Volume : 2 | Issue : 5 | May 2013 • ISSN No 2277 - 8179

Research Paper

Underwater Vehicle RPM Measurement System Using Programmable System on Chip (PSoC)

Engineering

KEYWORDS : PSoC, CY8C27443, Optical sensor, RPM Test panel.

Yenni Venkateswara Rao — N Sampath Kumar, plot no 49, Chamundeswari nagar, Thylavaram, Near SRMUniversity, Kancheepuram, Tamil Nadu-603203, India

M Srinivasa Rao — Scientist 'E', CSS Division, NSTL, Vishakhapatnam, Andhra Pradesh - 530016, India

ABSTRACT

PSoC family consists of many Mixed Signal Array with On Chip Controller devices. PSoC devices include configurable blocks of analog and digital logic, as well as programmable interconnects. Using RPM, the speed of the vehicle is determined. RPM measurement is important when controlling or monitoring the speed of motors, conveyors, turbines, etc. It uses digital counters with interrupt service routines to accurately measure the RPM of a rotating wheel. Optical sensor is being used to sense the RPM signal and a PSoC device is used to process the sensed signal to count the RPM. The number of pulses from the optical sensor is measured using counters of PSoC and display the RPM value on the serial port of PC or LCD. This provides a cost-effective solution in industries and R&D.

Summary: Finally, this method has been proposed to use the Programmable System On chip which a single PSOC device replaces a wide variety of discrete components and dramatically reducing board size, chip count, and cost while improving manufacturability. These simple in-expensive designs accurately measure the RPM of a rotating wheel, which requires only a 5-volt power supply. Pulse counting is an essential application requirement in industry and R&D laboratories. This provides a cost-effective solution in industries and R&D.

1. INTRODUCTION

Programmable System on Chip (PSoC) represents a whole new concept in microcontroller development. Programmable System on Chip devices and PSoC Designer provide a more flexible solution for creating the required functionality on a single device. The PSoC family consists of many Mixed-Signal Array with On-Chip Controller devices. PSoC devices include configurable blocks of analog and digital logic, as well as programmable interconnects. PSoC devices are designed to replace multiple traditional MCU-based system components with one, low cost single-chip programmable device.

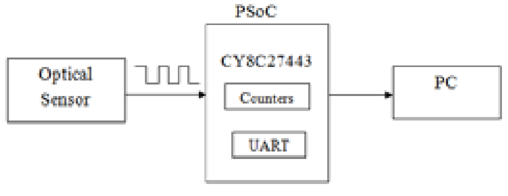


Figure 1.Block diagram of RPM Measurement system.

In industries as well as in R&D laboratories the measurement and control of variables such as pressure, temperature, flow, position, rotation of a wheel, and others demands precise measurement and control. This design accurately measures the speed, in RPMs, of a rotating wheel using optical sensor connected to a PSoC device. Using RPM, the speed of the vehicle is determined. This provides a cost-effective solution using a single-chip PSoC. Optical sensor is being used to sense the RPM signal and a PSoC device is used to process the sensed signal to count the RPM. The number of pulses from the optical sensor is measured using counters of PSoC and display the RPM value on the serial port of PC or LCD. It uses digital counters with interrupt service routines to accurately measure the RPM of a rotating wheel.

2. Optical Sensor

A sensor is necessary to sense shaft speed. Sensors play a vital role in the front end of the design. To measure rotations per minute of a rotating wheel, an optical sensor is also used as a source detector for generating TTL pulses per revolution. Optical sensors include light-emitting diode (LED), light-sensitive phototransistor (photo cell), and a slotted disc called a light beam interrupter. The slotted disc rotates between the LED and the photo cell. Whenever light hits a photo cell an electric signal is generated.

Figure 2.Optical sensor SX670.

When there is a slot between the LED and photo cell, light passes through the slot and falls on the photo cell, causing the photo cell to produce a full voltage. When the LED is blocked causing the photo cell to produce a no voltage. The on-and-off reflection of the LED beam into the photocell produces a pulsating current; it produces TTL pulses at its output while rotating. The PSoC device is programmed to convert the number of pulses from the photo cell to an electronic measurement of vehicle speed.

3. RPM Measurement Techniques

RPM measurement is important when controlling or monitoring the speed of motors, conveyors, turbines, etc. Two factors affect the quality of this data:

- Number of pulses per revolution (PPR) of the shaft. Higher PPR values result in better resolution.
- Symmetry of pulses. The symmetry of one pulse to the next can play a role in how consistent the RPM readings are. Symmetrical pulses give more accurate data.

There are two methods for determining RPM

1) Frequency measurement method (Measure the number of cycles in a fixed period of time).
2) Period measurement method (Measure the time of one cycle).

4. Frequency Measurement Method

The frequency, F, is the number of cycles of that signal per unit of time, represented by the equation: $F= n / t$. It is apparent that the accuracy of the frequency measurement is dependent on the accuracy in which t is determined. If t = 1 second, then the frequency is expressed as n cycles per second.

$$RPM = \frac{(\text{Pulse Frequency in pulses/sec}) \times (60)}{(\text{Sensor pulses/revolution})} = \frac{\text{Revolutions}}{\text{minute}} \quad \text{-- (1)}$$

This Application describes how a two 16-bit Counter User Module, with its interrupt service routine, are used to count the pulses for one second. It contains two 16 bit counters. One counts the number of input cycles and the other generates exact one second clock pulses.

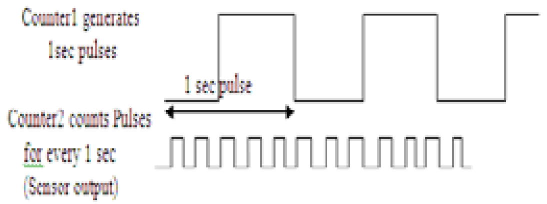


Figure 3.Pulses from counters.

The frequency counting method is to count an exact one second and count the number of input cycle trigger events that occur during that second. The counter contains the number of cycles counted during exactly one second, which is a sort of frequency (cycles/sec).

In this Application, PSoC Designer uses six digital blocks for two counters, and serial communication (UART) for interfacing with the PC. The processor clock is divided by using VC1, VC2 and VC3 to generate a clock for counter16_1 and the serial communication UART. The baud rate is set to 19,200 in this design application. Two digital blocks are used for serial communication with the PC through the UART. The design tools play a vital role in sourcing user module functionality and providing interaction with the built-in M8C microcontroller. A fixed microcontroller does not have the flexibility to dynamically set the baud clock for UART communication.

The transmit block (TX) is placed in DCB12 and the receive block (RX) is placed in DCB13 for serial communication. In this application, the UART baud rate is set to 19,200. The serial communication transmits through the Row_1 Output_3 line and the GlobleOutEven_7 bus that is connected to port 2_7 pin on the PSoC , and wired to receives through the Row_1 Input_2 line and the GlobleInOdd_6 bus that is connected to port 1_6 pin on the PSoC device. The PSoC device interfaces with the PC through the serial port at the baud rate of 19.2k.

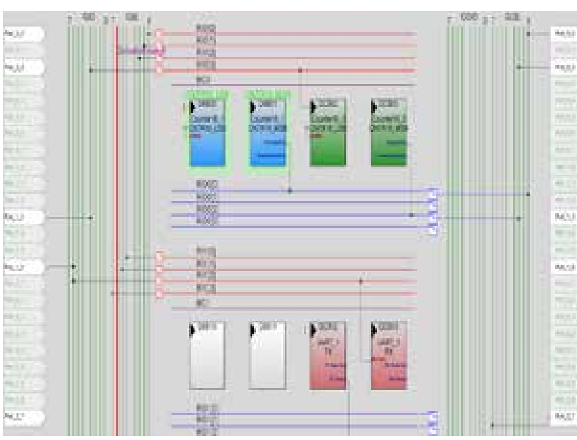


Figure 4.PSoC Designer User Module Placement.

This application uses two 16-bit counters and UART PSoC device resources. The Counter16_1 that occupies two digital communication blocks (DBB00 and DBB01) is programmed to generate one second pulse to set the counter for counting pulses. These blocks are wired to generate one second pulses through the Row_0 Output_0 line and the GlobleOutEven_3 bus that is connected to port 0_0 pin on the PSoC device. Counter16_2 that occupies two digital communication blocks (DCB02 and DCB03) is wired to receive the optical sensor pulses through the Row_0 Input_3 line and the GlobleInOdd_3 bus that is connected to port 1_3 pin on the PSoC device. The starting and stopping of the counter activates the one second timer interrupt service routine written in 'C' in the Application Editor. The counter16_1 in the main.c program is generated timer interrupt service routine for every one second counted. This enables the 16-bit counter to count for 1 second.

Boot.tpl must be modified in the root project directory to implement the jump vector for interrupt. The interrupt comes from last block in counter chain, in this case DBB01, so that jump vector must be modified to "ljmp _Counter16_1_ISR". In main () flag is tested, if true, tells main () to service counter. Two Counters are stopped and read the value from Counter16_2. Get the counter value. Stopping counter allows the reload to immediately update counter. Subtract counter read value from full counter value (0xFFFF), that result is counts pulses in 1 sec.

The one second counts are multiplied by 60; that provides the count for 1 minute and hence the counts per minute can be achieved. The count value is converted into string and written to UART or LCD.

5. Period measurement method

Measuring the pulse period is the best method of measuring RPM. Period is the time from the start of one pulse to the start of the next pulse. The period, P, of an input signal is the inverse of its frequency. Period measurement allows more accurate measurement of unknown low-frequency signals.

The main issue when using Period measurements occurs when the PPR is greater than one and the pulses are not symmetrical. For example, when shaft speed is constant and sensing two bolt heads per revolution, if the bolts are not exactly evenly spaced, the periods will be different, causing the RPM indication to be erratic.

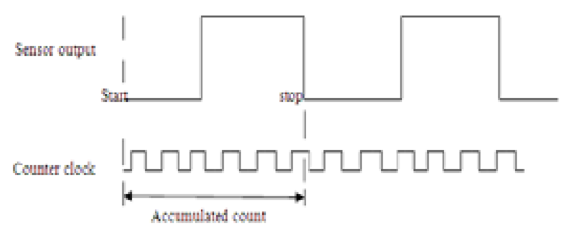


Figure 5.Time interval measurement.

A time interval measurement is a measurement of the elapsed time between some designated START phenomena and a later STOP phenomena. , It contains two counting registers. One counts the number of input cycles and the other counts the clock pulses, to measure the time duration. To obtain the mean frequency value, the following division is made:

$$\text{Frequency} = \frac{\text{Counted input cycles}}{(\text{counted clock pulses}) \times t} = \frac{N}{MT} \quad \text{----- (2)}$$

Where t is the time of one clock cycle.

A time interval measurement is a measurement of the elapsed time between some designated START phenomena and a later STOP phenomena. In this role, the counter makes an elapsed time measurement between two electrical pulses. During this time interval, the optical sensor gives output to the Counter16 and generates pulses, counts the number of input cycles while the Counter24 would have accumulated a number of clock pulses. The number of accumulated clock pulses is multiplied by the clock period to give the period of the input signal. When using period measurement to monitor RPM, calculate the RPM using this equation:

If counted input cycle is one then

$$RPM = \frac{1*60}{\text{Pulse Period x Pulse Per Revolution}} \quad \text{--- (3)}$$

It contains one 16 bit counter and one 24bit counter. 24 bit counter counts the number of clock cycles and the other 16 bit counter generates pulses from sensor output, counts the number of input cycles. This method is to count the number of clock cycle trigger events that occur during the time from the start of one pulse to the start of the next pulse of sensor output. This Application describes how a 16-bit Counter and 24 bit counter User Module, with its interrupt program routine, are used to count the pulses for elapsed time between some designated START phenomena and a later STOP phenomena.

In this Application, PSoC Designer uses all digital blocks for counters, and serial communication (UART) for interfacing with the PC. The counter register is reloaded whenever the counter reaches the terminal count. The processor clock is divided by using VC1, VC2 and VC3 to generate a clock for counter and the serial communication UART. The baud rate is set to 19,200 in this design application.Two digital blocks are used for serial communication with the PC through the UART.

The transmit block (TX) is placed in DCB12 and the receive block (RX) is placed in DCB13 for serial communication. In this application, the UART baud rate is set to 19,200. The serial communication transmits through the Row_1 Output_3 line and the GlobleOutEven_7 bus that is connected to port 2_7 pin on the PSoC , and wired to receives through the Row_1 Input_2 line and the GlobleInOdd_6 bus that is connected to port 1_6 pin on the PSoC device. The PSoC device interfaces with the PC through the serial port at the baud rate of 19.2k.

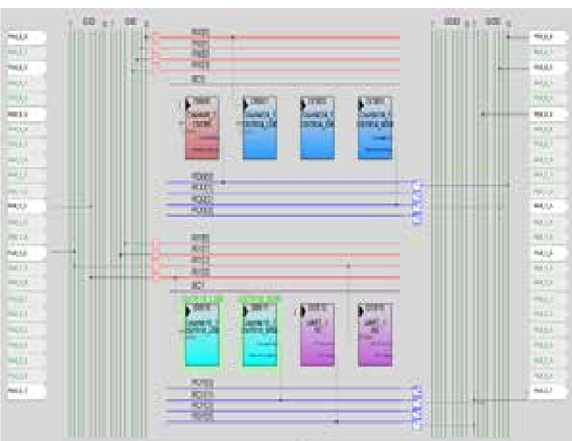


Figure 6.PSoC Designer User Module Placement

This application uses one16-bit counter, one 24-bit counter and UART PSoC device resources. The 16-bit Counter that occupies two digital communication blocks (DBB10 and DBB11) is programmed to generate pulses from output of sensor to set the counter for counting pulses. These blocks are wired to receive the optical sensor pulses through the Row_0 Input_3 line and the GlobleInOdd_3 bus that is connected to port 1_3 pin on the PSoC device. An 8-bit counter user module placed in digital block DBB00 is used as the clock generator, which generates a 100 kHz clock from its basic clock source. The 24-bit counter is programmed to generate 10 microsecond pulses.A 24-bit counter that occupies three digital communication blocks (DCB01, DCB02 and DCB03) is wired to receive clock through the Row_0 Input_0 line and the GlobleInEVEN_0 bus that is connected to port 0_0 pin, counts pulses through the Row_0 Output_2 line and the GlobleOutEven_2 bus that is connected to port 0_2 pin on the PSoC device.

The starting and stopping of the counter activates the elapsed time between some designated START phenomena and STOP phenomenons interrupt service routine written in 'C' in the Application Editor. The counter16_1 in the main.c program is generated interrupt service routine for every the start of one pulse to the start of the next pulse. This enables the 24-bit counter to count for particular period.Boot.tpl must be modified in the root project directory to implement the jump vector for interrupt. The interrupt comes from last block in counter chain, in this case DBB11, so that jump vector must be modified to "ljmp _Counter16_1_ISR".

In main () flag is tested, if true, tells main () to service counter. Two Counters are stopped and read the value from Counter24_1. Get the counter value. Stopping counter allows the reload to immediately update counter. Subtract counter read value from full counter value (0xFFFF), that result is counts pulses in time from the start of one pulse to the start of the next pulse. The number of accumulated clock pulses is multiplied by the clock period (10 microseconds) to give the RPS of the input signal. The one second counts are multiplied by 60; that provides the count for 1 minute and hence the counts per minute can be achieved. The count value is converted into string and written to UART or LCD.

5. Results

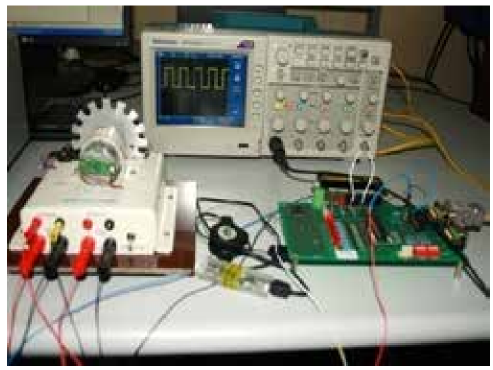
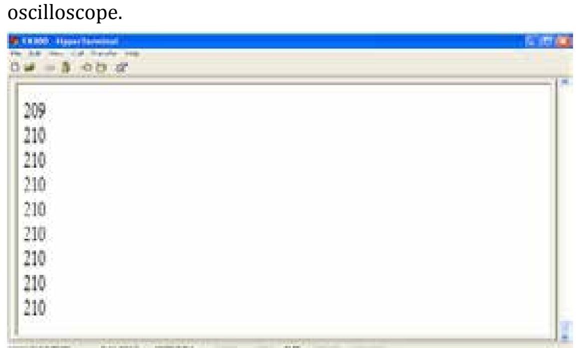


Figure7.RPM Test Panel and Cypress Kit.

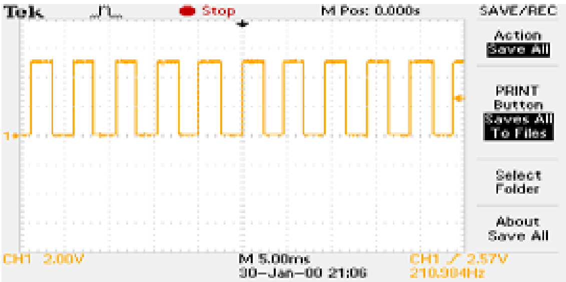
Figure 8. Testing RPM with the help of Oscilloscope.

In the following figures figure (a) shows the RPM value on the hyper terminal of PC and figure (b) shows the RPM value in the oscilloscope.



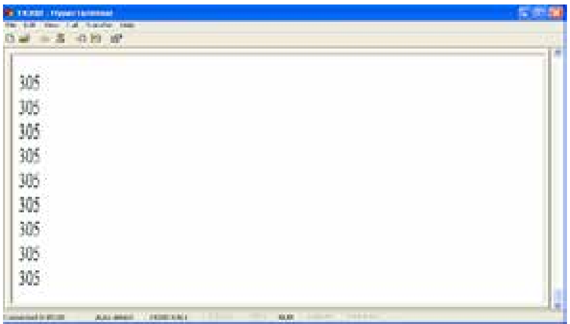
(a)

Volume : 2 | Issue : 5 | May 2013 • ISSN No 2277 - 8179

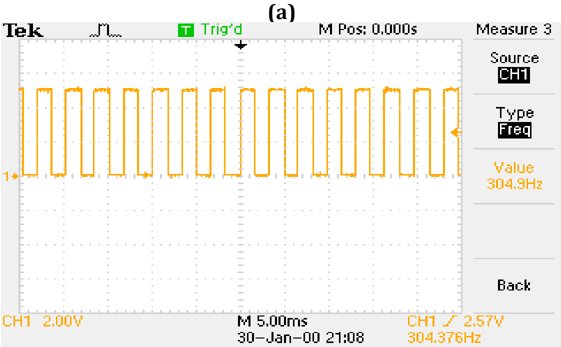


(b)

Figure 9: RPM values at 5v motor speed.

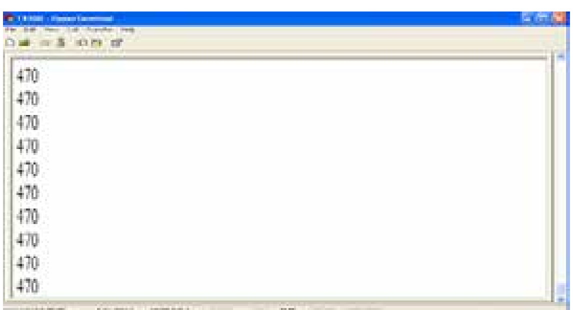


(a)



(b)

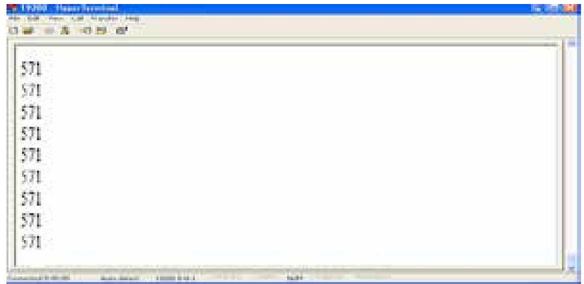
Figure 10: RPM values at 7v motor speed.



(a)

(b)

Figure 11 : RPM values at 10v motor speed.



(a)

(b)

Figure 12 : RPM values at 12v motor speed.

6. Conclusion

Finally, this method has been proposed to use the Programmable System On chip which a single PSOC device replaces a wide variety of discrete components and dramatically reducing board size, chip count, and cost while improving manufacturability. These simple inexpensive designs accurately measure the RPM of a rotating wheel, which requires only a 5-volt power supply. Pulse counting is an essential application requirement in industry and R&D laboratories. This provides a cost-effective solution in industries and R&D.

REFERENCE

[1]. PSoC Product Web Site: htpp://www.cypress.com. | [2]. PSoC Training: htpp://www.cypress.com/training. | [3]. Cypress Microsystems: http://www.cypressmicro.com/ | [4].http://www.cypressmicro.com/forums PSoC cypress forums | [5]. http://www.psocdeveloper.com/forums/ for PSoC developer forums | [6]. http://www.cypressmicro.com/documentation. | [7]. http://www.ia.omron.com/sensors for optical sensor. [8]. www.opto22.com/ RPM techniques. |