# PWM management

Arduino pins allocations is presented here bellow :

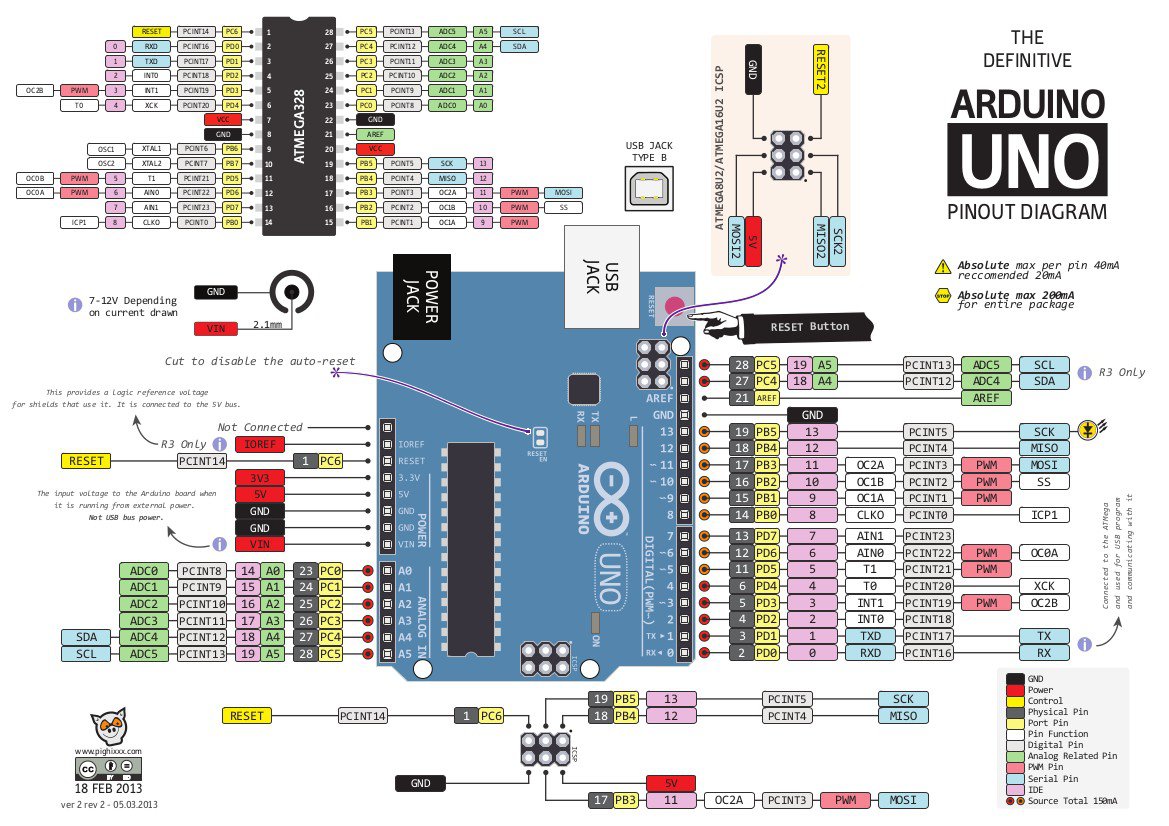


Figure : Ardino pins allocations

During Arduino computations, PWM’s signals used to trigger upper MOSFET will run repeatedly. The only variable that will change is the OCRx threshold use to flip the associated output OCx. Indeed, this procedure is quite fast while setting up a new PWM is about 100 us which is too long for this application.

HA is associated to Arduino pin 9. And HB is associated to Arduino pin 10

Which are related to timer1, in order to set PWM of timer1:

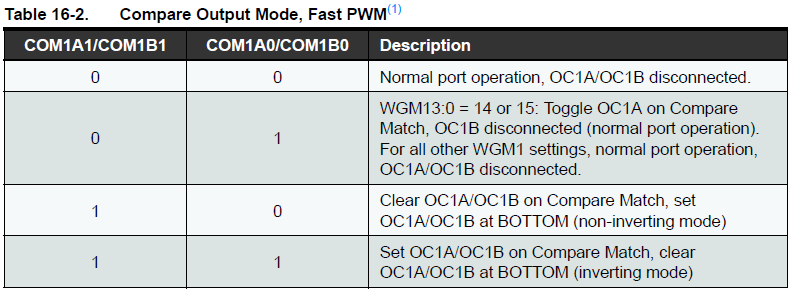
CSn2:0 = 1 => CS12:0 = 1 (n being timer value) the prescaler would be 1 (see p138 atmega328 datasheet)

WGM13:0 = 5 which means timer counter based on 8bits (0x00FF) ( p123 atmega328 datasheet)

Frequency:

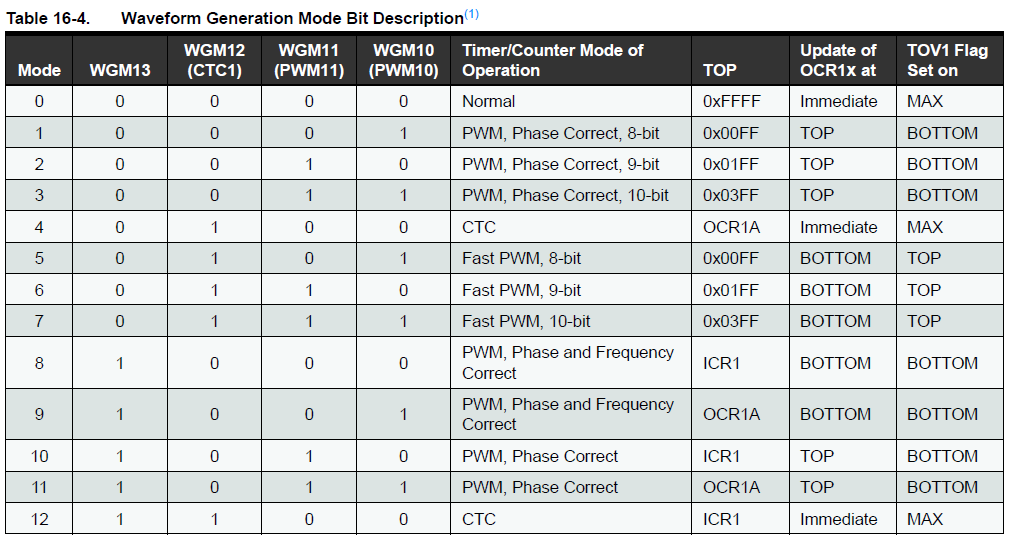


f= 16Mhz/1/256 = 62.5 Khz

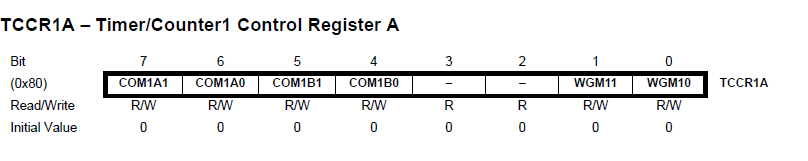


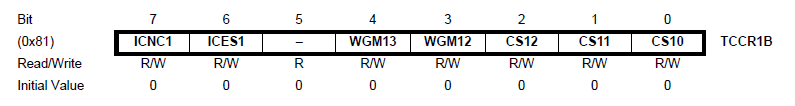
COM1A=3 => (COM1A1:0=3)

COM1B=3



WGM1 = 5 => (WGM13:0 = 5)





Timer1 set up code:

pinMode(9, OUTPUT);

pinMode(10, OUTPUT);

TCCR1A= 11110001 = 241

// TCCR1A = \_BV(COM1A1) | \_BV(COM1B1) | \_BV(WGM11) | \_BV(WGM10);

TCCR1B =00001001 =9

//TCCR1B = \_BV(CS12);

Timer1 in main loop:

OCR1A= INV\_PWM (max0 if ON state) or 255(if OFF state)

OCR1B= PWM (max256 if ON state) or 0(if OFF state)

Timer2 set up code:

pinMode(11, OUTPUT);

TCCR2A= 11110011 = 243

// TCCR1A = \_BV(COM1A1) | \_BV(COM1B1) | \_BV(WGM11) | \_BV(WGM10);

TCCR2B =00000001 =9

//TCCR1B = \_BV(CS12);

Timer1 in main loop:

OCR2A= INV\_PWM (max0 if ON state) or 255(if OFF state)

Set timer 0 for interrupt :

Frequency of interruption = 16MHz/256/8(prescaler) = 7 812, 5 Hz = 128 uS

TCCR0A = 0;

**TCCR0B = 2 (prescale of 8)**

**TIMSK0 = 2 ; (enable compare interrupt – TCNT0 will be equal to** OCR0A**)**

**OCR0A = 255 (comparison reference)**

OCF0A **(set to one when timer found comparison ok) 🡪should be associated to interupt function: (automatically reset to 0 when interruption occurs**

ISR(TIMER0\_COMPA\_vect)

{

Update\_Switch();

}

Void Setup()

{

//…

sei(); // ensures that interrupts are globally enabled

}