

Quadrature Decoder Implementation on MCF51AC256

How to Read Encoder Position and Speed on a Platform With No Dedicated Peripheral

by: Jaroslav Musil
Czech Republic

1 Introduction

This application note describes the evaluation of speed and position on the MCF51AC256 platform using a conventional quadrature encoder position sensor. MCF51AC256 does not have any hardware support to decode the quadrature signals, so it is necessary to somehow reuse the timer inputs with the help of software.

This application note discusses the common difficulties faced when implementing a software quadrature decoder.

2 Quadrature Encoder

The quadrature encoder is a sensor that generates two signals (phase A, phase B) of rectangular pulses if the rotor changes its position. The signals are logical, similar to the clock signal. They are offset from each other by 90 degrees. If we consider that one period of such a signal is

Contents

1	Introduction	1
2	Quadrature Encoder	1
3	Connection	2
4	Implementation	2
4.1	Low Speed Mode	2
4.2	High Speed Mode	6
4.3	Transition between the Modes	8
4.4	TPM Configuration	9

Connection

360 degrees long, then 180 degrees is the period of the on- or off-pulse. Therefore a position of 90 degrees is in the middle of the on- or off-pulse.

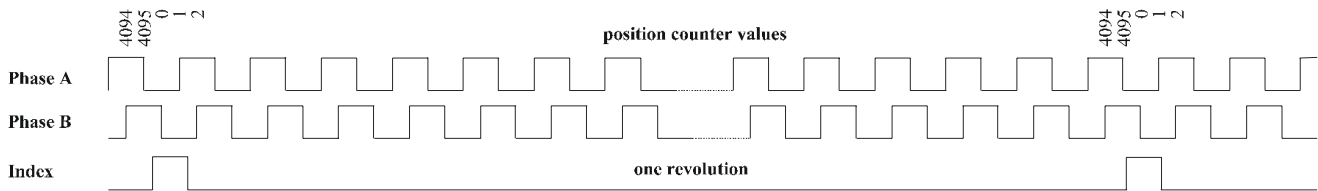


Figure 1. Quadrature Encoder Signals

The quadrature encoder has a certain number of pulses per one mechanical revolution. Taking into consideration that the signal has one phase edge per one pulse, two phases of such signals must generate four edges per one pulse. It means that the encoder has a capability of position recognition with a precision that is four times higher than the number of pulses in the specification.

3 Connection

MCF51AC256 does not offer any hardware peripheral for quadrature encoder signal processing. Therefore it is necessary to connect the quadrature encoder's signals to a timer peripheral.

A timer that can be used for the quadrature encoder must have the capability of running in the input capture mode, must have two free inputs, and must offer reading of the input pin status. The clock frequency of this timer must be high enough to enable measurement of the time between the capture events — in this particular case, between the encoder signals' edges that can be generated with a really high frequency. The timer must also have the possibility of setting the timer for a free running mode with an external clock.

A good timer that fulfills the above-mentioned conditions to be used with MCF51AC256 is the TPM3. This timer has two input channels: TPM3CH0 and TPM3CH1. Phase A of the quadrature encoder will be connected to the pin TPM3CH0 and phase B to pin TPM3CH1. TPM3 has also another pin that serves as an external clock input for its counter. The pin is named TPMCLK and the quadrature encoder's phase A signal will also be connected to this pin.

4 Implementation

The principle of position recognition using the quadrature encoder requires two modes:

- Low speed mode
- High speed mode

While the former is used when the speed of rotation is low (around zero), the latter is used for high rotor speeds. Description of these modes is found in the following sections.

4.1 Low Speed Mode

In low speed mode the TPM3 module is set up to use the bus clock (20 MHz) as the source clock for the counter. Channel 0 and channel 1 are configured for the input capture mode of both edges. In other words on any arriving edge the timer saves the time of the arriving edge and generates an interrupt.

In this interrupt the state of the channel 0 and the channel 1 pins is read. This state is compared to the previous state and the rotation direction is detected this way. Then the position variable is incremented or decremented by one with respect to the rotation direction.

To evaluate the speed the time of the previous phase A edge (which is stored in a variable from the previous step) is subtracted from the actual phase A's edge time. This period says how many bus clock periods were measured between two phase A's edges.

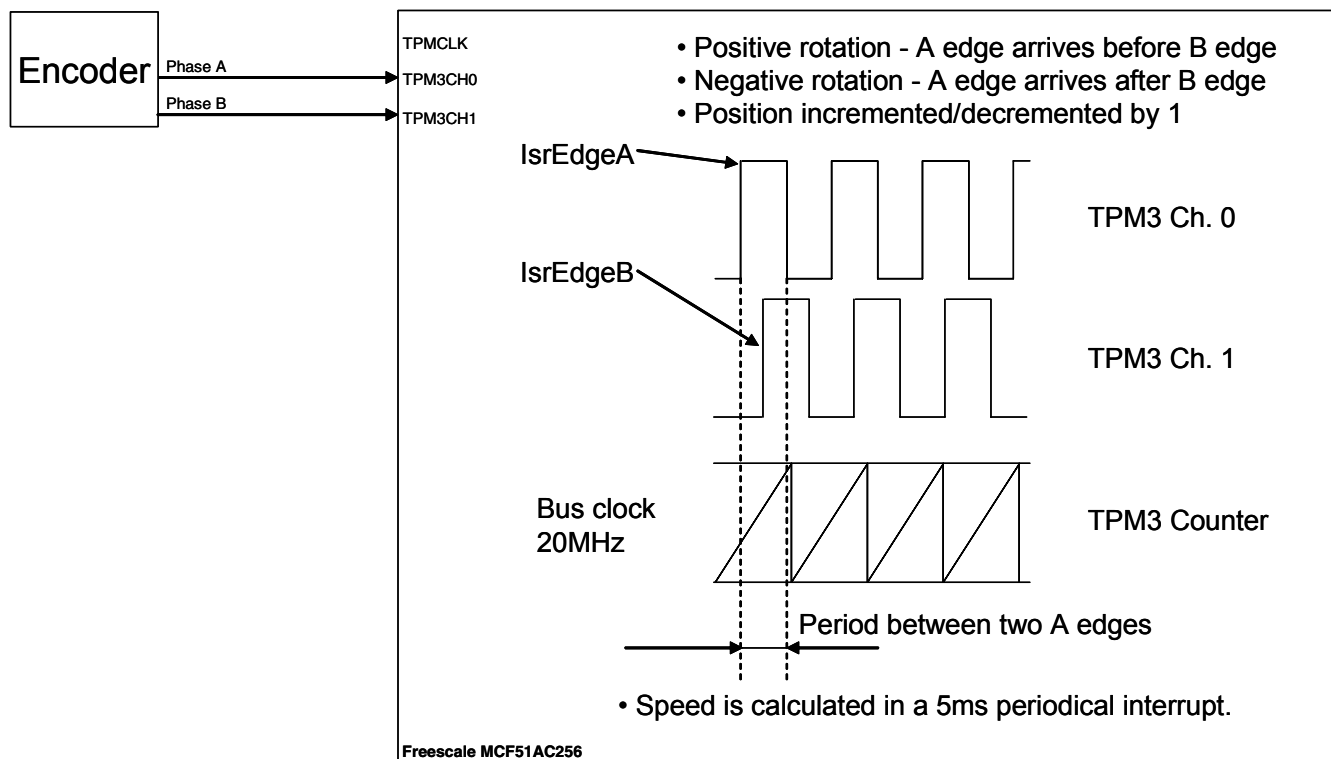


Figure 2. Low Speed Mode

The zero speed detection is made via TPM3's overflow interrupt. An overflow counter variable is zeroed in the TPM3 channel 0 capture interrupt. When the TPM3's counter overflows and its interrupt is generated, an overflow counter variable is increased. If this variable is greater than two, in other words the max 16-bit range was reached, the variable that carries the information of the time between two edges is set to its maximum, 65535.

The speed itself is calculated by the means of time between two edges in a periodical timer interrupt of FTM2. This interrupt is generated with a period of 5 ms, so the speed variable is updated once per 5 ms. If the time between two edges is 65535, the speed is set to 0.

Knowing the number of encoder pulses per one mechanical revolution and knowing the bus clock frequency, we can state:

$$t_{rev} = t_{bte} \times N_{ph} \quad \text{Eqn. 1}$$

where:

t_{rev} — time of one revolution

Implementation

t_{bte} — time between two edges

N_{ph} — number of encoder edges in one phase

$$N_{ph} = N \times 2 \quad \text{Eqn. 2}$$

where:

N — number of encoder pulses

$$t_{bte} = \frac{t}{f_{cnt}} \quad \text{Eqn. 3}$$

where:

t — measured timer clocks between two edges

f_{cnt} — frequency of the counter source (20 MHz)

$$f_{rotor} = \frac{1}{t_{rev}} \quad \text{Eqn. 4}$$

where:

f_{rotor} — rotor frequency (revolutions per 1 s)

For the mechanical speed in RPM we can state:

$$\omega_{RPM} = 60 \times f_{rotor} \quad \text{Eqn. 5}$$

where:

ω_{RPM} — speed in revolutions per 1 minute

Putting [Equation 1](#), [Equation 2](#), [Equation 3](#), and [Equation 4](#) into [Equation 5](#) we obtain the following equation:

$$\omega_{RPM} = 60 \times \frac{f_{cnt}}{2 \times N \times t} \quad \text{Eqn. 6}$$

To simplify this statement we can write the following equations:

$$\omega_{RPM} = \frac{K_1}{t} \quad \text{Eqn. 7}$$

$$K_1 = 30 \times \frac{f_{cnt}}{N} \quad \text{Eqn. 8}$$

where:

K_1 — speed constant

The minimum measurable speed can be calculated as follows:

$$\omega_{RPMmin} = 30 \times \frac{f_{cnt}}{N \times t_{max}} \quad \text{Eqn. 9}$$

where:

t_{max} — the maximum period that can be measured (limitation of the 16-bit register)

If the encoder has 1024 pulses per mechanical revolution the counter source is the bus clock of 20 MHz:

$$\omega_{RPMmin} = 30 \times \frac{20 \times 10^6}{1024 \times 65535} = 8.94 RPM \quad \text{Eqn. 10}$$

As mentioned above, this mode requires an interrupt to be generated on each arriving edge. If we consider the encoder of 1024 pulses per one mechanical revolution, in other words 4096 interrupts per one mechanical revolution, we can calculate the maximum speed for the lowest acceptable period between two input capture interrupts:

$$t_{revmin} = t_{int} \times 4 \times N \quad \text{Eqn. 11}$$

$$\omega_{RPMmax} = \frac{60}{t_{revmin}} = \frac{60}{4 \times N \times t_{int}} \quad \text{Eqn. 12}$$

where:

t_{revmin} — minimum time of one mechanical revolution

ω_{RPMmax} — maximum possible speed

t_{int} — minimum acceptable time between two interrupts

Therefore if we accept the minimum time between two interrupts as 100 μs and the encoder has 1024 pulses, the maximum possible speed for this mode will be:

$$\omega_{RPMmax} = \frac{60}{4 \times 1024 \times 100 \times 10^{-6}} = 146.5 RPM \quad \text{Eqn. 13}$$

If the application should run over 146 RPM it is necessary to use another mode that will not occupy the processor so much.

4.2 High Speed Mode

High speed mode was created because of high processor usage in low speed mode. In this mode channel 0 and channel 1 of the timer are set up for free running. In this mode no interrupt is generated when an edge arrives. The counter clock source is set up for the external clock source which is the phase A signal in this case. So the counter will count the pulses generated by the encoder's phase A.

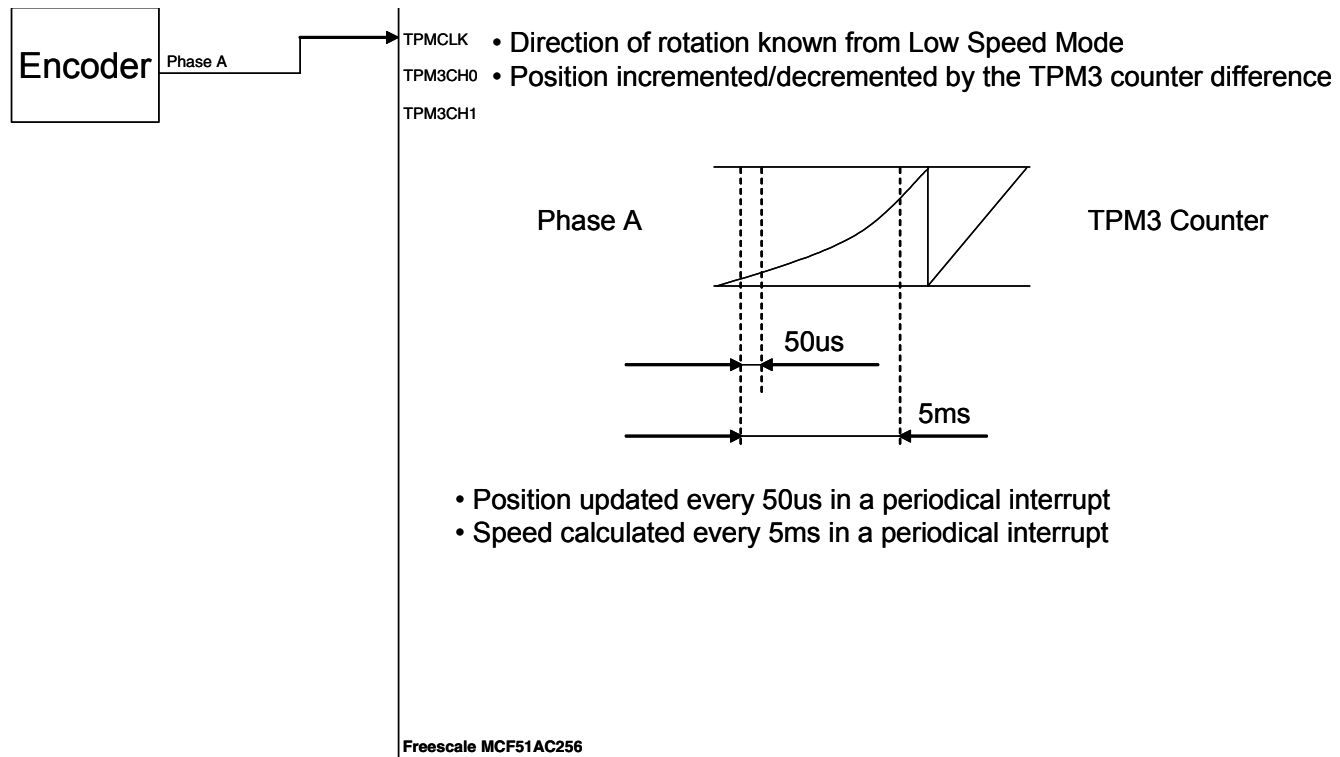


Figure 3. High Speed Mode

To read the position a periodical interrupt is necessary. In this case FTM2 was chosen where the period of the interrupt generation is set to 50 µs. When this interrupt is generated, the application subtracts the current TPM3 counter position from the previous position (which is stored in a variable from the previous interrupt). As the counter only counts one type of the edge (rising) and as just one channel is used, this information must be multiplied by a factor of four before it is used. The multiplied position difference is added to the position variable in the case of positive rotation and/or subtracted from the position variable in the case of negative rotation.

To evaluate speed in this mode a divider variable was put into the periodical interrupt to get a period of 5 ms for the speed calculation. The longer the period is, the more speed precision is gained, but the delay is longer. So an optimum value must be set up.

Knowing the period of speed evaluation interrupt and knowing the number of encoder pulses, we can state:

$$f_{rotor} = \frac{N_{eds}}{N} \quad \text{Eqn. 14}$$

where:

f_{rotor} — frequency of rotor (revolutions per 1 s)

N_{ed1s} — encoder edges per one 1 s

N — number of encoder pulses

$$N_{\text{ed1s}} = \frac{N_{\text{edt}}}{t_{\text{speed}}} \quad \text{Eqn. 15}$$

where:

N_{ed1s} — edges in the speed evaluation interval

t_{speed} — speed evaluation period

$$\omega_{\text{RPM}} = 60 \times f_{\text{rotor}} \quad \text{Eqn. 16}$$

where:

ω_{RPM} — speed in revolutions per 1 minute

Putting [Equation 14](#) and [Equation 15](#) into [Equation 16](#) we obtain the following equation:

$$\omega_{\text{RPM}} = 60 \times \frac{N_{\text{ed1s}}}{N \times t_{\text{speed}}} \quad \text{Eqn. 17}$$

To simplify this equation we can write the following equations:

$$\omega_{\text{RPM}} = K_2 \times N_{\text{edt}} \quad \text{Eqn. 18}$$

$$K_2 = \frac{60}{N \times t_{\text{speed}}} \quad \text{Eqn. 19}$$

where:

K_2 — speed constant

As mentioned above, this mode's precision is four times lower in comparison to low speed mode.

Nevertheless to obtain two times higher resolution in this mode, it is possible to use a XOR gate with inputs of the encoder's phase A and phase B signals and its output connected to the TPMCLK pin. In this case the precision would be only two times lower in comparison to low speed mode.

The speed evaluation precision can be calculated as follows:

$$\omega_{\text{RPMprec}} = \frac{60}{N \times t_{\text{speed}}} \quad \text{Eqn. 20}$$

So for the 1024-pulse encoder and for the speed evaluation period of 5 ms the result is:

$$\omega_{RPM_{prec}} = \frac{60}{1024 \times 5 \times 10^{-3}} = 11.7 RPM \quad \text{Eqn. 21}$$

So the system is able to detect the speed with a precision of ± 11.7 RPM. In case of the speed when the system is switched from low speed mode to high speed mode (that is, 146 RPM), the relative error is $11.7 / 146 \times 100 = 8\%$.

In high speed mode, the processor load is not very high, but in this mode it is not possible to recognize the direction of rotation. Therefore this mode can be used for a certain speed where it is absolutely sure that the velocity will not immediately reverse.

4.3 Transition between the Modes

As mentioned above, the application uses two modes of position evaluation. What was not mentioned is how the transition between these two modes is made.

From a standstill, the application is in low speed mode where the application is able to recognize the direction of rotation. As the absolute speed grows, the system must switch the speed mode to high speed mode at a certain speed threshold. When the speed reaches this threshold, the application must execute several steps that are important for safe mode switching:

1. In the periodical interrupt where the speed is updated every 5 ms, the application sets a command for a transition to high speed mode. At this moment the application is in a transition mode where the speed is not updated.
2. The transition is completed in the phase A interrupt when the phase A edge arrives. If the edge is rising, the timer setting is reconfigured in this way:
 - The TPM3 counter clock is switched to the external clock source.
 - The TPM3 interrupts are disabled.
 - The actual counter register of TPM3 is stored as the current step position.
3. Now the application is in high speed mode.

When the absolute speed falls below a certain threshold, the system is switched to low speed mode. The following steps are important for successful switching of the mode:

1. In the periodical interrupt where the speed is updated every 5 ms, the application sets a command for a transition to low speed mode. At the same time TPM3's channel 0 is configured for a rising edge capture only, its possible flag is cleared and interrupt enabled. At this moment the application is in a transition mode where the speed is not updated.
2. The transition is completed in the phase A interrupt when the phase A rising edge arrives. The following steps are taken:
 - The TPM3 counter clock is switched to the bus clock source.
 - The actual TPM3 channel 0 and channel 1 states are stored.
 - The TPM3 channel 1 possible flag is cleared and its interrupt enabled.
 - The TPM3 channel 0 is switched to capture both edges.

- The TPM3 overflow flag is cleared and its interrupt is enabled.
 - TPM3 channel 0's capture value is read and incremented/decremented by one because the capture time was not updated as the edge of clock occurred at the same moment.
3. The position variable is updated according to TPM3 channel 0's captured value.
 4. Now the application is in low speed mode.

As the transition is initiated according to the speed information that is updated every 5 ms and the system then waits for the rising edge, the system has certain limitations that consist in the slope of the speed. If the slope of the speed ramp is very steep and if the speed reverses, there is a risk of error in position evaluation. For safety reasons the slope of speed should be about 100 ms from the threshold speed to zero. The reason is that the system must switch to low speed mode before the speed is reversed — otherwise a few pulses can have the wrong sign.

4.4 TPM Configuration

To be able to use the TPM as the quadrature decoder it is necessary to make the following setting:

- PTBD bits 0, 1 to 0 (as input)
- PTBPE bits 0, 1 to 0 (pull-up disable)
- PTED bits 2, 3 to 0 (as input)
- PTEPE bits 2, 3 to 0 (pull-up disable)
- TPM3 C0SC = 0xC (channel 0 input capture on any edge)
- TPM3 C1SC = 0xC (channel 1 input capture on any edge)
- TPM3 CNT = 0 (counter reset)
- TPM3 SC = 0x8 (bus clock on TPM3, divider 1)
- TPM3 C0SC CH0IE to 1 (channel 0 interrupt enabled)
- TPM3 C1SC CH1IE to 1 (channel 1 interrupt enabled)
- TPM3 TOIE to 1 (overflow interrupt enabled)

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN3775

Rev. 0

09/2008

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

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 Semiconductor 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 Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.
© Freescale Semiconductor, Inc. 2008. All rights reserved.