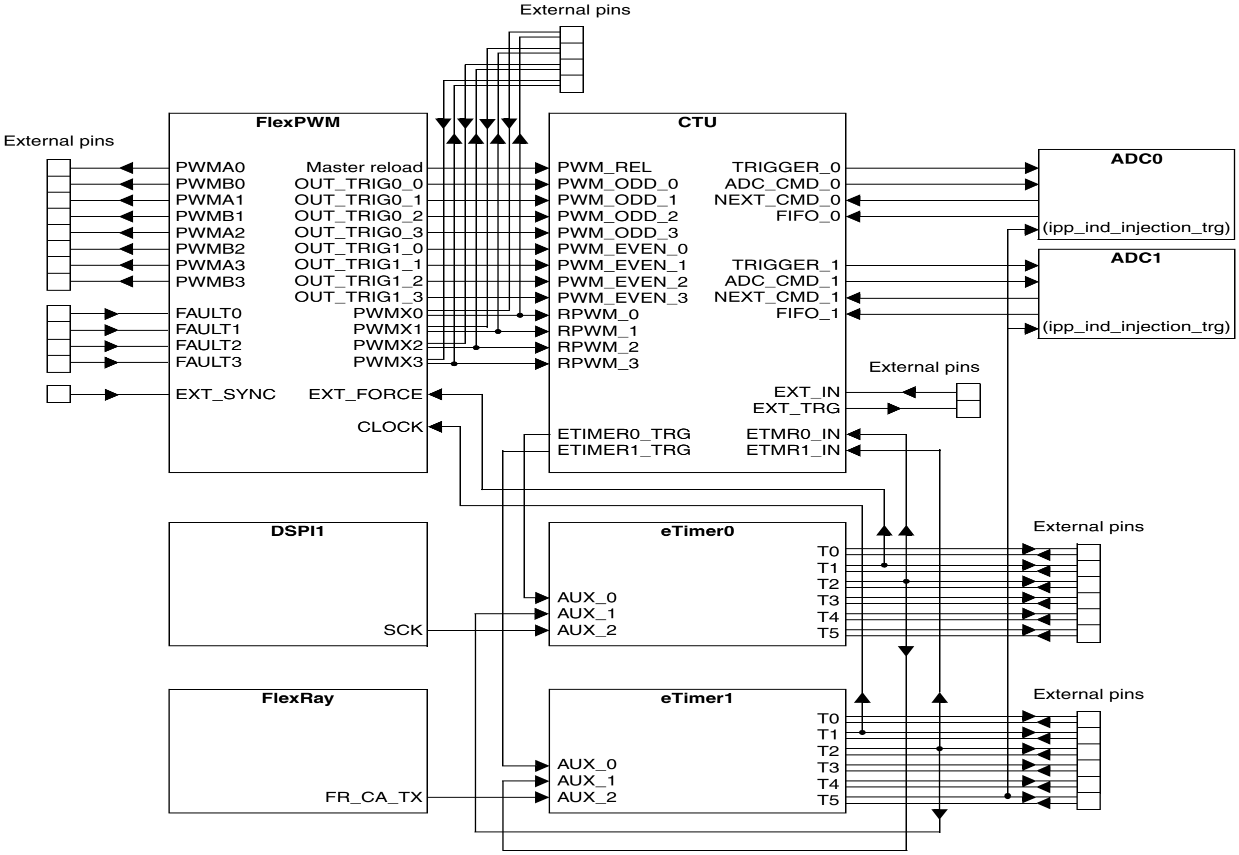

Figure 8. MPC5604P motor control peripherals interconnection diagram

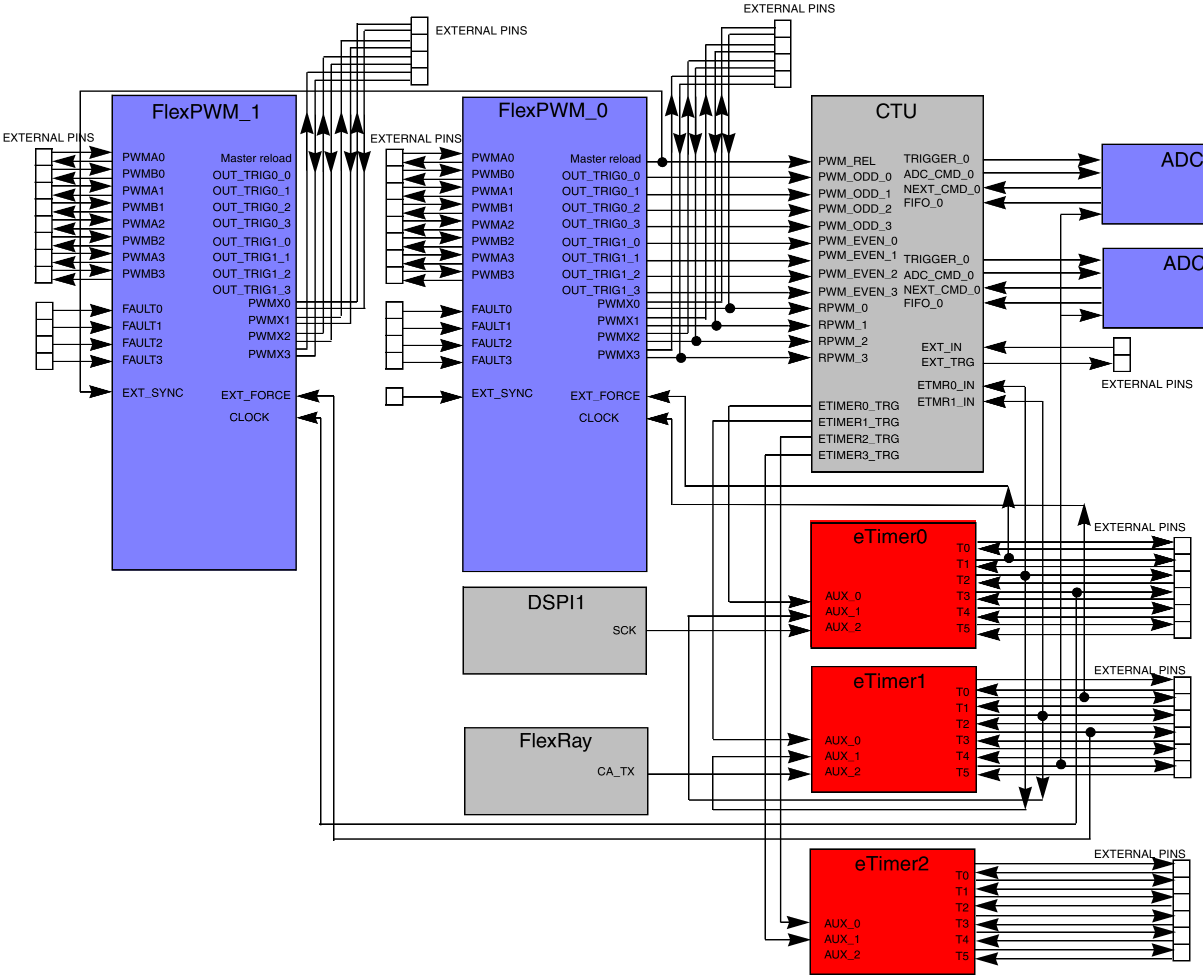

Figure 9. MPC5643L motor control peripherals interconnection diagram

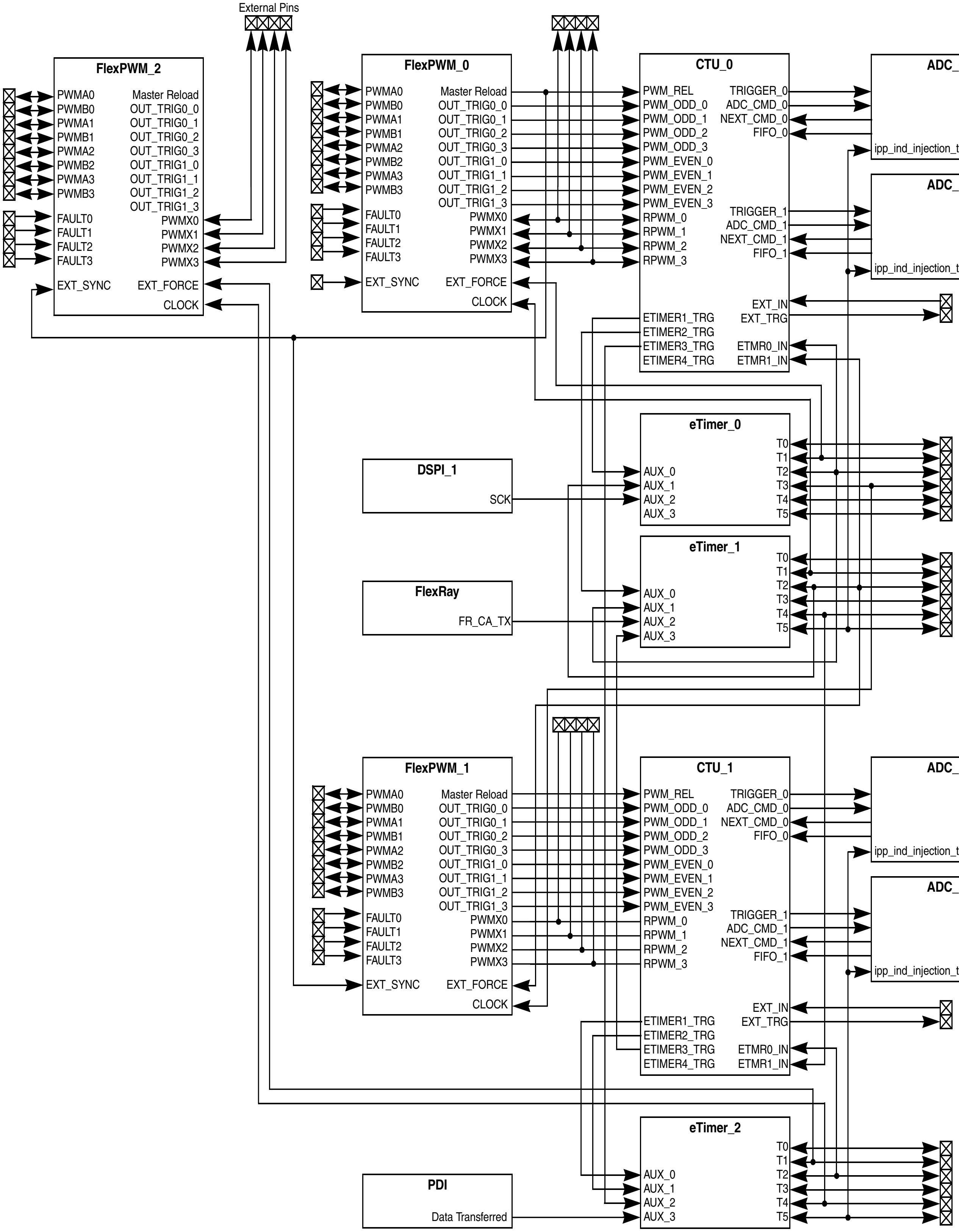

Figure 10. MPC5675K motor control peripherals interconnection diagram

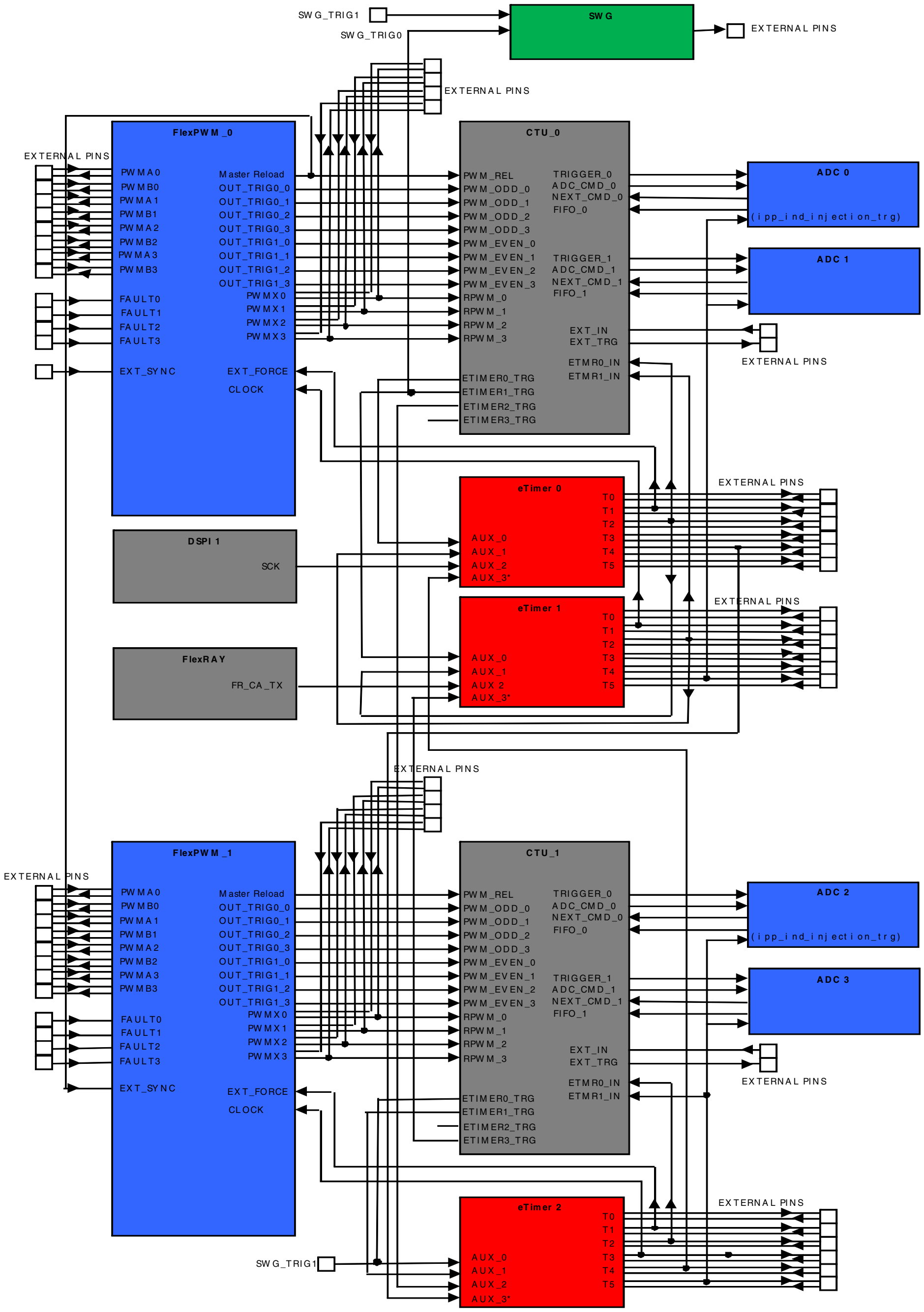

Figure 11. MPC5744P motor control peripherals interconnection diagram

How to Reach Us:

Home Page:
freescale.com

Web Support:
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.