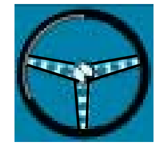
Speed Measurement



Return to Home Page

A Motorola Low-Level Driver Component

The Speed Measurement (SPM) driver provides the application with the frequency of an input signal. The application defines the beginning and end of a time window by issuing a read request, and the SPM driver returns the number of pulses seen in that window, as well as the time accumulation of pulse periods seen in the window and the timestamp of the most recent edge.

One example application is the speed measurement of a rotating shaft. In this case, the input pulse train could come from a sensor monitoring teeth mounted around the perimeter of a rotating shaft.

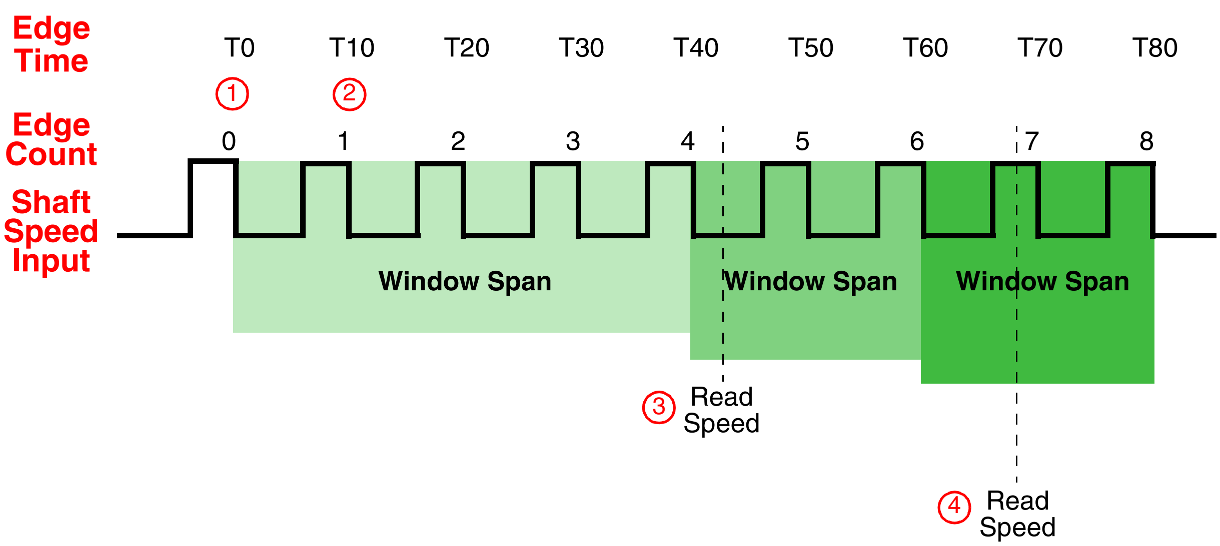
The sensor detects the passing of teeth distributed evenly around the shaft and produces a digital signal whose frequency is directly proportional to shaft rotation speed; the number of input pulses per shaft revolution is fixed.

SPM can be initialized to detect rising edges, falling edges, both edges, or the sense of the first edge detected after initialization.

After initialization, the SPM driver runs autonomously from any other channel functions or operations. At each detected edge, the driver increments an edge count, updates the edge timer, and accumulates the time since the last read request.

To obtain the current speed, the application software issues an API call to the SPM device driver, which calculates the number of input periods since the last read command and the total time spanned by those periods. The application software can then use these two values to calculate the rotational speed of the shaft.

The diagram below illustrates the normal operation for the SPM driver, assuming it is initialized to detect falling edges only. The circled numbers represent actions or events this example illustrates, and are described in the table on the back of this datasheet.



①	At the first edge detected after initialization, the SPM driver sets all internal variables to 0 and starts the timer.
②	At each subsequent tooth transition, the driver updates its internal variables to reflect the current edge count and timer value.
③	The application software requests a Read Speed. The driver uses the first edge detected after initialization and the active edge immediately preceding the Read Speed request to define the beginning and end of the window span.
④	The application software requests a second Read Speed. The driver defines this window span as the time between the last Read Speed request's timestamp (from Step ③) and the edge preceding the current request.

Because of its self-sufficiency, the SPM driver includes only two function calls. One function call initializes the driver and sets the base parameter values. The second function call allows the application software to request a read of the current speed at any time.

The read speed function returns data for computation of shaft speed. The returned data can be used to calculate the shaft speed based on the following formula:

Shaft speed (i.e., revolutions/second) =
Window Edge Count $\times K \div$
(Edges/Revolution $\times$ Window Span)

Where:

- **Edges/Revolution (i.e., how many teeth there are per wheel) is a constant known to the application**
- **K is the inverse of the resolution used for the TPU's timer count register.**
  As an example, if the application software has configured the TPU's counter to increment every microsecond, the application software would set $K = 10^6$.

The Low Level Driver System

The Low Level Driver system includes a set of drivers with an API that interfaces to and controls the hardware for a microcontroller unit (such as the Motorola MPC555)

Engine Position

Tracks the angular position in the engine cycle based on input from an automobile's crankshaft and camshaft sensors

Spark & DTS

Generates pulses defined by duration and end angle; can be used to time the firing of spark plugs

Fuel

Generates pulses immediately upon request or defined by duration and end angle; can be used to control fuel injection duration and frequency

Speed Measurement

Determines the speed of a rotating shaft

Synchronous PWM

Synchronizes an output pulse width modulation (PWM) signal to an input PWM signal

Synchronous Output

Transmits a clock signal and serial data, following a specific protocol

Angle Toggle

Toggles an output pin and generates interrupts on selected crank angles

QADC Trigger

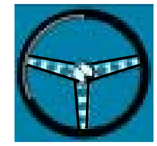
Generates pulses defined by a start angle and duration

Knock Window

Generates pulses defined by a start and end angle

Discrete Input/Output (DIO)

Operates as a general-purpose digital input or output pin



Return to Home Page

For more information, contact your local Motorola sales representative or:

Motorola, Inc.
Advanced Vehicle Systems Division
Attention Software Operations
6501 William Cannon Drive West, Mail Drop OE-39
Austin, TX 78735-8589