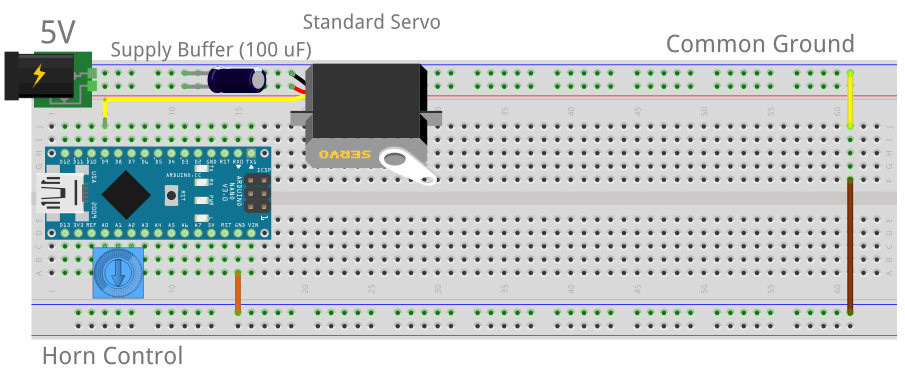
**1.0 Minimal Confirmation Prototype (Trim Pot not used)**



ATmega328 Datasheet: <http://darcy.rsgc.on.ca/ACES/Datasheets/ATMEGA328P.pdf#page=111>

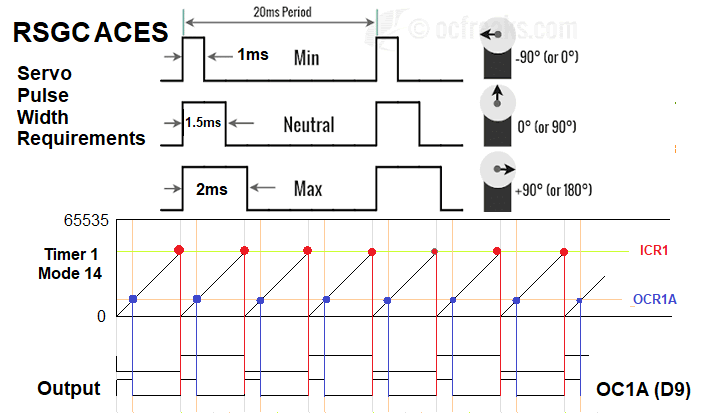
**1.1 ATmega328P Timers: 3 Options**



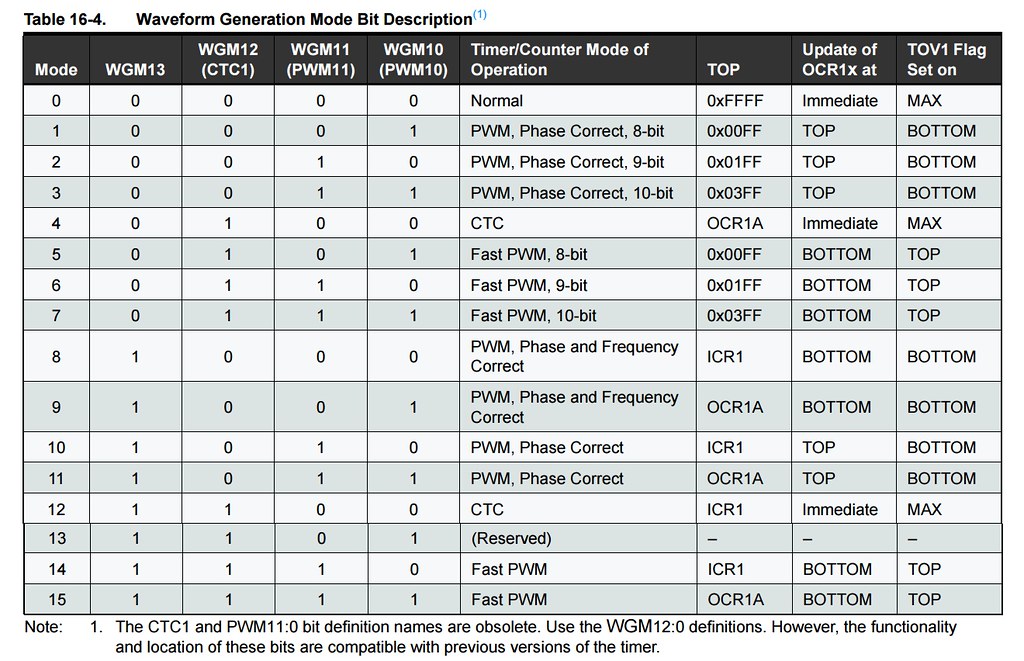
**1.2 ATmega328P Timer Presentation Pins (Timer1 Mapped to Digital Pins 9 and 10)**



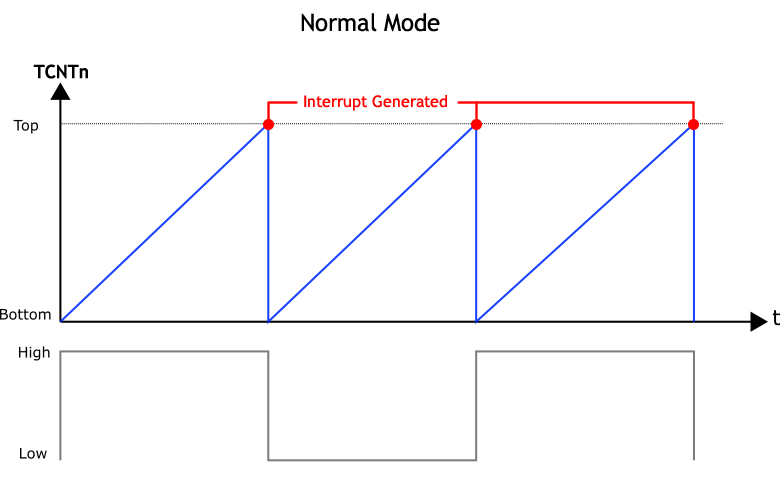
**1.3 Timer1 (Fast PWM Mode 14): Servo Waveform Generation**



**1.4 Timer1 Waveform Generation Modes**

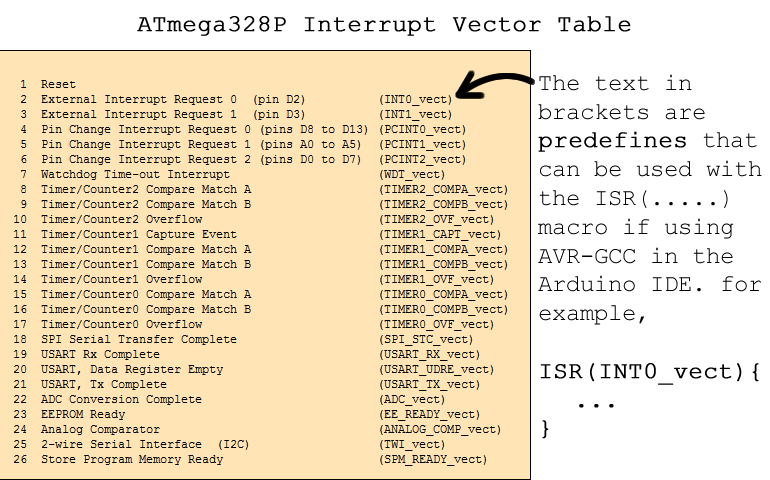


**1.5 Timer1 Mode 0 Waveform with Interrupt**



**1.5.1 Code Example: Register-Level Blink Sketch** [http:darcy.rsgc.on.ca/ACES/TEI3M/ArduinoCode/Timer1NormalMode0Blink.ino](http://darcy.rsgc.on.ca/ACES/TEI3M/ArduinoCode/Timer1NormalMode0Blink.ino)

**1.6 ATmega328P Interrupt Vector Table: Column 2 Suitable for Direct Code Use**



**1.6.1 ISR Code Template**

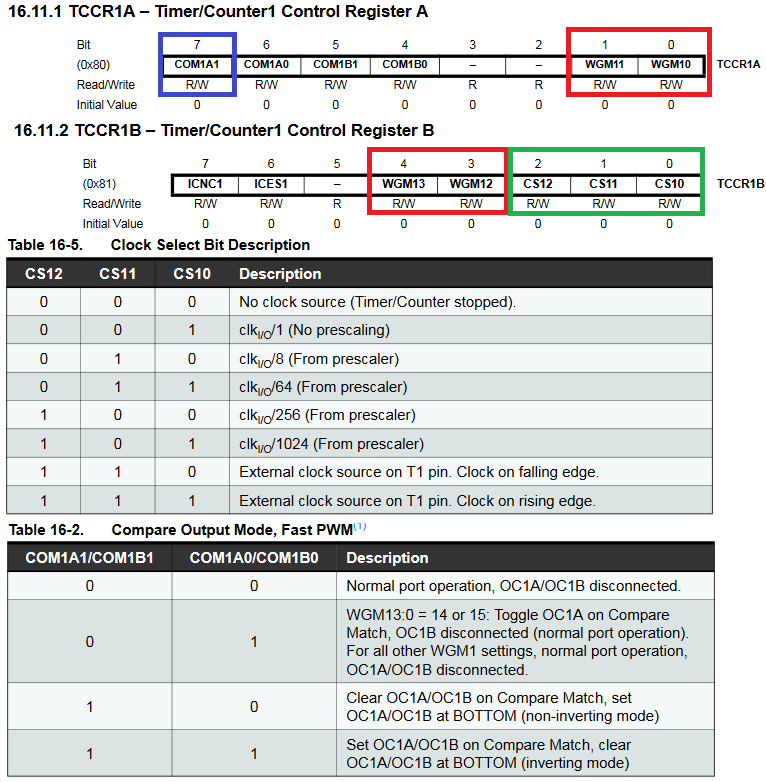
// Timer/Counter 1 Interrupt Service Routine

ISR(TIMER1\_OVF\_vect) {

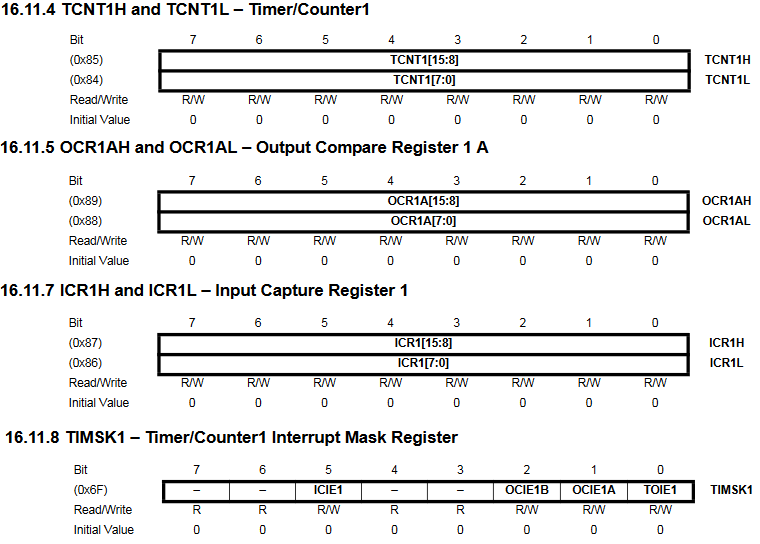
PORTB ^= (1 << PB5);

}

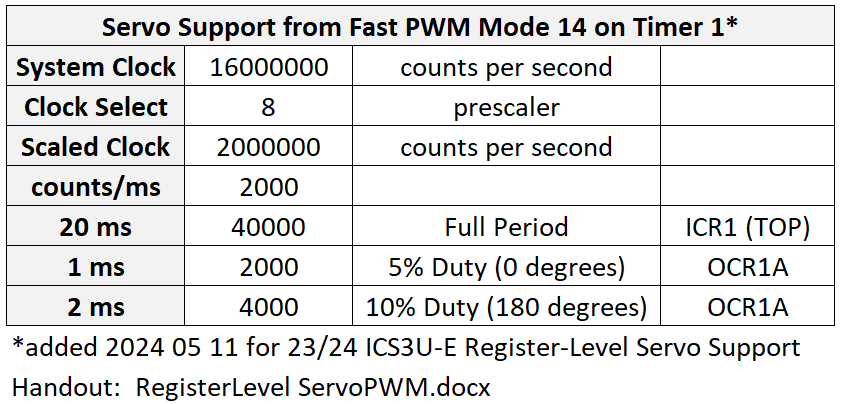
**1.7 Timer 1 Primary Control Registers and Prescalers**



**1.8 Timer 1 Secondary Registers**



**1.9 Timer1 Mode 14 Configuration Considerations**



**1.10.1 Code Example: Register-Level Blink Sketch**<http://darcy.rsgc.on.ca/ACES/TEI3M/ArduinoCode/Timer1FastPWMMode14.ino>

// PROJECT  :MyTimer1NormalMode0Blink

// PURPOSE  :Gentle introduction to AVR Timers. Sets up SERVO PWM (Mode 14)

// COURSE   :ICS3U

// AUTHOR   :C. D'Arcy

// DATE     :2024 05 11

// MCU      :328P

// STATUS   :Working

// REFERENCE:

void setup() {

  //  pinMode(LED\_BUILTIN, OUTPUT);

  // Register level...

  DDRB |= (1 << PB5);

  // Normal Mode

  TCCR1A = 0;

  // set up timer with CLKio/8 prescaler

  TCCR1B = 1<<CS12;

  // Enable Timer1 interrupt ability

  TIMSK1 = 1 << TOIE1;

  // Enable global interrupt ability...

  sei();

}

// Timer 1 Interrupt Service Routine

ISR(TIMER1\_OVF\_vect) {

  PORTB ^= (1 << PB5);

}

void loop() {}

**1.10.2 Register-Level PWM Waveform for Servo Rotation (ATmega328P Timer1 Mode 14)**

// PROJECT  :Timer1FastPWMMode14

// REFERENCE:http://darcy.rsgc.on.ca/ACES/TEI3M/RegisterLevelServoPWM.docx

// NOTES    :Frequency and duty cycle depend on Timer configuration

//          :Optimum thresholds vary with particular Servo

// Trial & Error on an FS5103B Servo from ABRA

#define LOWER   1200           // Lower bound for OCR1A: 0°

#define UPPER   4800           // Upper Bound for OCR1A: 180°

#define TOP     10\*UPPER       // Threshold for 50Hz servo waveform

// Trial & Error on an HXT5010 Servo

//#define LOWER   1200         // Lower bound for OCR1A: 0°

//#define UPPER   4800         // Upper Bound for OCR1A: 180°

//#define TOP     10\*UPPER     // Threshold for 50Hz servo waveform

// flag to signal interrupt occurence

volatile boolean triggered = false;

int8\_t delta = 50;            // influences the speed or the servo horn rotation

void setup() {

  Serial.begin(9600);

  while (!Serial);

  //disable global interrupt system while configuring...

  cli();

  // Define OC1A (Arduino pin 9) for output

  DDRB |= (1 << PB1);     //aka pinMode(9, OUTPUT);

  // Configure OC1A pin to clear on Compare Match (non-inverting)

  //  and WGM parameters for Fast PWM Mode 14  (ICR1 as TOP)

  // NOTE: Do not use |= as the Arduino Toolchain sets some flags before this

  TCCR1A = (1 << COM1A1) | (1 << WGM11);

  // Set prescaler and further WGM parameters for Fast PWM Mode 14

  TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS11); //CLKDIV8

  // Configure PWM characteristics

  ICR1  = TOP;    // Establish frequency of 50Hz (period of 20ms)

  OCR1A = LOWER;  // Duty cycle:

  // Enable Output Compare Interrupt ability

  TIMSK1 = 1 << OCIE1A;

  //reenable global interrupt system...

  sei();

}

// Timer1 Output Compare triggered on TCNT1 == TOP (ICR1)

ISR(TIMER1\_COMPA\_vect) {

  triggered = true;

}

void loop() {

  if (triggered) {        // did a Compare Match occur between TCNT1 and OCR1A?

    triggered = false;    // if so, prepare to recognize the next one...

    //Serial.println("OCR1A Compare Match"); // say so, if necessary...

    OCR1A += delta;     // prepare to rotate the horn

    if (OCR1A < LOWER || OCR1A > UPPER)  //rverse direction if necessary...

      delta = -delta;

  }

}