

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/324971223>

Introduction to Industrial Communication Networks, Integration Systems and Control (PCWorx Express) – Apostilled material of the practical class – 003_B: PWM Fan Speed Control

Technical Report · May 2018

CITATIONS

0

2 authors:



[Hermes Jose Loschi](#)

University of Campinas

53 PUBLICATIONS 37 CITATIONS

[SEE PROFILE](#)



[Yuzo Iano](#)

University of Campinas

122 PUBLICATIONS 204 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Compact Rectennas [View project](#)



Pattern Recognition for Partial Discharge Analysis in Instrument Transformers [View project](#)

PWM fan speed control

1. Pulse Width Modulation (PWM) principle.

Pulse-width modulation (PWM) is a technique used to control the mean voltage value of load by change the width of square waves. Its main use is to allow the control of the power supplied to electrical devices, especially to inertial loads such as motors [1].

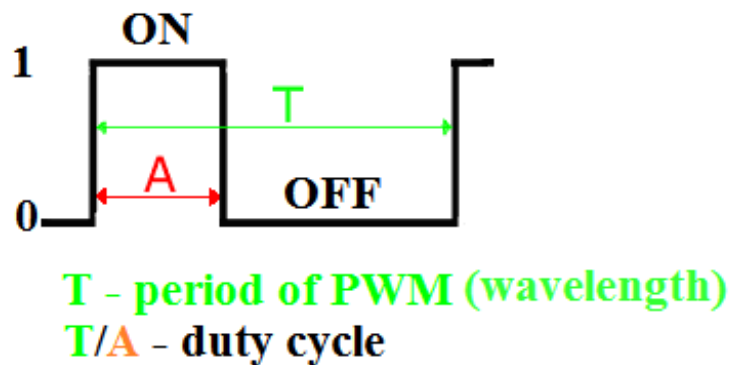


Fig.1 - Duty cycle of PWM

The duty cycle is the amount of time (t) that the *period* is in a logic 1 state, divided by the period (T) of the wave. The period (T) is the duration of *one cycle*, also known as *wavelength*.

The duty cycle therefore describes a fraction of the cycle for which the square wave is in the logic 1 state. When you multiply this value by 100, you get the duty cycle value in terms of a percentage.

$$\text{Duty Cycle} = (t / T) \times 100$$

For example, if the period (T) of the signal was 2 seconds, and the duration of the logic 1 state (t) was 1 second, then the following expression gives the duty cycle in percentage [2].

$$(1 / 2) \times 100 = 50 \%$$

On Fig.2 you can see three PWM signals with different duty cycles (10%, 50%, 90%). For example, if you have 24 V power supply you get 2.4 V, 12 V and 21.6 V on your motor respectively.

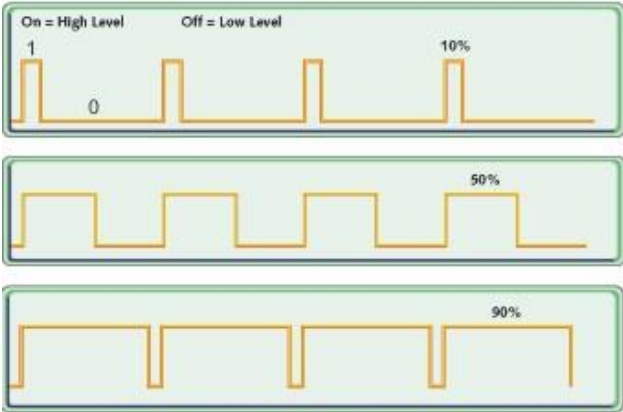


Fig.2 - Three PWM signals with different duty cycles

Therefore, the PWM technique is used for imitation the analog signal by digital output of controller.

2. Lab equipment

According to Fig.3 we connect the 24 V fan to the first digital output of ILC 130 ETH directly (We will not use the RS-232 connection).

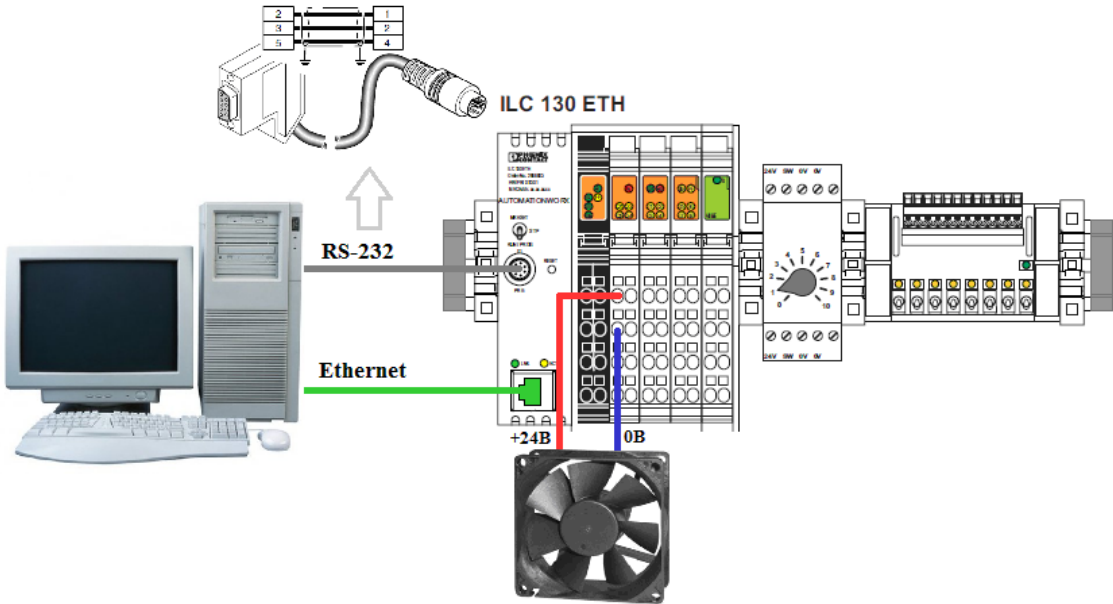


Fig.3 - The lab equipment connection

The nominal current of the fan is 0.2 Amps. If you have more powerful fan you should use intermediate control module (transistor or another).

The PWM switching frequency has to be much higher than what would affect the load (the device that uses the power), which is to say that the resultant waveform perceived by the load must be as smooth as possible [2].

In our case, the PWM switching frequency is limited by cycle period of controller, which equals or more than 1 ms. Practically the frequency 100 Hz is enough for quite smooth control of fan speed. This frequency corresponds to 10 ms PWM period. If the PLC cycle is 1 ms we get 10 gradations of fan speed.

Note: for higher PWM frequency you should use the special PWM module from Phoenix Contact like IB IL PWM/2-PAC [3].

The task: to create a program for control the fan speed smoothly by potentiometer on ILC 131 ETH Starter Kit;

3. The PWM user function block on ST and FBD.

The simple function block for PWM is shown on the Fig.4:

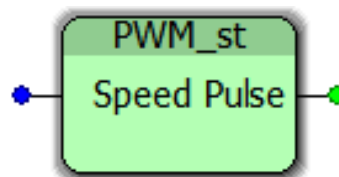


Fig. 4 - User PWM function block

Speed - speed of fan (integer value from 0 to 9); Pulse - logical output, the series of pulses (BOOL type).

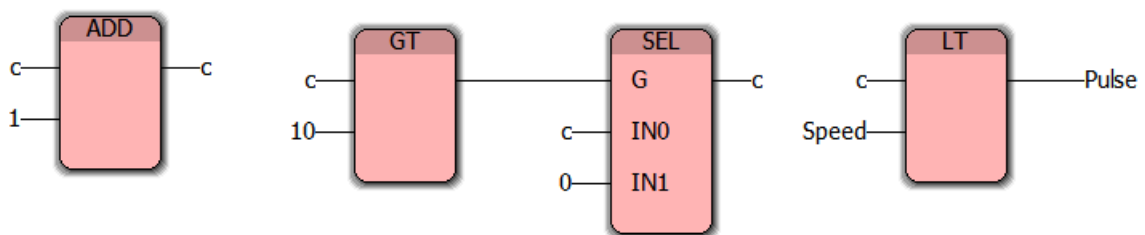
The program code on ST:

```

1   c:=c+1;
2   if c=10 then c:=0; end_if;
3   if c<Speed then Pulse:=true;
4       else Pulse:=false;
5   end_if;

```

The same algorithm on FBD (Fig. 5):



The variables table:

	Name	Type	Usage	Description
☐ Default				
	Speed	INT	VAR_INP...	input
	Pulse	BOOL	VAR_OU...	output
	c	INT	VAR	counter

Note: if you insert this block in program (for example Main program) the PWM frequency is equals to call period of the program. For default task (Main program) call period will depend on the quantity of another tasks.

5. The Main program

The main program is shown below on Fig.6. You can see the channels for fan speed control from potentiometer on the board.

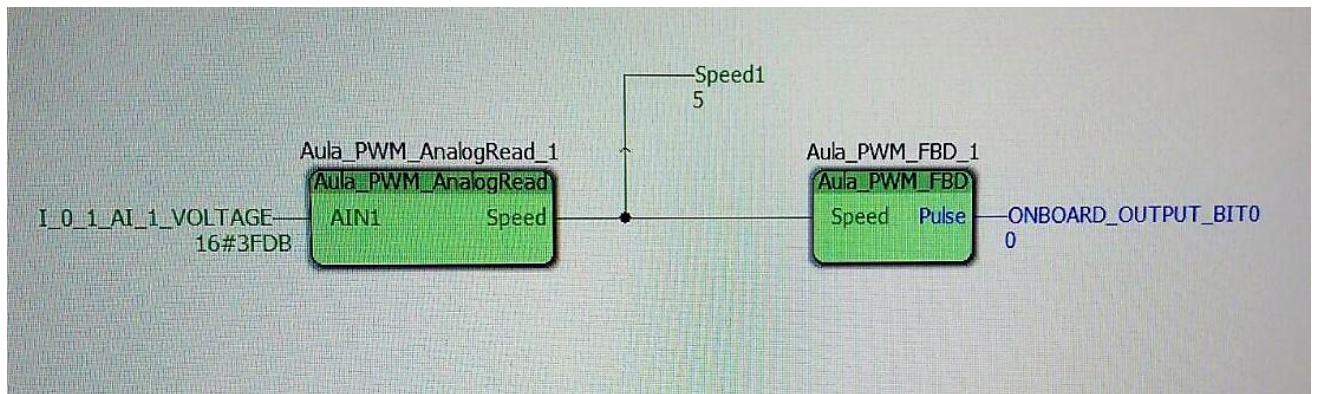


Fig. 6 - The Main program

The variable **ONBOARD_INPUT_BIT0** selects the on/off for speed control. It is provided by **AND** functions (Fig.6).

User Function Block **AnalogRead** reads the voltage from potentiometer and scales it to the range [0..9]. The implementation of the block on FBD:

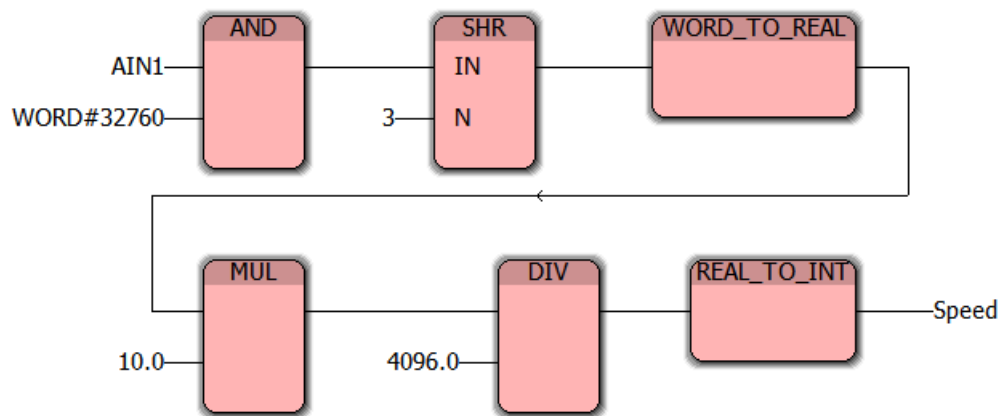


Fig.7 - Function block AnalogRead

The constant **WORD#32760** is a mask. It is used to clear all unused bits and get the value from the IB IL AI2/SF-ME format [5].

6. Test of the whole program

To test the whole program you need to connect ILC 131 Starter Kit to PC as shown in Fig. 3. Also you should connect the 24V fan to first onboard digital output.

After downloading the program into PLC try to rotate the potentiometer. Speed of the fan must change smoothly (if the analogue variable does not exist, create the same).

7. References:

- 1) Pulse-width_modulation. Wikipedia. Available: http://en.wikipedia.org/wiki/Pulse-width_modulation
- 2) PWM: Pulse Width Modulation Principle. Available: http://www.petervis.com/Raspberry_PI/Raspberry_Pi_Motor_Controller_L6203/PWM_Modulation_Principle.html.
- 3) IB IL PWM/2 (-PAC). Inline Function Terminal for Pulse Width Modulation and Frequency Modulation. Data Sheet. Phoenix Contact GmbH & Co, Blomberg, Germany.
- 4) Installing and operating the ILC 130 ETH, ILC 150 ETH, ILC 155 ETH and, ILC 170 ETH 2TX Inline Controllers. User Manual. Phoenix Contact GmbH & Co, Blomberg, Germany.
- 5) IB IL AI 2/SF-ME . Inline Terminal With Two Analog Input Channels. Data Sheet 703700. Phoenix Contact GmbH & Co, Blomberg, Germany.