SMC/ZPD implementation in TRAVEO™ T2G family

About this document

Scope and purpose

AN226036 describes the implementation of the Stepper Motor Control (SMC) and Zero Point Detection (ZPD) using the TRAVEO™ T2G Family MCU.

Table of contents

1 Introduction

This application note describes how to implement SMC along with ZPD using the TRAVEO™ T2G family CYT4D series MCU device. A stepping motor is typically used to operate the needle of an automotive meter. The SMC is implemented by using the counter and PWM functions of the TCPWM block in the TRAVEO™ T2G MCU. ZPD is a function to recognize the stop position in SMC. ZPD is implemented by using the pulse detection function of the ADC block in the TRAVEO™ T2G MCU.

To understand the functionality described and terminology used in this application note, refer to “Timer, Counter, and PWM” and “SAR ADC” chapters of the architecture technical reference manual (TRM).

2 About SMC and ZPD

This section provides an overview of SMC and ZPD.

Figure 1 shows the basic operation of a simple 4-step stepping motor with two coils.

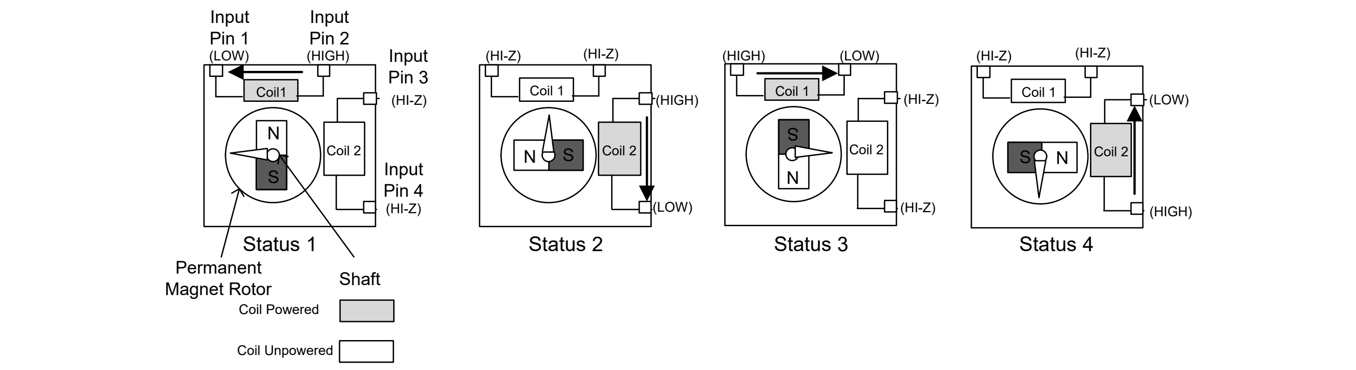

Figure 1 **Stepping motor basic operation**

In Figure 1, the stepping motor has two coils (Coil 1 and Coil 2) and a permanent magnet rotor with a shaft. The motor operates by the rotation of the permanent magnet rotor, which is done by powering one of the two coils at a time. Each coil has two inputs and the motor has a total of four signal input pins.

In the example shown in Figure 1, the permanent magnet rotor makes one complete clockwise rotation from status 1 to status 4 in steps of 90 degrees from one status to the next. In status 1, coil 1 is energized or powered by providing a HIGH signal at input pin 2 and a LOW signal at input pin 1. Coil 2 is unpowered by the High-Z signals at both its inputs. Next, in status 2, when coil 2 is powered as shown and coil 1 is unpowered by the High-Z signal inputs, the internal permanent magnet makes a 90-degree rotation in the clockwise direction. In status 3, coil 1 is powered in the opposite direction as compared to status 1, with coil 2 unpowered, which makes the magnet rotate further by 90 degrees. Finally, in status 4, coil 2 is powered but in the opposite direction compared to status 2 with coil 1 unpowered, making the permanent magnet rotor rotate further by 90 degrees. Thus, by successively changing the powering of the coils, the rotor makes one complete (360 degrees) clockwise rotation and the stepping motor operates.

Figure 2 shows the block diagram of the stepping motor controlled by the TRAVEO™ T2G MCU.

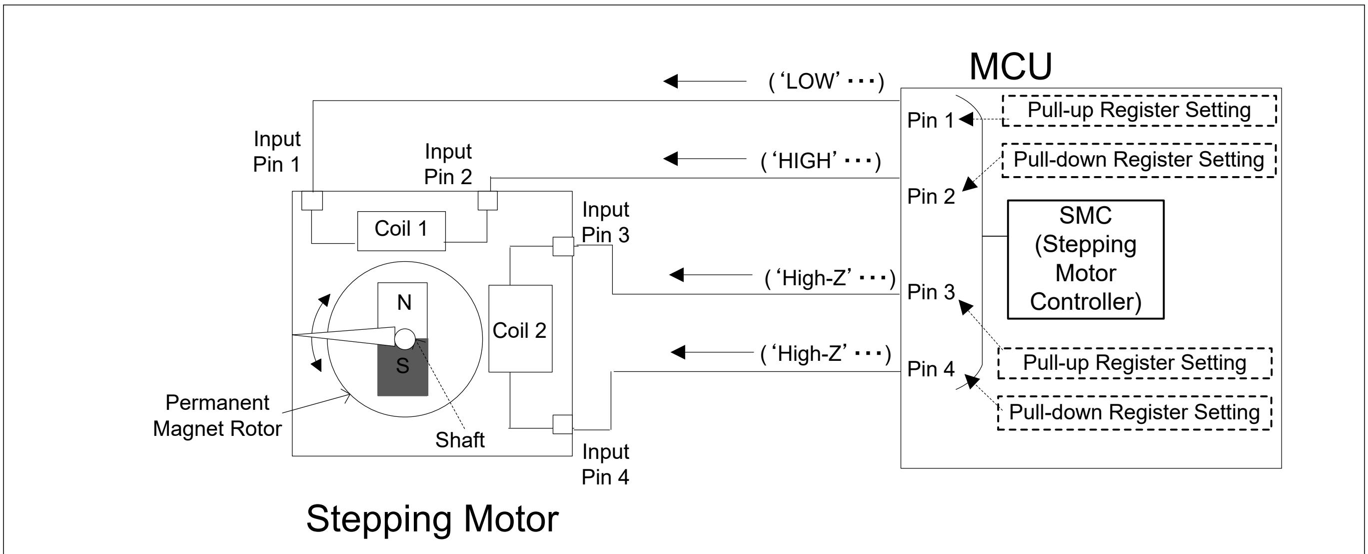

Figure 2 **Block diagram stepping motor and MCU**

Figure 2 shows the operation of the same stepping motor shown in Figure 1 controlled by the outputs of the MCU. These output signals are set to LOW, High-Z, or HIGH. The MCU operates the stepping motor by outputting LOW, High-Z, and HIGH signals in a fixed pattern to the four input pins. Thus, SMC refers to this operation of outputting a signal to the motor and changing the signal. The MCU pin 1 through pin 4 need to have internal pull-up and pull-down resistors enabled via register settings.

Figure 3 shows the MCU port settings needed for the four pins, pin 1, pin 2, pin 3, and pin 4 connected to the stepping motor inputs.

As shown in circuit A of Figure 3, pin 1 and pin 3 are set to pull-up settings (DRIVE_MODE = 2) in the internal GPIO register. Pin 2 and pin 4 are set to pull-down settings (DRIVE_MODE = 3) in the internal GPIO register. This is to set a middle level for the High-Z signal. Circuit A measures the ADC at the pins connected to both sides of the coil. As shown in circuit B of Figure 3, pin 1 and pin 3 are set to pull-up/down settings (DRIVE_MODE = 1) in the register. Pin 2 and pin 4 are set to strong settings (DRIVE_MODE = 6) in the register. Circuit B measures ADC only on the pin connected to one side (pin 2 or pin 4) of the coil.

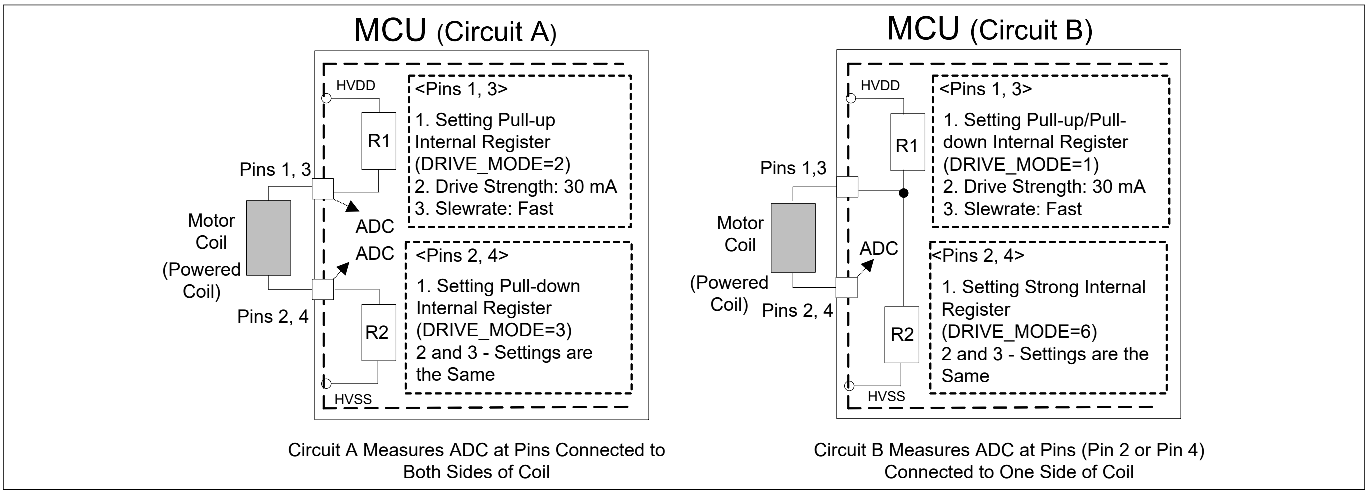

Figure 3 MCU port settings from Pin 1 to Pin 4

Figure 4 shows ZPD in the stepping motor.

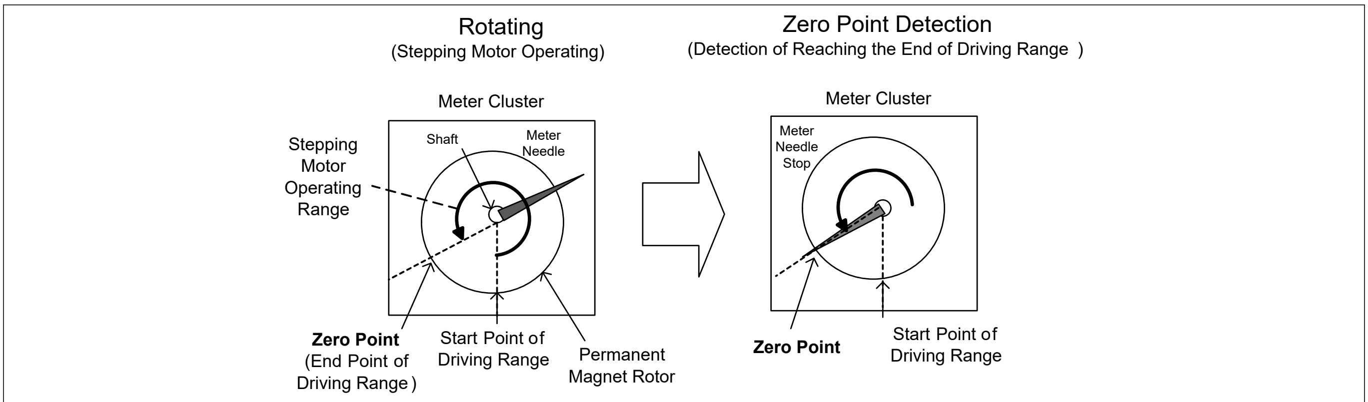

Figure 4 Zero point detection

An example of a stepper motor used for automotive meter cluster is shown in Figure 4. In this case, it is necessary to recognize the actual stop position after the motor rotates. In this application note, the end point of the driving range is defined as Zero Point. Detecting Zero Point is called ZPD. ZPD can be recognized by observing the input High-Z signal of the stepping motor.

3 Operation overview

3.1 Operating stepping motor

This section explains how to operate the stepping motor. The stepping motor is connected to the MCU with four terminals as shown in Figure 2. In this application note, the two input terminals of coil 1 are defined as pin 1 and pin 2, the two input terminals in coil 2 are defined as pin 3 and pin 4. Table 1 lists the signal patterns at pins 1 to 4.

Table 1 Stepping motor pin input signal pattern

MCU and Motor Connection Pins	Step 1 (Signal)	Step 2	Step 3	Step 4
Pin 1 (Coil1)	LOW	High-Z	HIGH	High-Z
Pin 2 (Coil1)	HIGH	High-Z	LOW	High-Z
Pin 3 (Coil2)	High-Z	HIGH	High-Z	LOW
Pin 4 (Coil2)	High-Z	LOW	High-Z	HIGH

For pins 1 to 4, the signal input to each pin is either LOW, High-Z, or HIGH signal. If the signal input to coil 1 (pin 1, pin 2) is LOW or HIGH, the input to coil 2 (pin 3, pin 4) is High-Z. Similarly, if the signal input to coil 1 is High-Z, the input to coil 2 (pin 3, pin 4) is HIGH or LOW.

These LOW, High-Z, and HIGH signals are output by TCPWM in the MCU. The MCU output signal to one pin is repeated at LOW, High-Z, HIGH, High-Z transition. These statuses are transitioned by the overflow interrupt of TCPWM. According to the pattern repeating in the Table 1, it is possible to drive a stepping motor.

3.2 Detecting zero point

This section explains how to detect the zero point. Figure 4 shows an overview of ZPD. In this application note, the pulse detection of the ADC is used for ZPD in a High-Z signal.

Figure 5 shows the pulse detection function of the ADC. The pulse detection can filter by counting the events of range detection. It has a positive counter and a negative counter. For the range detection threshold (RANGE_HI) of the ADC, the upper threshold is positive and the lower threshold is negative. When the positive event happens, the positive counter decrements. When the positive counter is underflow, the pulse detection interrupt occurs. Similarly, for a negative counter, a pulse detection interrupt occurs when an underflow occurs. After either of counter underflows, both counters are reloaded. This function detects the ZPD by setting a threshold for the High-Z signal.

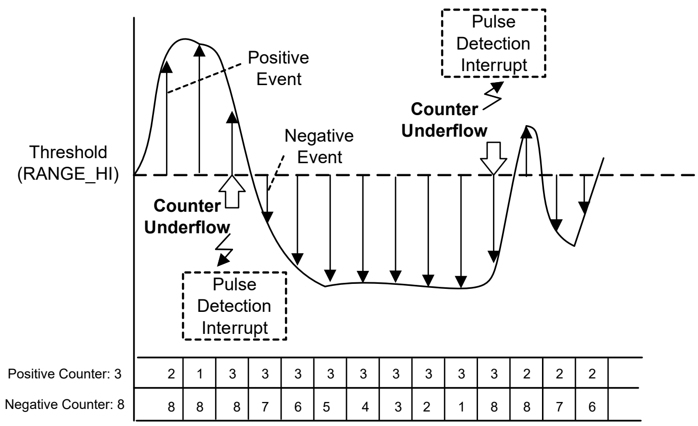

Figure 5 Example pulse detection function of the ADC

For details of the ADC pulse detection setting, see the "SAR ADC" chapter of the architecture TRM.

Figure 6 shows an example of pin 1 signal High-Z for ZPD.

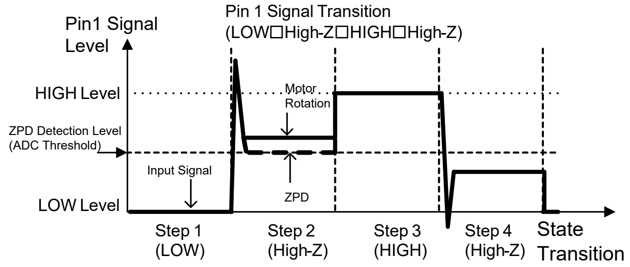

Figure 6 Example of Pin 1 Signal High-Z for ZPD

When the stepping motor operates, focus is on the High-Z signal in step 2 waveform as shown in Figure 6. In High-Z, set the pull-up/down setting to adjust the waveform of the middle level. The same waveform is output to pin 3. The adjustment of this middle level needs to be performed for both pin 1 and pin 3. When the motor stops, a slight drop in the middle level appears. This level is observed by a waveform. This point uses the ADC pulse detection function to set the threshold as a detection level.

In step 2 (High-Z), as shown in Figure 6, the waveform appears to rise momentarily in the first part. This is the back electromotive force (BEMF) characteristic of connecting the motor.

4 Operation example with use case

This section explains the timing chart of SMC by four signal inputs of one motor and ZPD by TCPWM and ADC. Figure 7 shows the timing chart of an example of SMC and ZPD implementation for one motor.

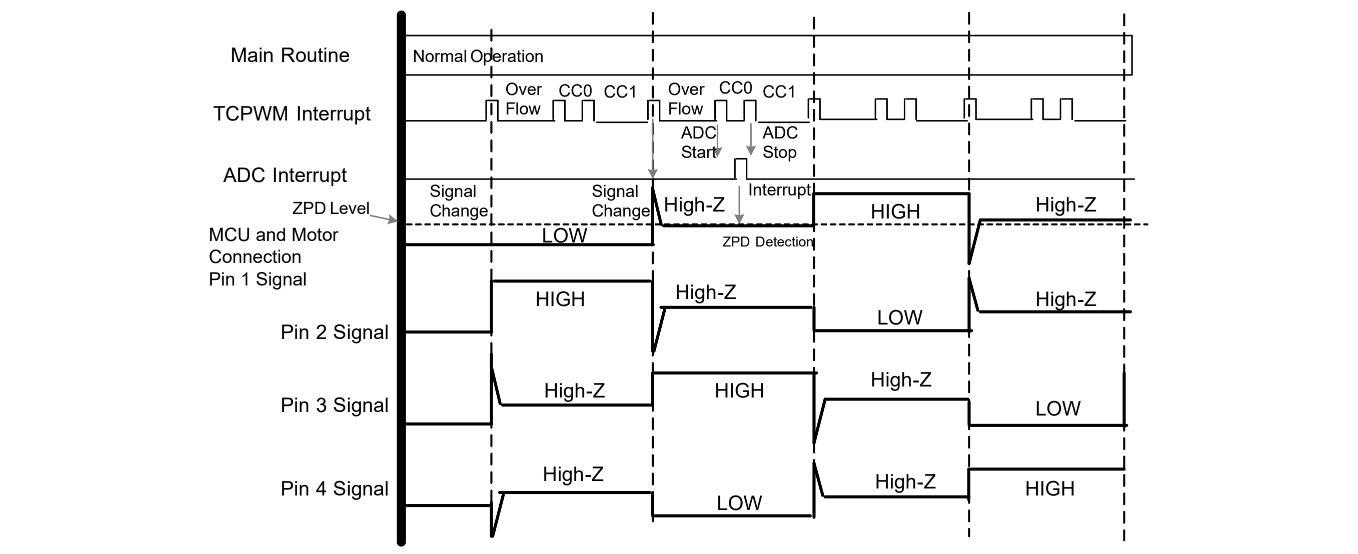

Figure 7 Example of SMC and ZPD implementation timing chart for one motor

Figure 7 shows the signals for motor pins 1 to 4 and the timing chart of the TCPWM and ADC interrupt. Figure 7 also shows an example of implementation of ZPD by TCPWM and ADC for motor 1 pin. First, the initialization routine initializes TCPWM and ADC. Then, when the TCPWM counter overflows, an overflow interrupt occurs. The MCU signal changes when the overflow interrupt occurs. TCPWM uses two compare match functions (CC0, CC1) for starting and stopping the ADC. If the TCPWM counter matches the count value of CC0 or CC1, a compare match interrupt occurs. A compare match (CC0) interrupt occurs in High-Z of the step 2 as shown in Figure 6. CC0 interrupt starts ADC. When the ADC is complete, an ADC interrupt is generated. In this ADC interrupt, it is possible to check ZPD by the threshold of the pulse detection function. If the middle point of the High-Z signal drops below the threshold, it is recognized as the state of ZPD. Therefore, it is possible to detect the rotating stop of the motor. When the second compare match (CC1) interrupt occurs, ADC stops.

Figure 8 shows the timing chart of an example of SMC and ZPD for six stepping motors.

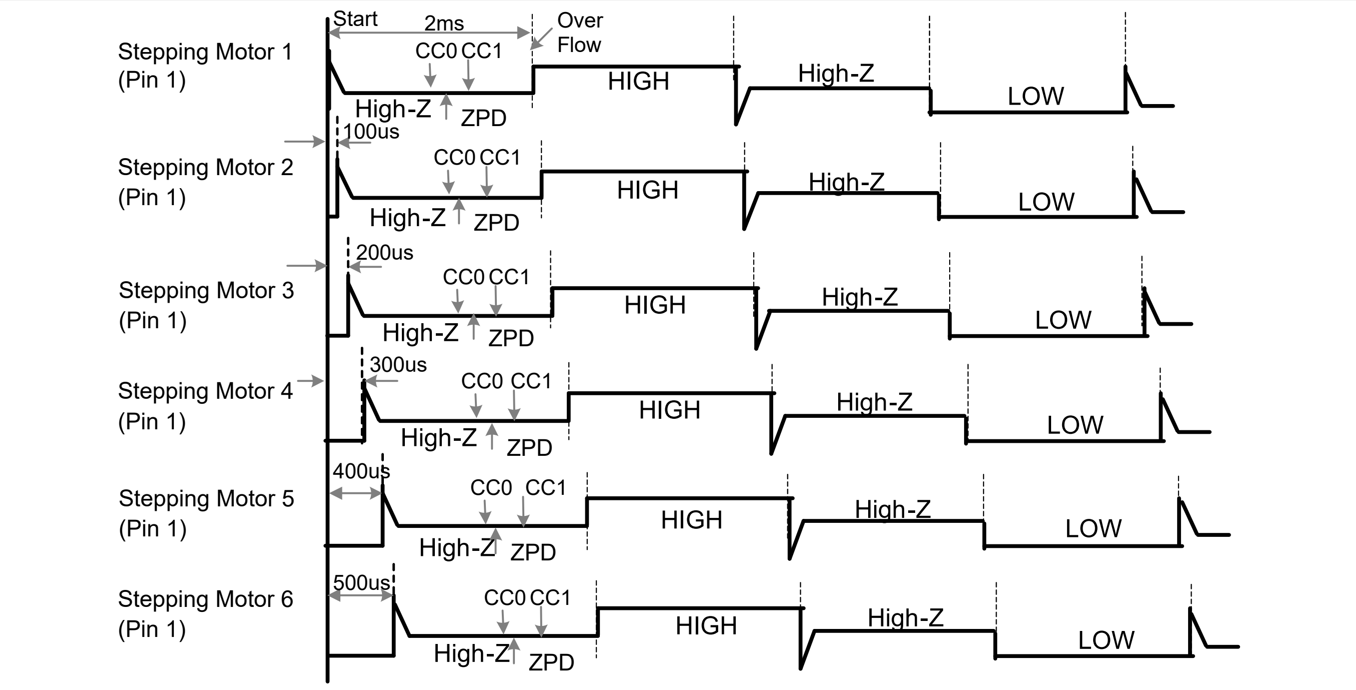

Figure 8 Example of the SMC and ZPD timing chart for six stepping motors

Figure 8 shows the SMC signal and timing for pin 1 of each of the six stepping motors.

As shown in Figure 8, ZPD detection of ADC in the six stepping motors connection is performed without overlap by delaying the start time of operation. Therefore, it is possible to control all six stepping motors.

Figure 9 shows the flow of SMC and ZPD implementation. This example shows the case for pin 1 of one motor.

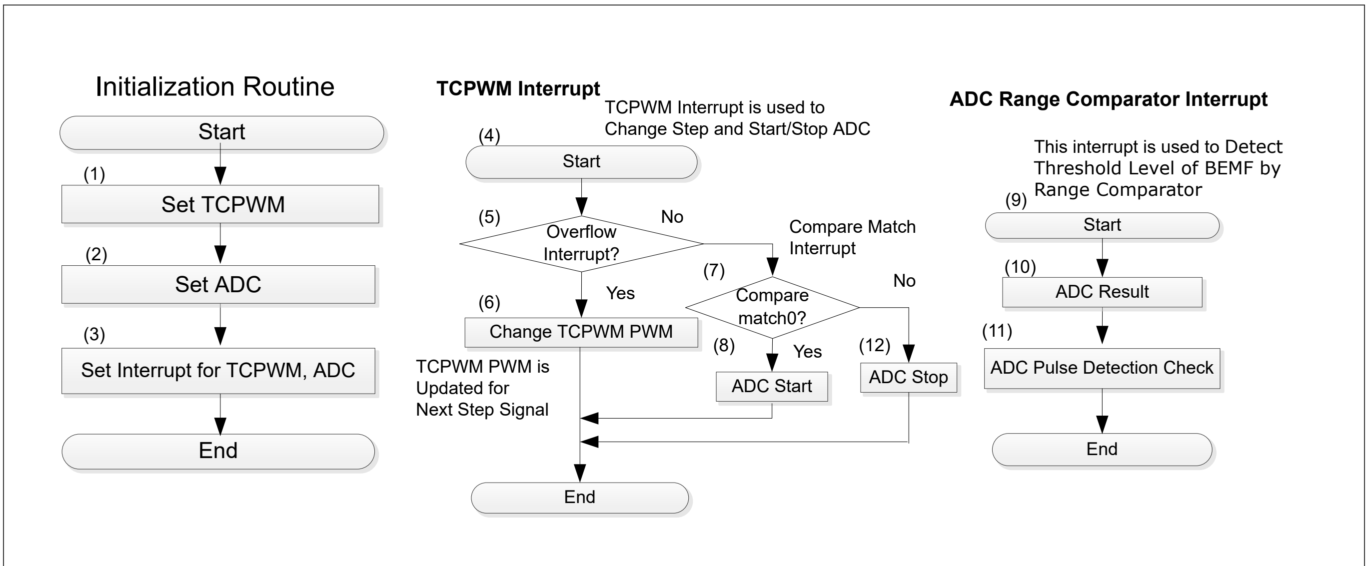

Figure 9 Example of SMC/ZPD implementation flow

Table 2 list MCU Peripheral function for one SMC and ZPD implementation.

Table 2 MCU peripheral function for one SMC and ZPD implementation

MCU peripheral function	Use for implementation
TCPWM0	TCPWM0 used for updating step state and ADC activation and ADC stop.
TCPWM1	TCPWM1 used for signal setting.
ADC0	ADC used for pulse detection to ZPD pin 1.
ADC1	ADC used for pulse detection of ZPD pin 3.

The following is the procedure for the flow shown in Figure 9:

1. In the initialization routine, initialize the counter and PWM of the TCPWM for SMC. The counter is used for interrupts. The PWM is used for signal setting and updating. Set the Initial signal to the PWM.

2. Initialize the ADC for ZPD. ADC is used in High-Z to get AD conversion result and use pulse detection function.

3. Set the interrupt. TCPWM is set to use overflow interrupt and compare interrupt. ADC is set to use ADC completion interrupt.

4. When the TCPWM counter overflow interrupt occurs, TCPWM interrupt routine is executed.

5. The interrupt routine checks if the interrupt is caused by the overflow or by the compare match. If it is overflow, go to step 6. If not, go to step 7.

6. TCPWM interrupt move to the next step and change the signals.

7. Check for compare match (CC0, CC1) factor. If it is compare match (CC0), go to step 8. If it is CC1, go to step 12.

8. Start the ADC.

9. When the ADC completes, the ADC interrupt occurs.

10. ADC interrupt gets the result of AD conversion of High-Z signal

11. Check for pulse detection function to detect ZPD.

12. Stop the ADC.

13. By repeating step 4 to 12 above, it is possible to implement stepping motor operation and ZPD.

5 Proposal for calibration of ZPD

Here is a sample procedure for calibrating ZPD.

1. Define separate threshold levels for each motor in the software.
2. Rotate the motor until stall and measure the waveform with ADC.
3. Send the data of waveform to external test equipment and calculate the safety margin for the threshold level of range comparator.
4. Write the calibrated threshold level back into Workflash.
5. Restart the application software with calibrated values and check the zero point of needle with visual inspection.

6 Glossary

Table 3 Glossary

Terms	Description
ADC	Analog Digital Conversion. See the "SAR ADC" chapter of the architecture TRM architecture TRM for details.
MCU	Microcontroller Unit
SMC	Stepper/Stepping Motor Controller
TCPWM	Timer, Counter, and Pulse Width Modulator. See the "Timer, Counter, and PWM" chapter of the architecture TRM for details.
ZPD	Zero Point Detection

7 Related documents

The following are the TRAVEO™ T2G family series datasheets and Technical Reference Manuals. Contact Technical Support to obtain these documents.

- Device datasheet
  - CYT4DN Datasheet 32-Bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G Family
- CYT4D Series
  - TRAVEO™ T2G Automotive Cluster 2D Family Architecture Technical Reference Manual (TRM)
  - TRAVEO™ T2G Automotive Cluster 2D Registers Technical Reference Manual (TRM)

Revision history

Document version	Date of release	Description of changes
**	2020-03-02	New application note.
*A	2021-03-11	Updated to Infineon template.
*B	2023-11-20	Template update; no content update

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-11-20

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-vst1685623709660