ESC Tester V.1

Wednesday, September 21, 2022 10:15 PM

Joseph De Vico

The V.1 iteration of the ESC Tester introduces latency testing for any and all ESCs that respond to PWM control signals. In the default configuration ESC response time is tested 40 times per test cycle.

10x at 0% to 25%
10x at 0% to 50%
10x at 0% to 75%
10x at 0% to 100%

The 16 bit Timer 1 of the ATMega328P (Standalone, XplainedMini328P, Arduino Uno and Nano) is utilized in a parametric (ie. Accurate with varying clock speeds) way to ensure repeatable timing results. A several cycle minimum is maintained in order to account for additional instruction execution between starting the timer -> starting the ESC -> interrupt -> stopping timer. The tester should be accurate to a fraction of a microsecond assuming a proper 16.000MHz crystal clock source for the MCU.

Connections are very simple to set the test platform up. Internal pullups and state machine logic are utilized to reduce the need for any external components aside from a single button (V.1 only) and protection circuitry for INT1.

Internal Peripheral Utilization:

- TIMER_1: Running at 1x sys_clk counting between test cycles
- TIMER_1_OVF: Timer 1 overflow calls an interrupt where an overflow counter is incremented
- EXT_INT_0: External Interrupt 0 is a rising edge detect interrupt utilized for the start test cycle button
- EXT_INT_1: External Interrupt 1 is a change detect interrupt utilized to trigger on ESC state changes
- TIMER_0: Utilized for generating phase correct ESC test PWM signals (PD6 Output)
- USART: Standard Serial output (9600 8N1)

How it Works:

As an example only a single test cycle -> print will be described.

Once the RUN command has been indicated (button press) the button interrupt is disabled, TIMER_1 stopped and reset, TIMER_1 overflow counter reset, ESC speed set to 0% (1500us)
The motor interrupt (INT_1) is enabled, ESC speed set to test frequency, and timer started.
The core waits in a while loop until test termination is signaled by the motor interrupt trigger.
The motor interrupt trigger stops TIMER_1 and turns the ESC speed to 0%
This value is checked against the TIMER_1 overflow counter, if the overflow counter is non-zero the appropriate value (TIMER_1_VALUE + TIMER_OVF_VALUE * 2^16) is saved, else just the timer value is saved.

This repeats as long as necessary.

For printing the data every TEST_CYCLES an average is calculated from the previously printed values.
dtostrf() is utilized to convert float values to C strings, ready for printing.

Future Iterations:

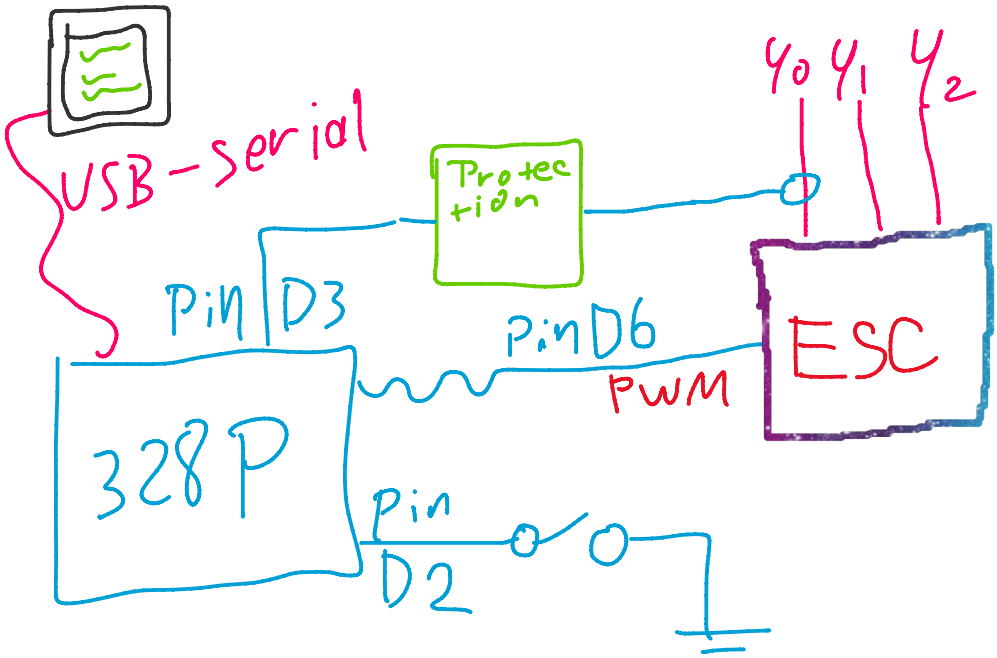
V.2 Will implement more state machine states to live control an ESC to test functionality. Possibly modes to verify slew-rate-limiting features will be implemented as well.

Live control should be possible through encoder/potentiometer input however this was a low priority.

External Resources Needed:

Avr-gcc and standard avr-gcc libraries are all that you need to compile from source successfully. Upload should be done through *avrdude* or your favorite flashing tool.

Hook Up Guide



Sample Outputs

```
Welcome to ESC Tester v.1

Run
At 25% = 200pwm val
DEC:        4308.00  cycles        0.00026925001 s
DEC:       30980.00  cycles        0.00193625000 s
DEC:       52089.00  cycles        0.00325556240 s
DEC:       22002.00  cycles        0.00137512500 s
DEC:       11005.00  cycles        0.00068781251 s
DEC:       49277.00  cycles        0.00307981250 s
DEC:           6.00  cycles        0.00000037500 s
DEC:       52949.00  cycles        0.00330931250 s
DEC:       13431.00  cycles        0.00083943747 s
DEC:           6.00  cycles        0.00000037500 s

AVG:       23605.30  cycles        0.00147533110 s

At 50% = 213pwm val
DEC:       20963.00  cycles        0.00131018750 s
DEC:       63760.00  cycles        0.00398499980 s
DEC:           6.00  cycles        0.00000037500 s
DEC:       25119.00  cycles        0.00156993740 s
DEC:       11513.00  cycles        0.00071956252 s
DEC:       19527.00  cycles        0.00122043750 s
DEC:       11727.00  cycles        0.00073293748 s
DEC:       15493.00  cycles        0.00096831250 s
DEC:       37006.00  cycles        0.00231287490 s
DEC:       20188.00  cycles        0.00126175000 s

AVG:       22530.20  cycles        0.00140813760 s
```

```
At 75% = 225pwm val
DEC:       44614.00  cycles        0.00278837490 s
DEC:       60923.00  cycles        0.00380768740 s
DEC:       22389.00  cycles        0.00139931250 s
DEC:       60757.00  cycles        0.00379731250 s
DEC:       28836.00  cycles        0.00180225000 s
DEC:       42992.00  cycles        0.00268700000 s
DEC:        3207.00  cycles        0.00020043750 s
DEC:        2703.00  cycles        0.00016893751 s
DEC:       54360.00  cycles        0.00339749990 s
DEC:       19337.00  cycles        0.00120856250 s

AVG:       34011.80  cycles        0.00212573750 s

At 100% = 238pwm val
DEC:       48360.00  cycles        0.00302250010 s
DEC:       11729.00  cycles        0.00073306251 s
DEC:       12495.00  cycles        0.00078093750 s
DEC:       45464.00  cycles        0.00284150010 s
DEC:       10779.00  cycles        0.00067368749 s
DEC:       19217.00  cycles        0.00120106250 s
DEC:       65101.00  cycles        0.00406881240 s
DEC:           6.00  cycles        0.00000037500 s
DEC:       36410.00  cycles        0.00227562500 s
DEC:       12007.00  cycles        0.00075043750 s

AVG:       26156.80  cycles        0.00163480010 s
```

```
Welcome to ESC Tester v.1

Run
At 25% = 200pwm val
DEC:     8265603.00  cycles        0.51660019000 s
DEC:    10317715.00  cycles        0.64485717000 s
DEC:           6.00  cycles        0.00000037500 s
DEC:     7669401.00  cycles        0.47933757000 s
DEC:    11778267.00  cycles        0.73614168000 s
DEC:     7771413.00  cycles        0.48571330000 s
DEC:     2660147.00  cycles        0.16625918000 s
DEC:     3583397.00  cycles        0.22396231000 s
DEC:     2203161.00  cycles        0.13769756000 s
DEC:     3523886.00  cycles        0.22024287000 s

AVG:     5777299.00  cycles        0.36108121000 s
```