SILEGO
TECHNOLOGY

AN-1158 Rotary Encoder to Analog Converter

In the past, Direct Current (DC) motor speed sensing was done by using a tachometer which converts speed directly to DC voltage. An analog Proportional-Integral-Derivative (PID) controller took this voltage and compared it with a reference voltage to generate a command signal to speed up or slow down the DC motor. But more recently, the method for speed feedback employs optical encoders which generate pulses instead of analog voltages.

With this, microcontrollers are used instead of analog PID controllers. A microcontroller receives the input pulses, transforms them into digital feedback signals, compares them with a digital reference and feeds it a digital control algorithm to create a motor speed command. All this adds additional hardware and requires software for motor speed control. The GreenPAK™ implementation shown is a Pulse–to–Voltage converter and can save this hardware and software cost, and an analog PID can still be used with DC motors having encoder feedback.

Decoding Incremental Rotary Encoder

The output of a rotary encoder is in the form of two pulse signals which are 90° out of phase, called Quadrature outputs. The number of pulses per second is directly proportional to the speed of the motor.

These signals are decoded to produce a count up pulse or a count down pulse. For decoding in software, the A & B outputs are read by software counters and counts per second are proportional to revolutions per second. These signals are also used to detect direction of rotation.

This requires an additional controller which receives these two pulse signals and processes them to extract speed and direction information.

In contrast, the GreenPAK CMIC will work as this controller by receiving the two pulse signals and producing an analog signal which will be directly proportional to the speed of motor.

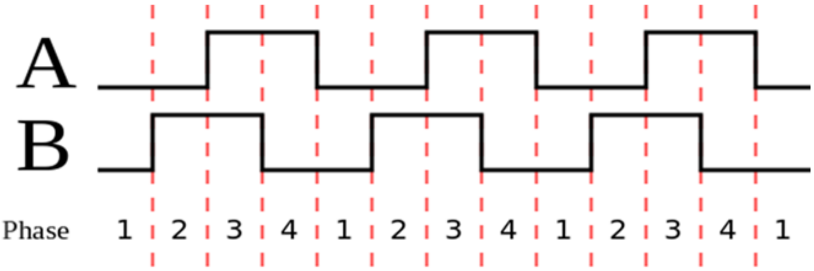


Figure 1. Rotary encoder output signals

It will also output two digital direction signals, one of which will be high when the direction is clockwise and other when the direction is counterclockwise. This is depicted as a block diagram in figure 2.

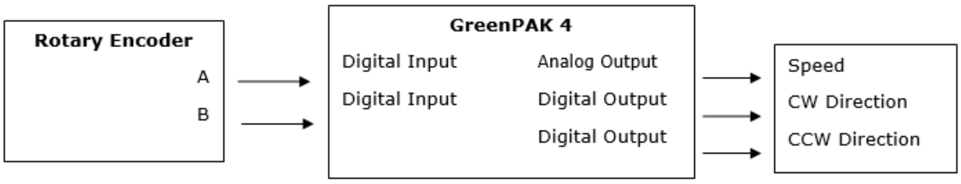


Figure 2. Block diagram of GreenPAK implementation

These speed and direction signals can be used by an analog circuit to control the speed of the motor. Thus, the GreenPAK eliminates the necessity of a separate controller and its software. Additionally, the circuit size and power requirements are also reduced.

Realization with GreenPAK

The basic idea here is to count incoming pulses for a finite and predefined duration and generate a DC voltage proportional to the count value.

The count value needs to be held until the next sample is taken, just like a sample and hold does for Analog–to–digital converters. CNT3/ DLY3/ FSM1 is used to count incoming pulses. Pulse A will be used for speed measurement and pulse B will be used for direction detection only.

Pulse A is connected to Pin 2 of SLG46140V, which serves as digital input. However, it cannot be directly connected to the clock input of CNT3/ DLY3/ FSM1 as it only allows clock sources from the Oscillator block. To work around this, Pin 2 is internally connected to the EXT. CLK0 input of the Oscillator block (OSC) and then the EXT. CLK0 output of the OSC block (labeled as Pulse) is connected to the clock input of CNT3/ DLY3/ FSM1 as shown in figure 3.

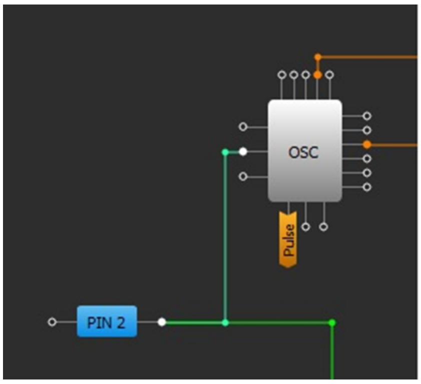


Figure 3. Pulse A connected to Ext Clk 0

Up input of CNT3/ DLY3/ FSM1 is connected to VDD so a low–to–high transition on Pin 2 causes count value of CNT3/ DLY3/ FSM1 to increase by one. CNT3/ DLY3/ FSM1 is an 8–bit counter so its value cannot reach more than 255. It needs a reset or refresh pulse at (Reset In) input to re–initialize it to zero.

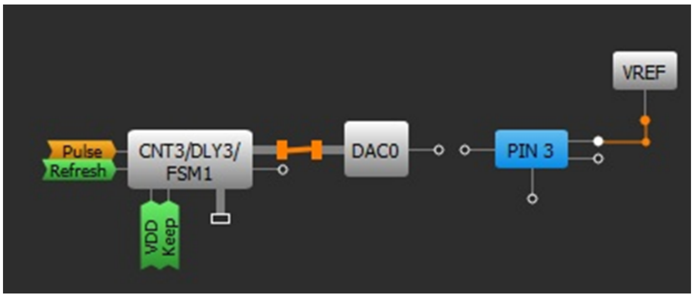
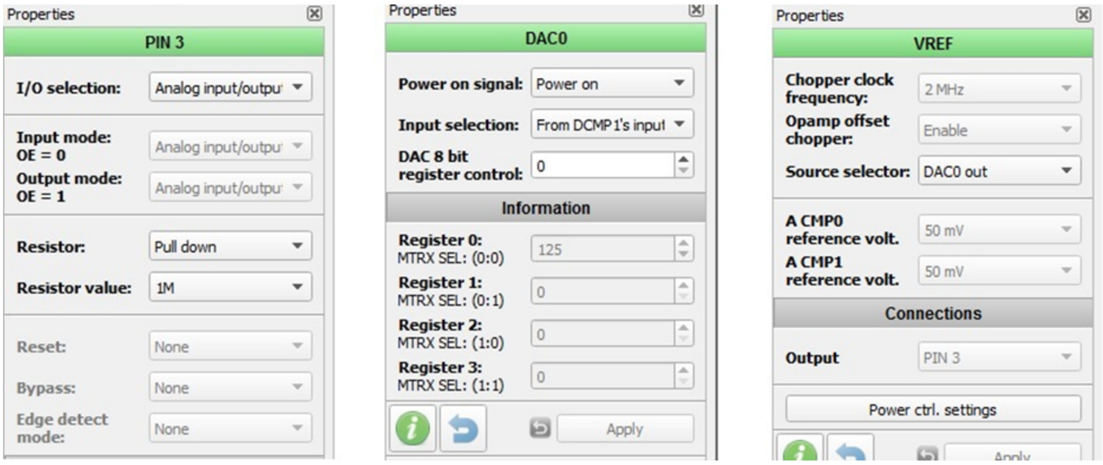


Figure 4. Pulse A connected to Ext Clk 0

The 8-bit output of CNT3/ DLY3/ FSM1 needs to be connected to the input of DAC0. CNT3/ DLY3/ FSM1's output cannot be connected to DAC0's input directly, but it can be connected to Digital comparator DCMP1's input; in this way whatever is connected to DCMP1's input will also be connected to DAC0's input.



Properties

PIN 3

I/O selection: Analog input/output

Input mode: OE = 0: Analog input/output

Output mode: OE = 1: Analog input/output

Resistor: Pull down

Resistor value: 1M

Reset: None

Bypass: None

Edge detect mode: None

Properties

DAC0

Power on signal: Power on

Input selection: From DCMP1's input

DAC 8 bit register control: 0

Information

Register 0: MTRX SEL: (0:0) 125

Register 1: MTRX SEL: (0:1) 0

Register 2: MTRX SEL: (1:0) 0

Register 3: MTRX SEL: (1:1) 0

Properties

VREF

Chopper clock frequency: 2 MHz

Opamp offset chopper: Enable

Source selector: DAC0 out

A CMP0 reference volt.: 50 mV

A CMP1 reference volt.: 50 mV

Connections

Output: PIN 3

Figure 5. Pin 3, DAC0 and VREF settings

Because of this, DCMP1 is also used in this design but is not in active use.

To get the DAC0 output to an external pin, it first needs to be tied to VREF. To achieve this, VREF is enabled and it source selector is switched to DAC0 output. Then pin 3 is selected as "Analog input/output". After this selection, pin 3 is automatically tied to VREF. This way DAC0's output is connected to pin 3 through VREF.

The output of DAC0 ranges from 0.00 volts corresponding to 0 input value (IN) to ~1 volts corresponding to 255 input value (IN).

The most important thing to consider in this design is the count up time. The count up time must be carefully chosen to allow only enough pulses to count up to 255. If more pulses are allowed after the count has reached 255, it will have no effect as count will not increase more than 255; it will stay at 255 (max out condition).

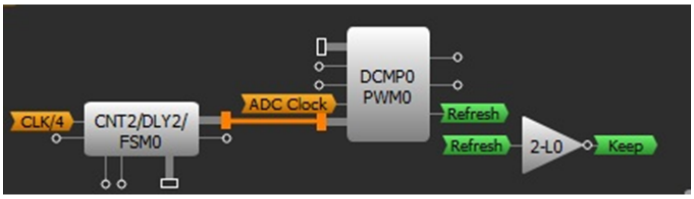


Figure 6. DCMP0 / PWM0 working as sample & hold

For example, if the incoming pulses have a maximum frequency of 5000Hz which means each pulse is 0.2ms long and it will take 51ms for incoming pulses to take count from 0 to 255, so the count up time should be set to approximately 51ms. In this situation the maximum output voltage of 1v will correspond to 5000Hz. If the count up time is set to less than 51ms it will increase the range of incoming pulses as it will not be able to max out to 255 with 5000Hz pulses. This will cause scaling change; for example if we set count up time to 40ms, the 5000Hz signal will only count to 200, resulting in 0.78V at output pin 3. With 40ms, 6375Hz will be the maximum frequency of input pulses which will correspond to 1V output. With decreasing count up time range/ scale of incoming pulse frequency increases while resolution decreases. This calculation is listed in table 1 for an easy implementation guide.

Maximum frequency	Time period (mSec)	Time to reach 255 (m Sec)
4000	0.250	0.250*255=63.75
5000	0.200	0.200*255=51.00
6000	0.167	0.167*255=42.50

Table 1. Reference value for count up time

The next important thing to consider is the sampling rate or refresh rate. In this design, incoming pulses are sampled (allowed to count up) with a predefined frequency.

The time period of this frequency must be greater than the count up time selected in the previous step. DCMP 0/ PWM 0 generates this frequency with the help of CNT 2/ DLY 2/ FSM 0. CNT 2/ DLY 2/ FSM 0 is used in counter mode and is clocked with RS OSC / 4 clock with counter reset data equal 249. It counts from 249 to 0 (250 values) in 40ms. The 8-bit output of CNT 2/ DLY 2/ FSM 0 is connected to IN- of DCMP 0/ PWM 0 which is used in PWM mode. IN+ of DCMP 0/ PWM 0 is set through register–0 to value 125. In this way DCMP 0/ PWM 0 generates a pulse train with a 40ms time period and 20ms On/Off time. This 20 mSec On time is basically count up time and 20 mSec off time is hold time. Count up/ hold time can be changed through the register 0 value.

Output (Out+) of DCMP 0/ PWM 0 is connected to RESET IN input of CNT3/ DLY 3/ FSM 1. Out+ is inverted through inverter 2-L0 and connected to the KEEP input of CNT3/ DLY 3/ FSM 1. Every 40ms a low–to–high transition on Out+ causes reset of CNT3/ DLY 3/ FSM 1. Then for 20ms (count up time) counts are allowed to increase with incoming pulses as KEEP input remains low. After 20 mSec Out+ becomes low for 20ms making KEEP input high and count is hold during this time period.

Direction detection

Direction detection is implemented in this simple manner.

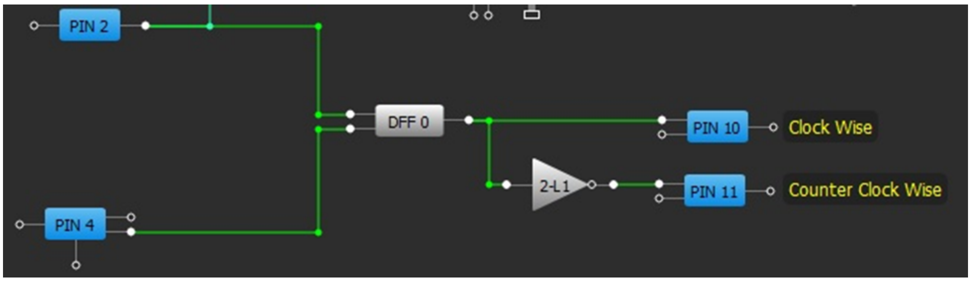


Figure 7. Direction detection circuit

Pulse A from pin 2 is connected to the D input of DFF0 while Pulse B is connected to the clock (CK) input of DFF0.

When the motor is rotating clockwise, pulse A will be leading pulse B. The rising edge of pulse B (CK input) will come when pulse A (D input) is high, output Q of DFF0 will remain high. When the motor is rotating counterclockwise, pulse B will be leading pulse A. Rising edge of pulse B (CK input) will come when pulse A (D input) is low, output Q of DFF0 will remain low. Q is connected to pin 10 and inverter L1. Inverted Q is connected to pin 11.

So pin 10 will be high when the direction is clockwise and pin 11 will be high when the direction is counterclockwise. Both pin 10 and pin 11 are configured as digital outputs.

Resources utilized

Table 2 shows the resources used in this design:

S #	Resource	Function
1.	Pin 2	Pulse A input
2.	Pin 3	Speed – analog voltage output
3.	Pin 4	Pulse B input
4.	Pin 10	Clock wise indication
5.	Pin 11	Counter clock wise indication
6.	CNT 2/ DLY2 /FSM0	Cyclic down counter
7.	CNT 3/ DLY3 /FSM1	Input pulse counter
8.	DCMP 0/ PWM 0	Count up pulse generator
9.	DCMP 1/ PWM 1	To facilitate use of DAC0
10.	DAC0	To generate analog voltage proportional to speed
11.	Vref	To facilitate use of Pin 3 as analog output
12.	LUT0	To generate Keep signal
13.	LUT4	Used as D flip flop in direction detection

Table 2. Utilized Resources

Example implementation

Example implementation for this app note was done using Pittman DC motor 14202 series. It had E30 incremental optical encoder having 1000 counts per revolution. Figure 8 show connections between encoder and development kit for this implementation.

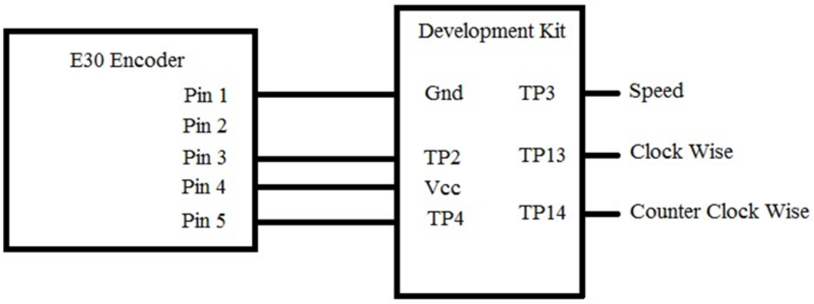


Figure 8. Connection between Optical encoder and development kit

This design is not limited to decoding a motor encoder. It can be used in any application requiring pulse to voltage conversion. The only things to consider will be the scale of frequency and the adjustment of count up time and hold time. Consequently, refresh rate will also change. Pulse signal is connected through external clock input which has certain frequency limitation needs to be considered.

This can also be used in flow regulating systems. Turbine type flow sensors generate pulses proportional to flow. In flow control application, a pump connected with VFD (variable frequency drive) is used to regulate flow. Frequency of VFD is varied based on flow–feedback voltage. A circuit based on this GreenPak implementation will take input from flow sensors and give feedback to VFD to regulate flow.

With some additional hardware it can be used to measure frequency of AC signals. To do this each zero crossing will be converted to a pulse signal and then zero crossing will be counted to determine the frequency of AC signal. Refer to App note AN–1124: "AC Lightbulb Dimmer with a GreenPAK and TRIAC" for how to convert zero crossings into pulses.

Some other applications may include: anemometer used for measurement of wind speed, odometer used for measurement of distance travel and depth meter used by drilling machines.

Analog output in GreenPak devices is limited to 1 volt so some applications might require amplification. This can be implemented using OpAmps.

Conclusion

This app note demonstrates how to implement a simple pulse to voltage converter with minimum external components. Only a few internal blocks of SLG46140V are used, leaving the bulk of the blocks available to build other circuitry around it. This is an ideal example of a mixed signal IC as both analog and digital blocks are used in this application.

About the Author

Name: Faizan Hasan

Background: Faizan received B.E degree in Industrial Electronics from NED University Karachi. He also has M.S degree in Control Systems. He has more than five years of experience in designing electronics and automation systems. Currently he is working as senior lead engineer in a privately owned research & development company.

Contact: appnotes@silego.com

Files

- AN-1158 Rotary Encoder to Analog Converter.gp4- (20 KB)
- AN-1158 Rotary Encoder to Analog Converter.pdf- (370 KB)
- AN-1158.zip- (280 KB)

See full list of Application Notes