

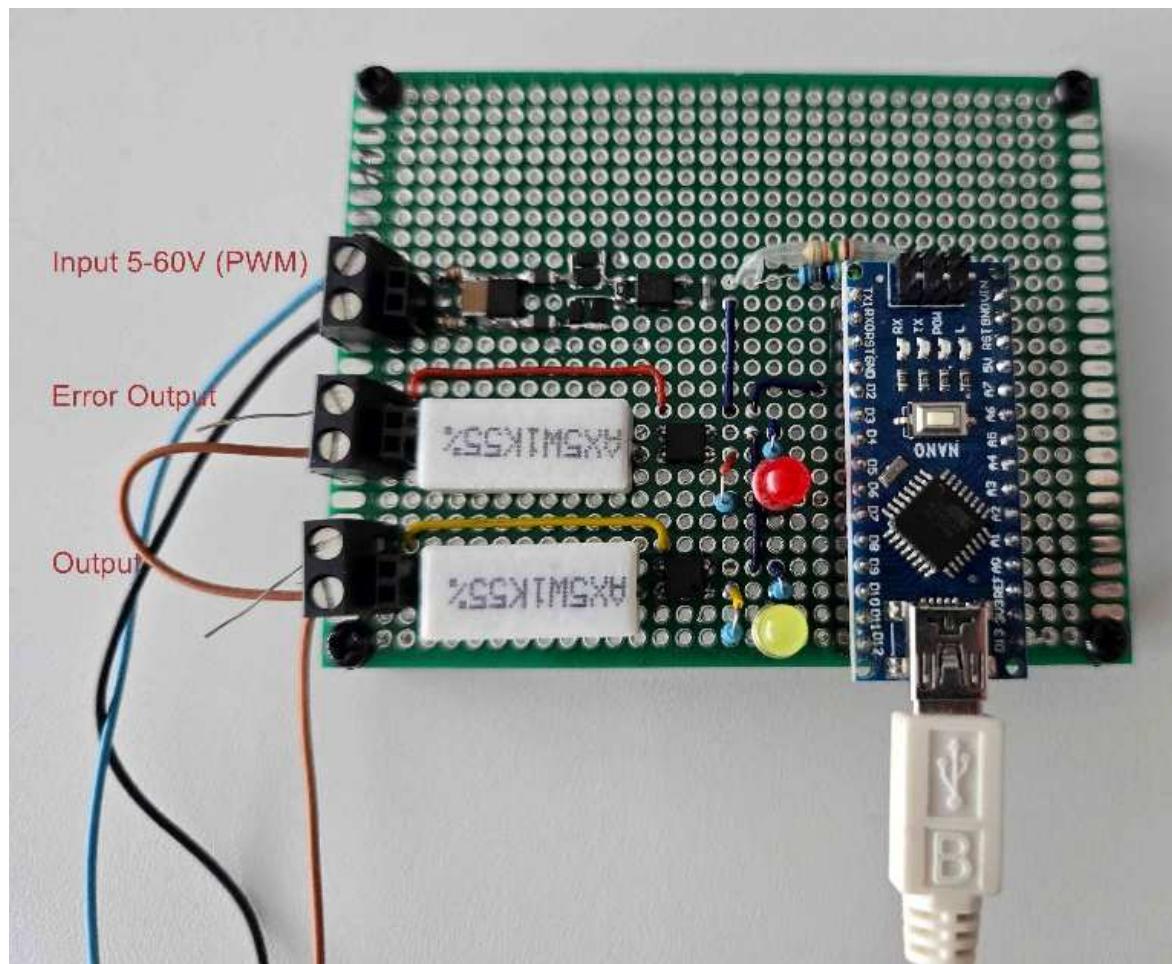
Evaluating PWM signals using Arduino NANO

This is a brief guide to the program and device for detecting and analyzing the #70197494 MB15-F33X-PWM-AMP sensor with PWM output. The device is based on the Arduino Nano board. Inputs and outputs are optically separated. The input signal must be in the range of 5-60 V / min 10mA.

The outputs are handled via opto-relay #70120618 OCPL S AQY210KSX SO4 a (350V/120mA). Current limitation is reliably handled by a 1k/5W series resistor for the range of 5-60V.

The program is primarily designed to detect a PWM signal with a frequency of 50Hz and two different duty cycles. A duty cycle of 30% corresponds to Log_0 and a duty cycle of 70% corresponds to Log_1. Settings can be called up or edited via serial line.

The delay after a short pulse is approximately 230 us, after a long pulse approximately 600 us.

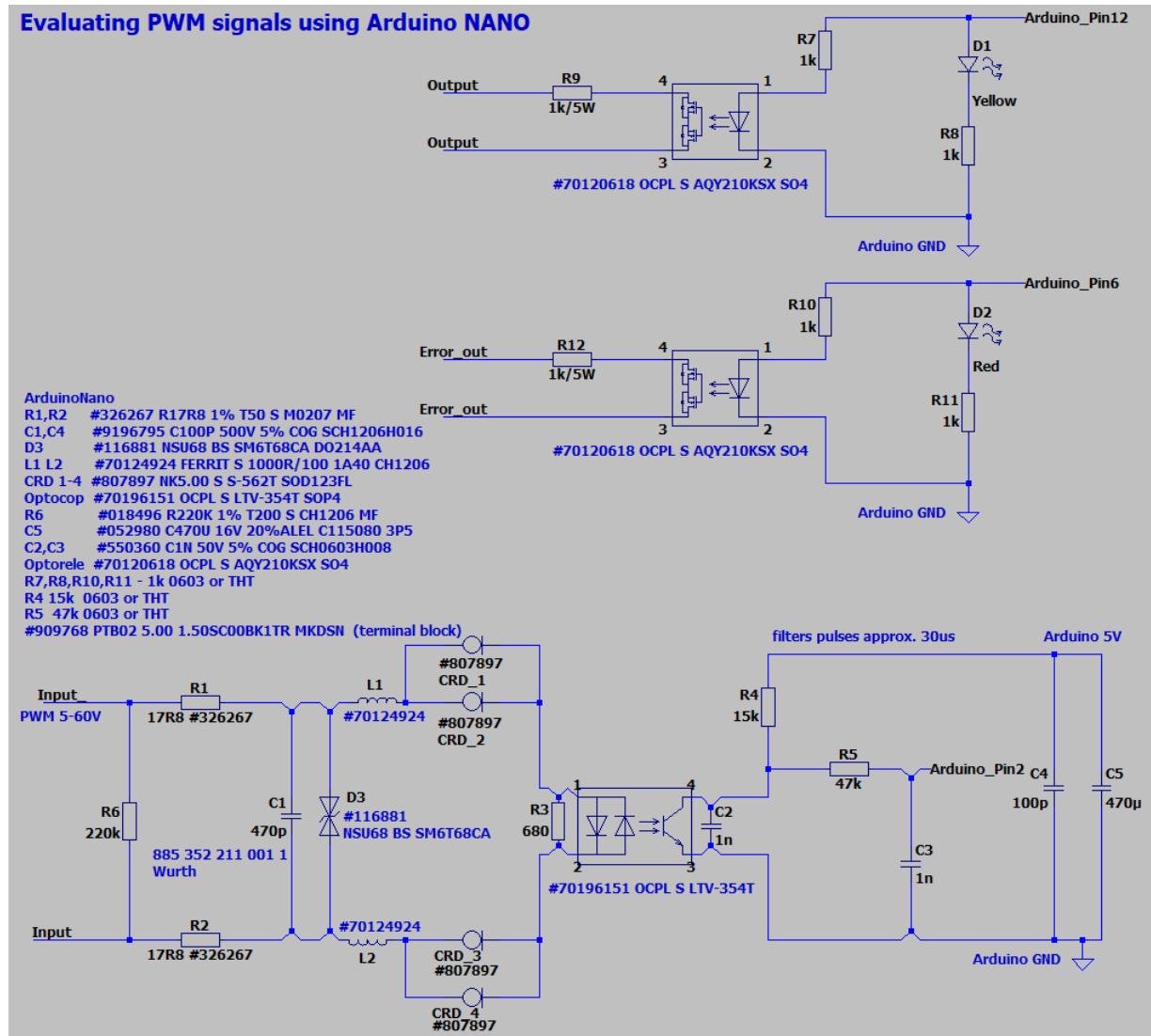


Commands: (the command must be terminated by a terminal char '\n')

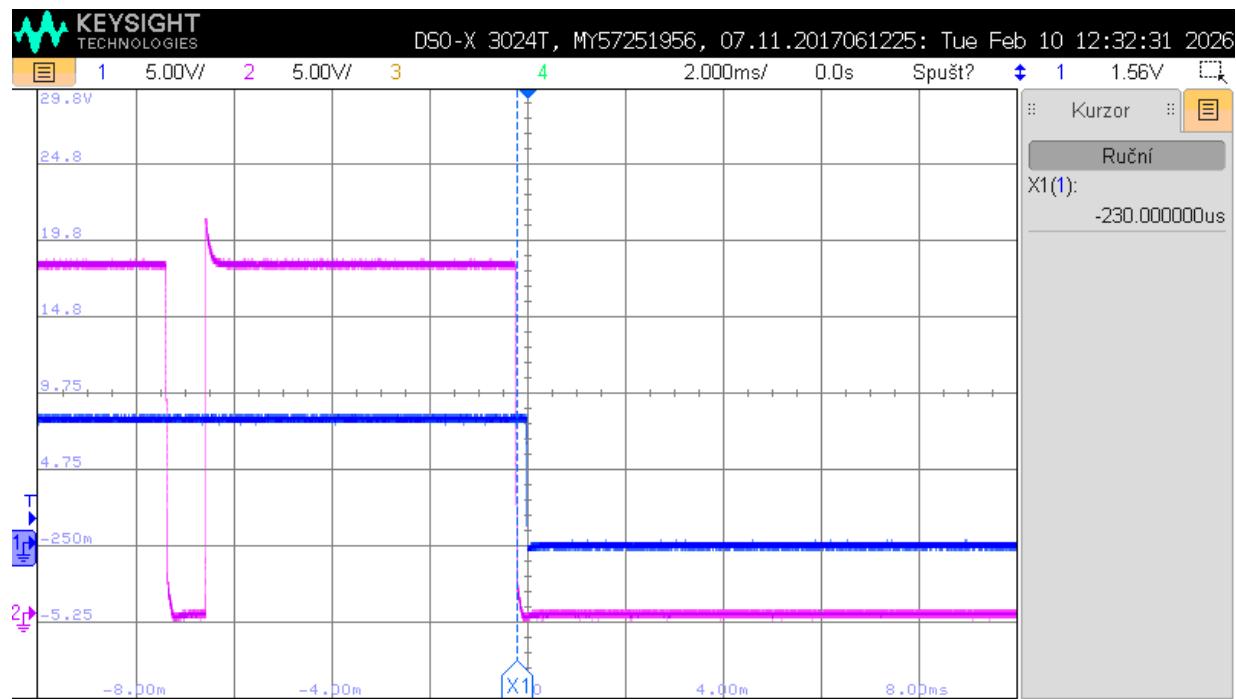
command	parameter	factory setting	description
-h or ? or HELP			prints a brief help
show			lists the current settings values
-t	[number1] [number2]	8000 12000	setting the lower and upper pulse limit of the PWM signal on pin2 in us (500-20000) for changing pin12
-i	[number]	0	inverts the logic of the evaluation output pin12 (0/1 - yes/no)
-p	[number1] [number2]	18000 22000	limits of the correct period of the PWM signal in us (100-65000)
-s	[number1] [number2]	280 330	setting the correct duty cycle of the PWM signal in per mille for log0 (it is assumed that for log1 these values are mirrored (1-499))
-e	[number]	2	setting the number of consecutive PWM period or duty cycle errors to evaluate the pin6 error output (0-255)
-te	[number]	800	minimum error signaling time in ms on pin6 (10-65000)
-b	[number]	96	serial buat rate 96 -> 9600, 144 -> 14400, 192 -> 19200, 288 -> 28800, 384 -> 38400, 576 -> 57600, 1152 -> 115200
-l	[number]	0	output of currently measured values to the serial line (0-No; 1-Yes error; 2- Yes periodic)
-ds	[char]	','	decimal separator
-cs	[char]	'.'	columns separator
*IDN?			returns the device IDN. (Arduino NANO for measuring PWM signal duty cycle. Version: 1.4)
*RST			sets all parameters to factory settings
*CLS			clear error counter
:FETCh?			returns the duty cycle values of the PWM signal in per mille
:MEASure:WIDth?			returns the length value of the HIGH signal
:MEASure:PERiod?			returns the signal period value
SYStem:ERRor:COUNt?			returns the number of errors since start or *CLS

Recommendation for detection on EMC 61000-4-6 set '-e 0\n'

Wiring diagram:



PWM duty cycle 70% to 30% (short) delay 230us, purple - input PWM signal, blue - output (yellow LED)



PWM duty cycle 30% to 70% (long) delay 600us, purple - input PWM signal, blue - output (yellow LED)

