THIS SPEC IS OBSOLETE

Spec No: 001-32911

Spec Title: CONSUMER/INDUSTRIAL - RATEMETER WITH A PRECISE PULSE DISCRIMINATOR FOR SPECTROMETRY - AN2231

Sunset Owner: Meenakshi Sundaram Ravindran (MSUR)

Replaced by: None

Consumer/Industrial - Ratemeter with a Precise Pulse Discriminator for Spectrometry

Author: Ilya Mamontov

Associated Project: Yes
Associated Part Family: CY8C27xxx
Software Version: PSoC Designer™ 5.4
Related Documents: AN2144

AN32911 describes the technique used to discriminate various pulses and measure their subsequent rate with the use of a PSoC® device. The associated project demonstrates proper configuration for high-speed pulse processing. The result is converted into voltage and can be measured by any voltmeter.

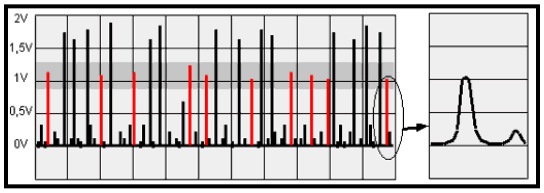
OBSOLETE

Introduction

In X-Ray Spectrometry (also in Alfa-, Beta- or Gamma-) the term “Rate” means the average number of pulses from the detector that occur over a certain period of time. The main task is to select and count the pulses having the applicable amplitudes.

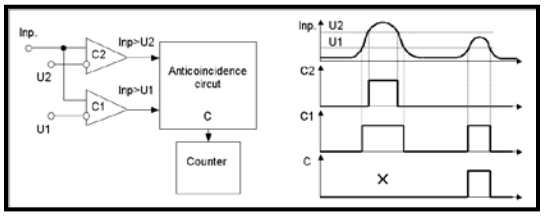
In Figure 1, the targeted region is shaded gray and the measured pulses are red.

Figure 1. Pulse Sequence from the Detector (after Preamplifier)



Traditionally, the way to discriminate the pulses is to combine two comparators and an anticoincidence circuit as shown in Figure 2.

Figure 2. Pulse Selection Diagram



Two comparators with a programmable threshold are capable of classifying the pulses into three categories:

- Within the target range
- Below level U1
- Above level U2

The task of the anticoincidence circuit is to pass the pulses from comparator C1 to the counter and remove those pulses that correspond to the pulses from C2.

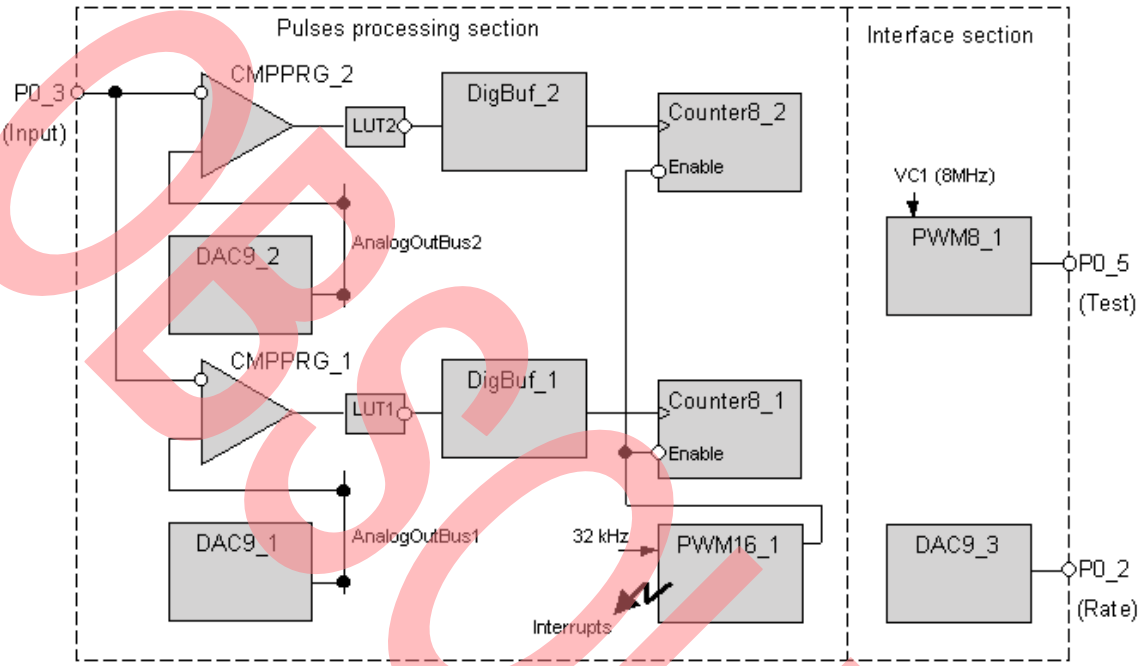
The rising and decay time of the real input pulse does not equal zero. Therefore, the initial output signals from the comparators are slightly skewed and the anticoincidence circuit is needed.

The Application Note AN2144 – Standard - “Window Discriminator” describes the pulse selection technique using the software “anticoincidence circuit.” Of course, such implementation limits the usage to slow-variation signals only. For high-speed signals (such as from a detector), it is desirable to use the hardware anticoincidence circuit. Unfortunately, the PSoC does not have this specific user module.

PSoC Implementation

The way to implement the “hardware anticoincidence circuit” is to use two separate PSoC counter user modules.

Figure 3. Block Diagram of PSoC Internals



You can determine the number of pulses within the targeted region by observing pulses below U1 and above U2. In this case, the software subtracts the two values extracted from the counters and converts the result into required output format.

The PSoC internals are shown in Figure 3.

The pulse sequence comes from P0[3] to comparators CMPPRG_1 and CMPPRG_2. To route the signal from comparators to counters, Counter8_1 and Counter8_2, the LUT blocks and DigBuf User Module are used. The DAC9_1 and DAC9_2 create the precise thresholds (lower and upper) for comparators.

Some obstacles occurred at the routing of the comparator’s input. The CMPPRG does not permit connecting the inverting input to the AnalogOutBus and the non-inverting input to the pulse source (according to the schematic in Figure 2). This is overcome by connecting the non-inverting input to the AnalogOutBus in the Device Editor and connecting the inverting input to the InputMUX by programming the control registers. See the following code:

Code 1. Inverting InputMUX

```
or reg[CMPPRG_1_COMP_CR1], 0b00111000
or reg[CMPPRG_2_COMP_CR1], 0b00111000
```

Note The operation of the comparator’s algorithm will be changed to its opposite. To achieve normal operation, the comparator output is inverted. The LUT with “~A” function does this, but this additional procedure is unnecessary for proper operation.

In addition, all of the comparator and the ComparatorBus latches must be made transparent using the following code:

Code 2. Making Latches Transparent

```
;do transparent the latches
;of ComparatorBus 1 and 2
or reg[CMP_CR1],0b01100000
;do transparent the comparator1 output
and reg[CMPPRG_1_COMP_CR2],~0x40
;do transparent the comparator2 output
and reg[CMPPRG_2_COMP_CR2],~0x40
```

Eight-bit counters are selected to reduce the number of occupied blocks. The height digit capacity (16 bits and more) is achieved by program expansion. For this application, the overflow interrupts are enabled and initiate user code, which increments the software counters.

OBSOLETE

Counter8_1 and Counter8_2 are enabled by only one source to prevent successive starting and stopping. The enable signal is created by PWM16_1, which controls the counters via the chain of BC1 and BC2 buses. The compare false period defines the time of measurement.

The compare true period when the counters are stopped is used for:

- Extracting data
- Preparing counters
- Processing

This procedure is initiated by the compare true interrupt.

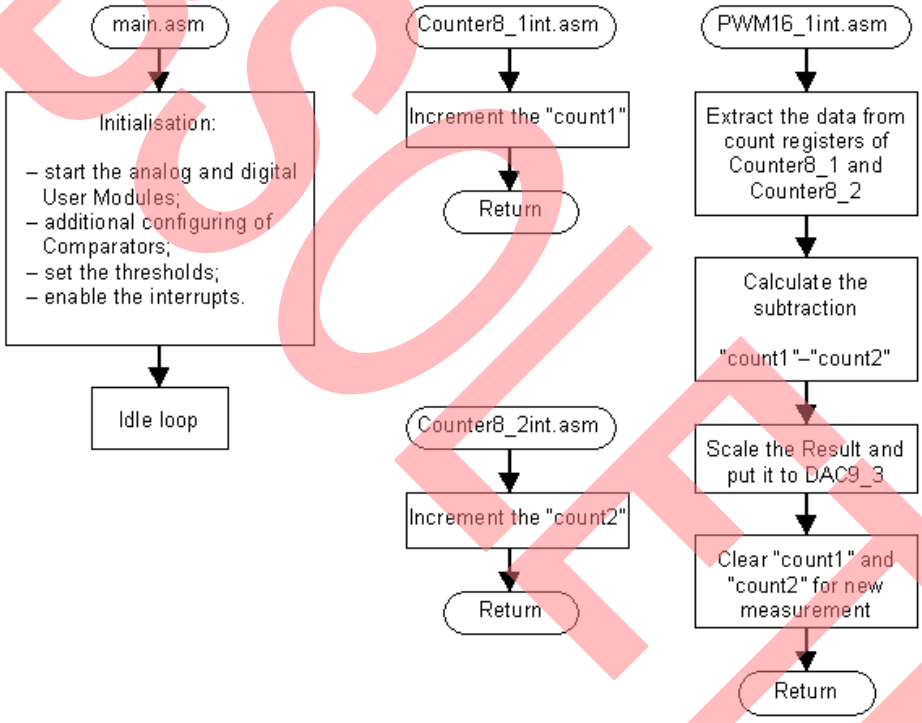
The DAC9_3 converts the current result to voltage. The output voltage is proportional to the input pulse rate and updated by the PWM16_1 ISR.

The PWM8_1 generates the pulse sequence for testing.

Both PWM8_1 and DAC9_3 user modules are optional and only implemented in this project to demonstrate the operating capabilities.

OBSOLETE

Figure 4. Program Flowchart



Program Flowchart

The flowchart shown in Figure 4 is very simple. The main program first sets up the analog and digital blocks, enables the interrupts and then is looped. After this, three threads operate simultaneously.

The first and second threads are the interrupts from Counter8_1 and Counter8_2. They only increment the 4 bytes of 5-byte variables, “count1” and “count2.” The fifth (least significant) byte is extracted from the counter registers in the third thread.

The third thread is produced by the PWM16_1 interrupts. Because all of the counters at this time are disabled, it is possible to safely manipulate the variables for “count1,” “count2” and counter registers.

This thread extracts the values from the counter registers, calculates the difference, scales it to a 9-bit range, places the result in DAC9_3 and clears all counters for the next measurement.

Main Characteristics

Following are characteristics of this project:

1. Supply voltage 5.0 V (defines the thresholds)
2. Amplitudes of input pulses 0…5 V
3. Level of zero line 0 V
4. Maximum pulse rate 400 kpps at minimum pulse width 1.2 µS (limited by internal comparators)
5. Threshold range 0…5 V with 511 steps. This software version sets the lower and upper thresholds to approximately 3 V and 4 V
6. Measurement time 0.1 s, output update rate is 9.7 samples per second
7. Analog output parameters for results show:
   - ❑ Voltage range 0…5 V with 511 steps
   - ❑ Linear relation between the output voltage and pulse rate is implemented
   - ❑ Conversion at 0.0125 V/kpps*. The 5-V output voltage corresponds to the 400 kpps input rate.
8. Test pulse generator:
   - ❑ Amplitude 5 V
   - ❑ Frequency 200 kHz
   - ❑ Pulse width 1.25 µs

Testing

The circuit is created as shown in Figure 5.

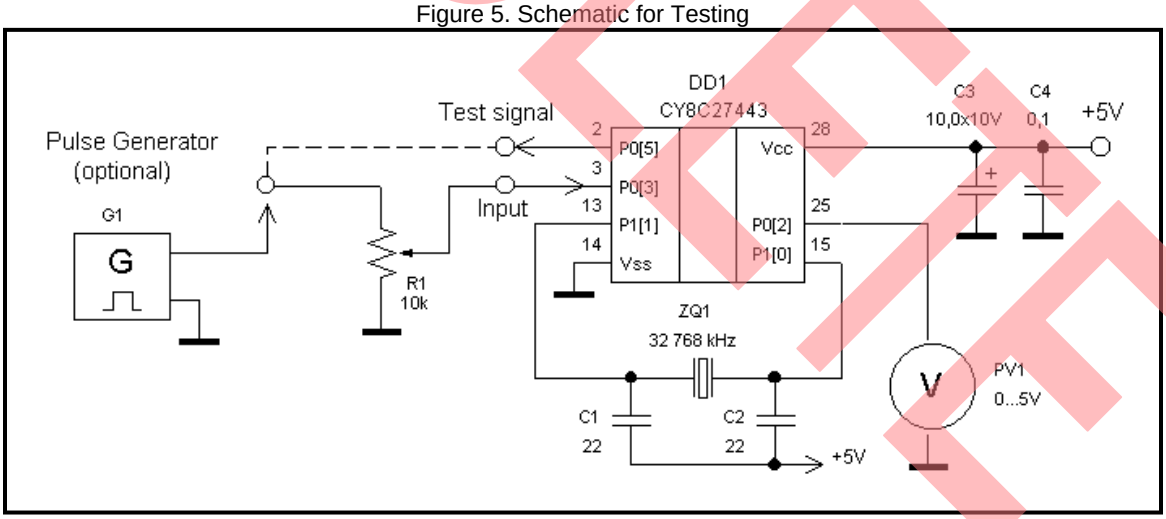
If you are using a standalone pulse generator, set the output level to approximately 5 V or use the internal regulator, R1, instead. Also, set the frequency to 200 kHz and the pulse width to 1.2 – 2 µs.

Begin testing with small amplitude (the potentiometer R1 is set to near ground). Observe the value on PV1. When the input pulse's amplitude does not exceed 3 volts, the result is zero. Adjust R1 to increase the amplitude above 3 V. The PV1 displays 2.5 V (2.5 V is half of 5 V as 200 kHz is half of 400 kHz).

If the amplitude is above 4 V, the result will be zero again. This means that the device selects the pulses from within a 3- to 4-volt range.

The standalone generator enables variation of pulse frequency. Now, if the amplitude is set to approximately 3.5 V, the output voltage ranges from 0 to 5 V while the frequency ranges from 0 to 400 kHz. By doing this, we simulate a different energy level to the detector.

Figure 5. Schematic for Testing



OBSOLETE

Summary

Using a single PSoC device enables designers to build a precise pulse discriminator in 0 to 400 kpps range and include functionality of the ratemeter.

The disadvantages are:

- Sensitivity to signal quality (noise and shape)
- False triggers (superfluous counting) if the pulse amplitude equals the threshold

The PWM8_1, PWM16_1 and DAC9_3 are used only to easily demonstrate PSoC capabilities and can be replaced by other resources. The free resources in PSoC permit the addition of a second channel or half of a channel. In the last case, the number of counter chains is three, the number of thresholds is three, too, and the two adjacent regions can be used for automatic tuning to the target pulses (if the amplitudes have a Gaussian random dispersion around the average value).

The advantages are:

- Use of internal reference sources (in this version, the supply voltage +5 V is used as a reference).
- The logarithmic or other relation between the output voltage and pulse rate may be implemented.
- Use of a more advanced postprocessor method such as a recalibrated result at maximum pulse rate that can be overlapped and the two counted as one pulse.

About the Author

Name: Ilya Mamontov
Title: Electronic Engineer
Background: Designing, modernization, repair and servicing of measurement devices.
Contact: ilka@elsite.ru

OBSOLETE

Document History

Document Title: Consumer/Industrial - Ratemeter with a Precise Pulse Discriminator for Spectrometry - AN2231

Document Number: 001-32911

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1494923	QVS	11/02/2007	Recataloged application note
*A	3250444	QVS	05/06/2011	Updated to PSoC Designer™ 5.1 Updated Template
*B	4387004	MSUR	05/22/2014	Updated Template
*C	4650317	ASRI	02/04/2015	Updated the project to PSoC Designer™ 5.4 Updated template
*D	4833879	DIMA	07/12/2015	Obsolete document. Completing Sunset Review.

OBSOLETE

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

OBSOLETE

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2007-2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.