EE3704 Embedded System

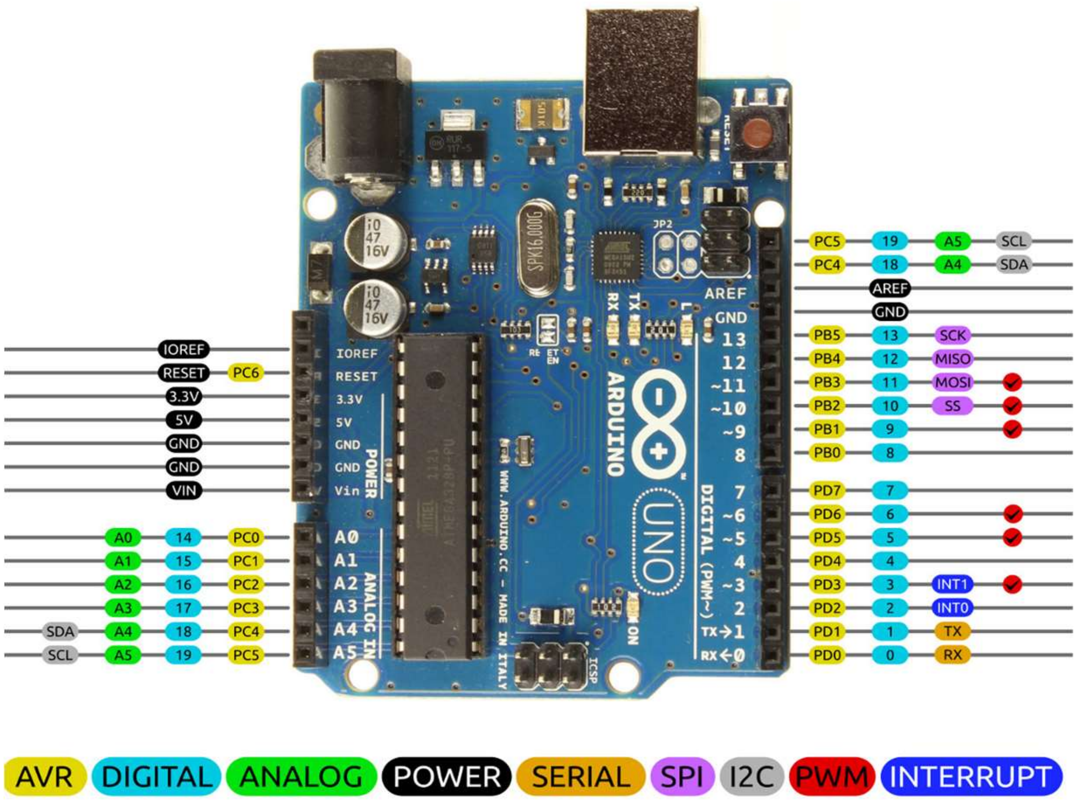
Chapter 6

Presented by

Asst. Prof. Dr.Narong Aphiratsakun

Chapter 6 : Analog Output (share with PWM (8 bits))

Arduino Uno R3 Pinout



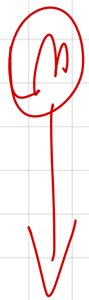
Function:

analogWrite();

Pulse Width Modulation (PWM)

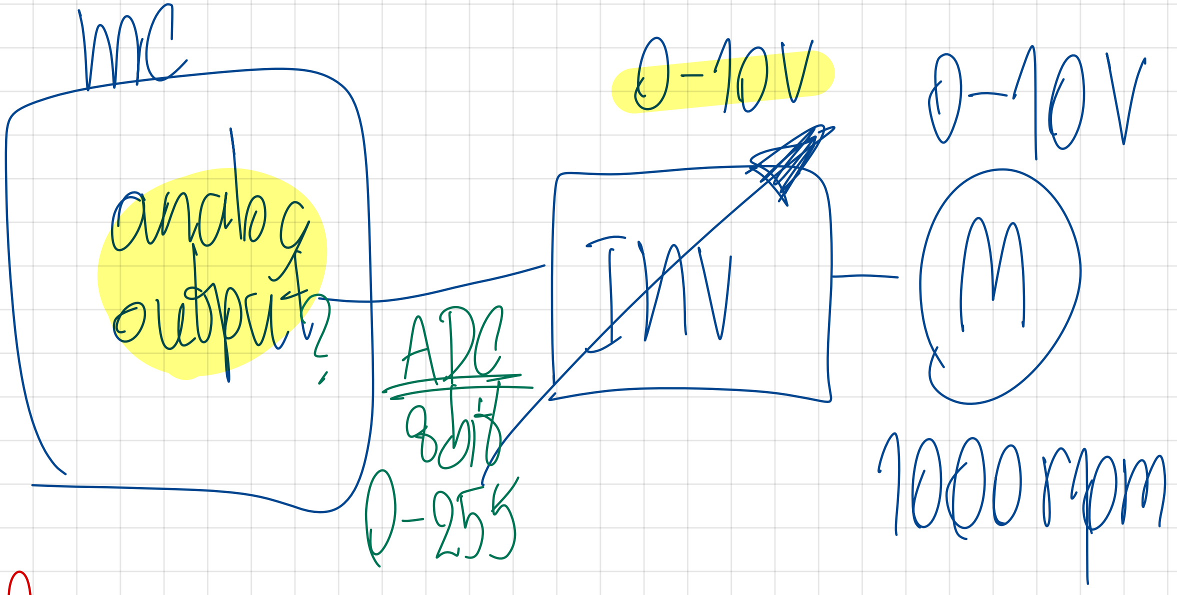
average
not proper

12V 6V



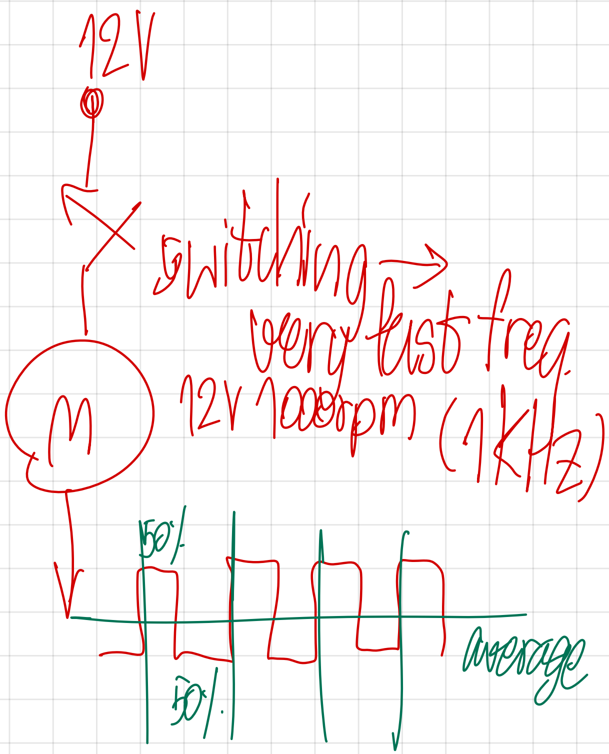
12V 1000 rpm
500 rpm

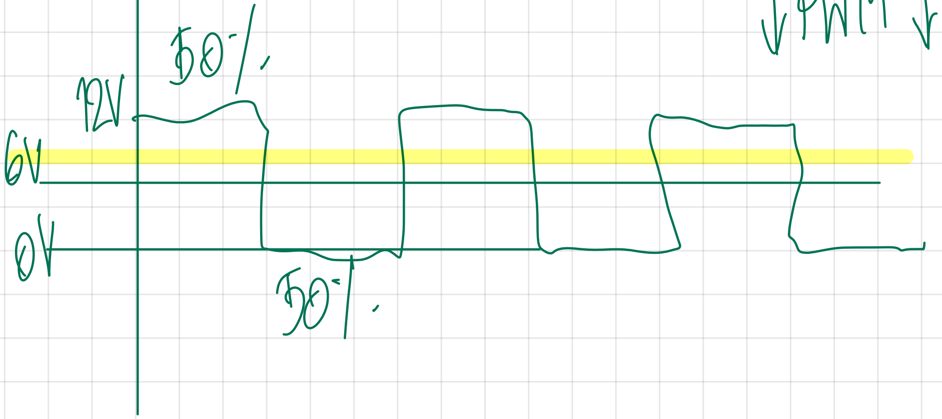
1V 83.33 rpm



For Arduino
0-0V
255-5V

0	0V - 0rpm
127	5V - 500 rpm
255	10V - 1000rpm

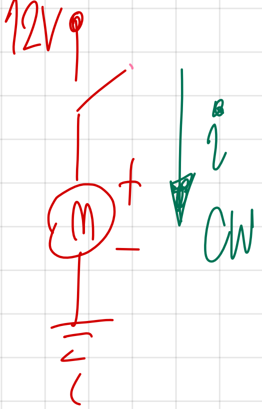


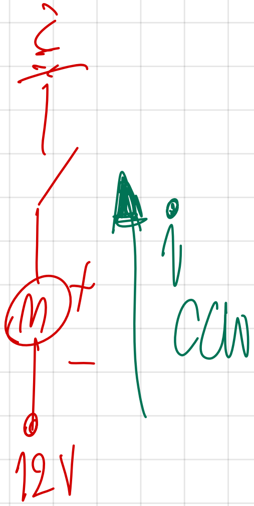


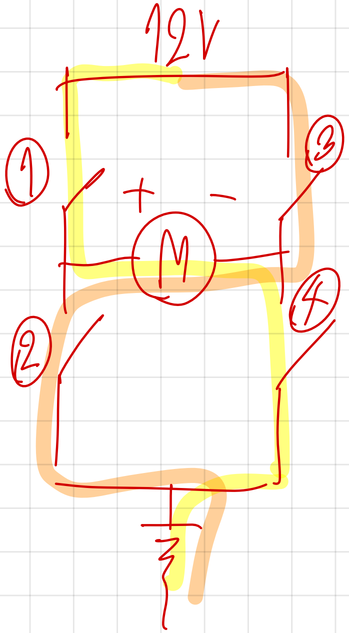
↑PWM ↑V

↓PWM ↓V

H-Bridge







① PWM → speed

② Direction A 0 } CW
Direction B 1
Direction A 1 } CCW
Direction B 0

①+④ ON → CW

②+③ ON → CCW

Chapter 6 : Pulse Width Modulation (PWM)

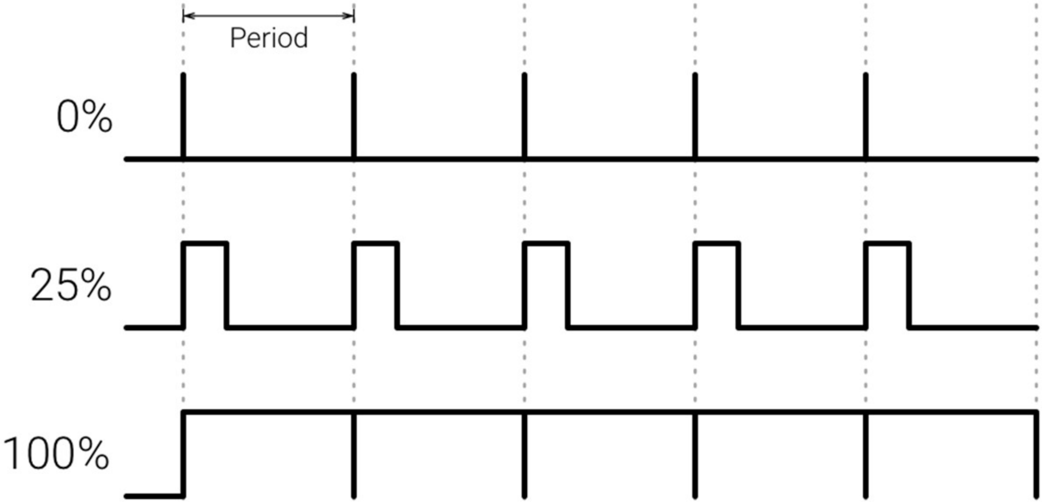
- Switching frequency of arduino Pins (default)
  - Pins 5, 6 : 1 KHz
  - Pins 9, 10, 11 : 500 Hz

Pin 3

- Duty cycle can be set
  - 0% as 0
  - 100% as 255

100   255

75   $\frac{255\times75}{100}$

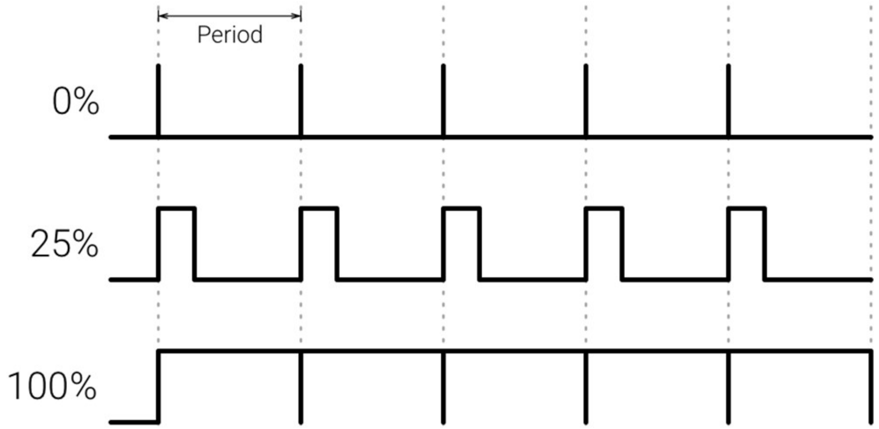


Chapter 6 : Pulse Width Modulation (PWM)

- Example 6.1:
- Use PIN 5 as PWM

- Measure with oscilloscope
  - Set duty cycle of PWM = 20%
  - Set duty cycle of PWM = 50%
  - Set duty cycle of PWM = 75%

- From each duty cycle, measure output voltage from respective duty cycle?

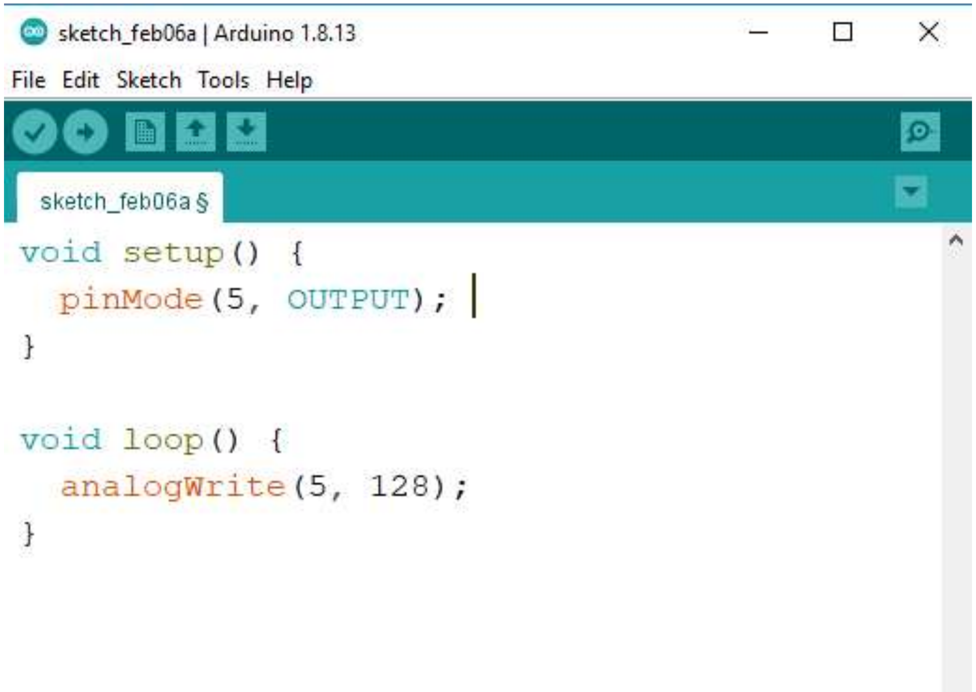


- What is frequency of the PWM

……………………………………………… Hz

Chapter 6 : Pulse Width Modulation (PWM)

- Set duty cycle of PWM = 50%



```
void setup() {
  pinMode(5, OUTPUT);
}

void loop() {
  analogWrite(5, 128);
}
```

Chapter 6 : Pulse Width Modulation (PWM)

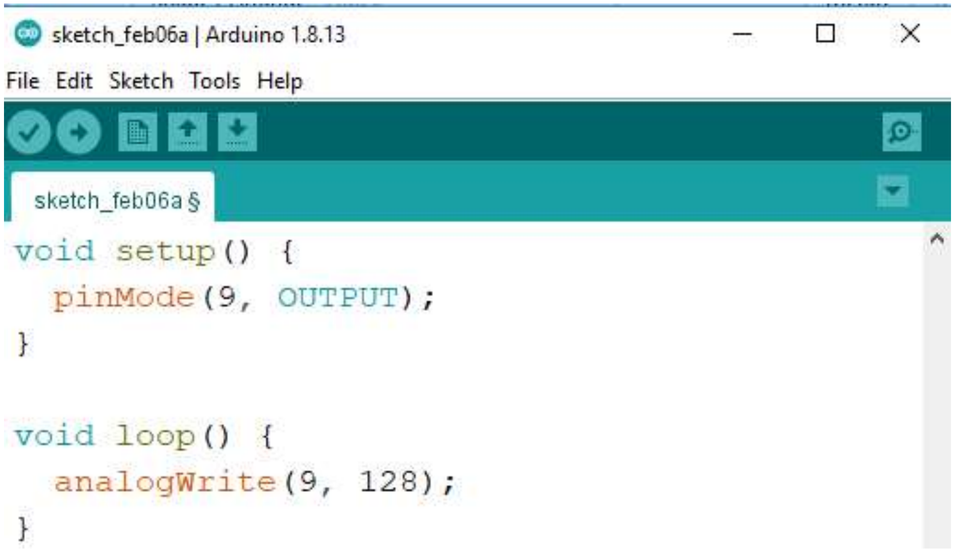
- Example 6.2:
- Use PIN 9 as PWM

- Measure with oscilloscope
  - Set duty cycle of PWM = 20%
  - Set duty cycle of PWM = 50%
  - Set duty cycle of PWM = 75%

- From each duty cycle, measure output voltage from respective duty cycle?

- What is frequency of the PWM

………………………………………………. Hz



```
void setup() {
  pinMode(9, OUTPUT);
}

void loop() {
  analogWrite(9, 128);
}
```

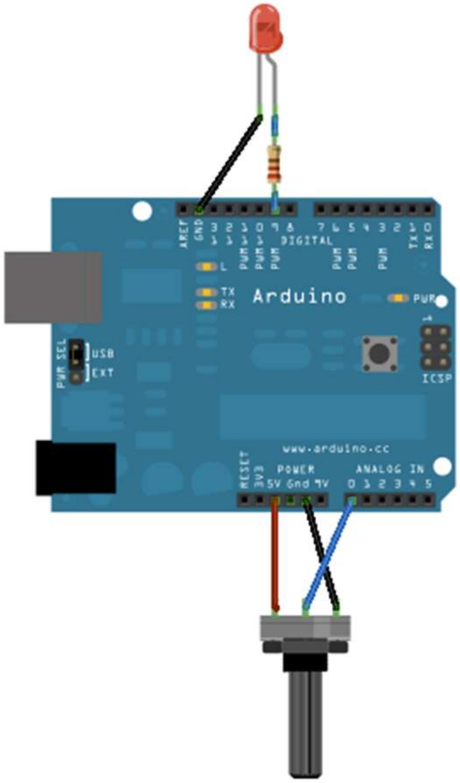
Chapter 6 : PWM

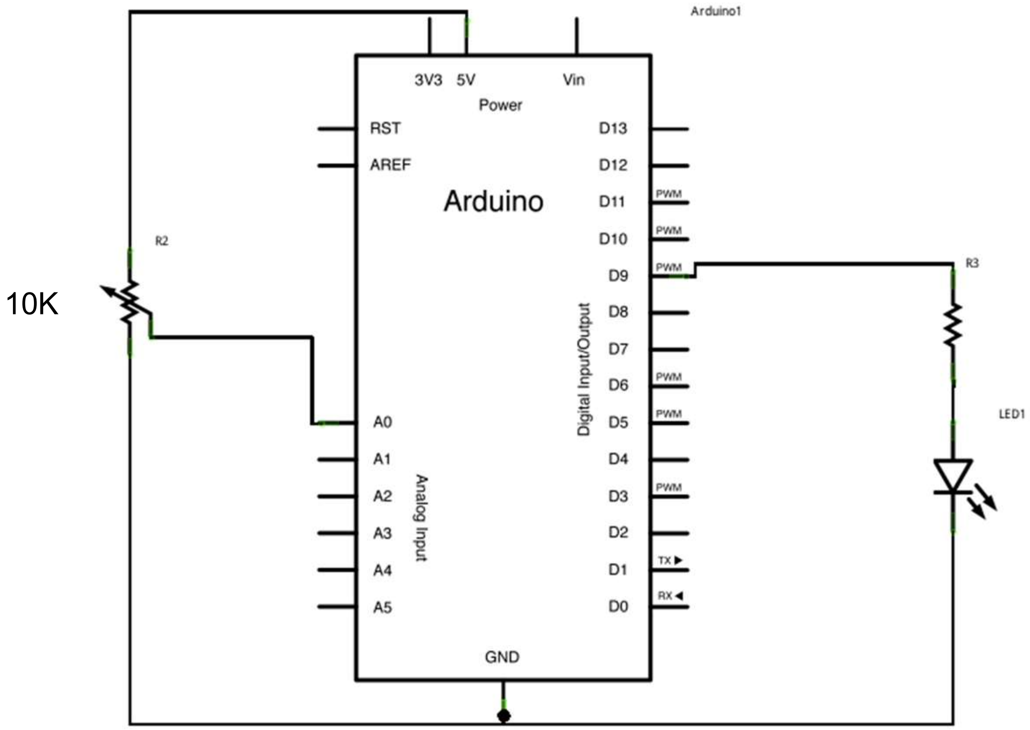
Example 6.3: Control duty cycle of PWM with DIP SW (active high connection), use switching frequency as 500 Hz.

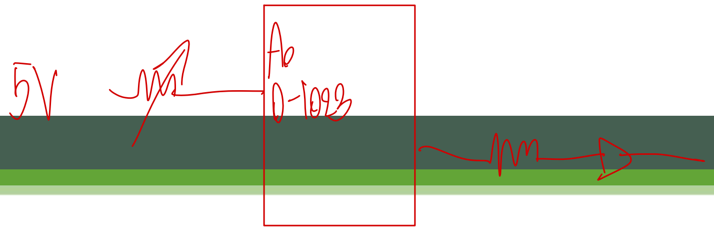
- Press Sw1 : AH-LED1 Blink 25%, PWM data show in serial and measure analog voltage output from multimeter
- Press Sw2 : AH-LED1 Blink 50%, PWM data show in serial and measure analog voltage output from multimeter
- Press Sw3 : AH-LED1 Blink 75%, PWM data show in serial and measure analog voltage output from multimeter
- Press Sw4 : AH-LED1 Blink 100%, PWM data show in serial and measure analog voltage output from multimeter
- Others : AH-LED1 OFF
- Show Circuit diagram, Coding, results (with LEDs and serial data)

Chapter 6 : Pulse Width Modulation (PWM)

Example 6.4: Control duty cycle of PWM with analog reading







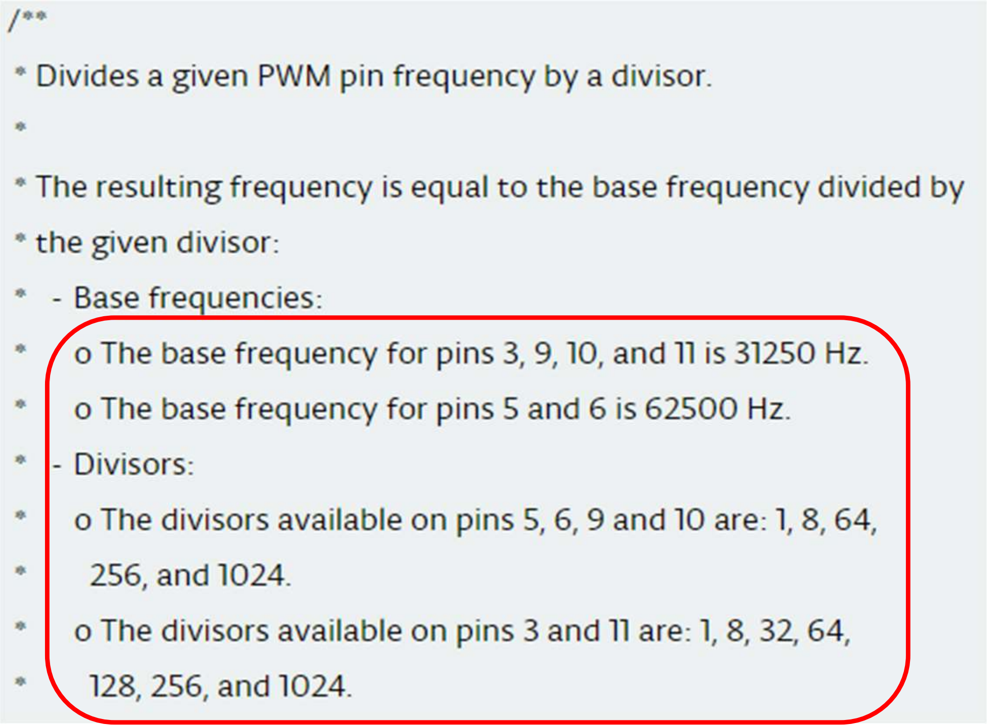
Chapter 6 : Pulse Width Modulation (PWM)

Example 6.4: Control duty cycle of PWM with analog reading

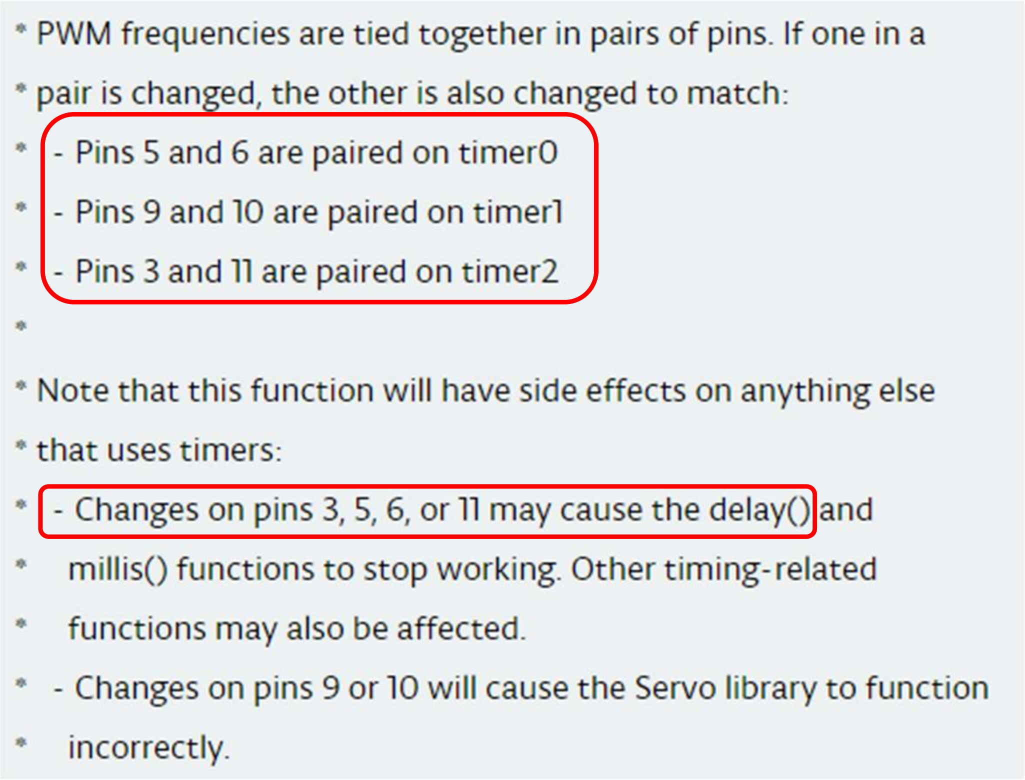
- Analog read XX% : AH-LED1 Blink XX%, analog reading value show in serial monitor as percentage
- Example xx = 10%
    - Analog read 10% : AH-LED1 Blink 10%, analog reading value show in serial monitor as percentage
- Show Circuit diagram, Coding, results (with LEDs and serial data)

Chapter 6 : Pulse Width Modulation (PWM)

- To set PWM switching frequency



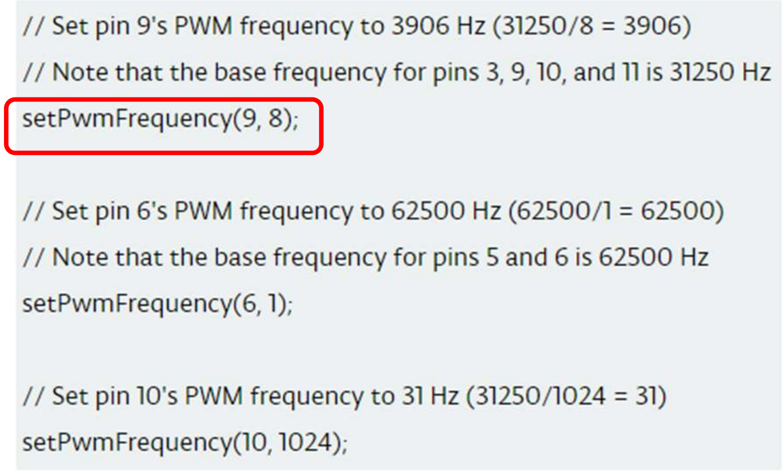
```
/**
 * Divides a given PWM pin frequency by a divisor.
 *
 * The resulting frequency is equal to the base frequency divided by
 * the given divisor:
 *   - Base frequencies:
 *     o The base frequency for pins 3, 9, 10, and 11 is 31250 Hz.
 *     o The base frequency for pins 5 and 6 is 62500 Hz.
 *   - Divisors:
 *     o The divisors available on pins 5, 6, 9 and 10 are: 1, 8, 64,
 *       256, and 1024.
 *     o The divisors available on pins 3 and 11 are: 1, 8, 32, 64,
 *       128, 256, and 1024.
```



```
 * PWM frequencies are tied together in pairs of pins. If one in a
 * pair is changed, the other is also changed to match:
 *   - Pins 5 and 6 are paired on timer0
 *   - Pins 9 and 10 are paired on timer1
 *   - Pins 3 and 11 are paired on timer2
 *
 * Note that this function will have side effects on anything else
 * that uses timers:
 *   - Changes on pins 3, 5, 6, or 11 may cause the delay() and
 *     millis() functions to stop working. Other timing-related
 *     functions may also be affected.
 *   - Changes on pins 9 or 10 will cause the Servo library to function
 *     incorrectly.
```

Chapter 6 : PWM

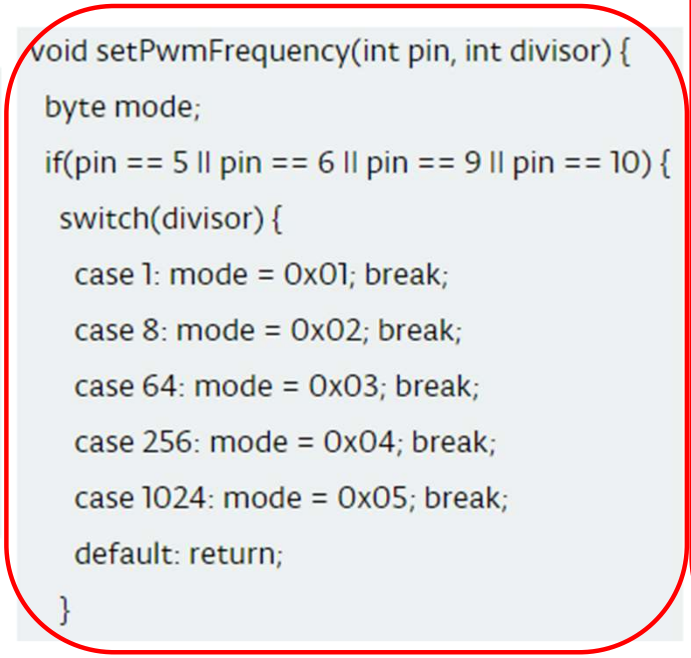
- To set PWM switching frequency



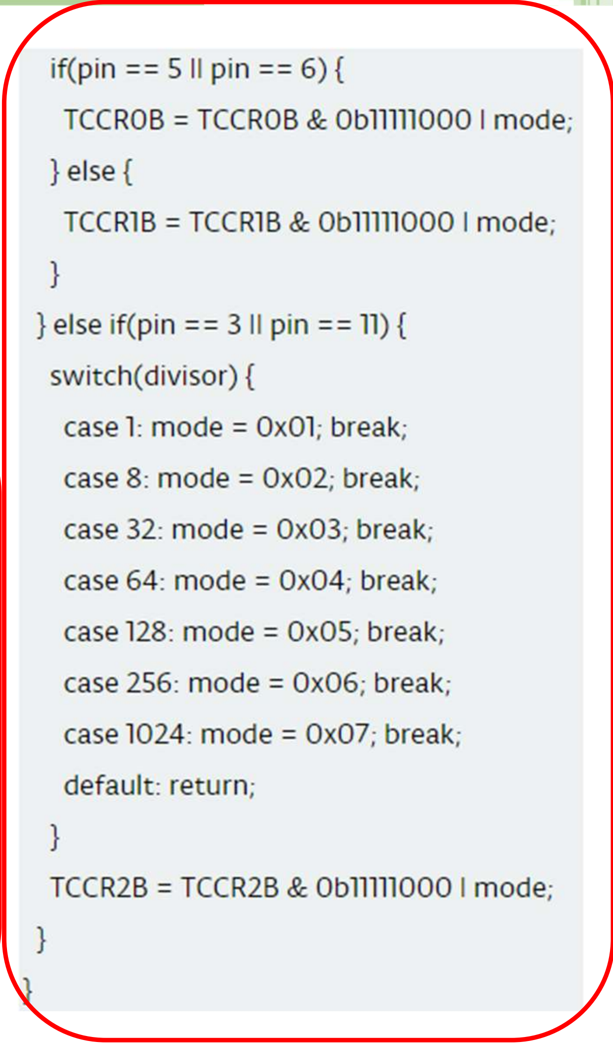
```
// Set pin 9's PWM frequency to 3906 Hz (31250/8 = 3906)
// Note that the base frequency for pins 3, 9, 10, and 11 is 31250 Hz
setPwmFrequency(9, 8);

// Set pin 6's PWM frequency to 62500 Hz (62500/1 = 62500)
// Note that the base frequency for pins 5 and 6 is 62500 Hz
setPwmFrequency(6, 1);

// Set pin 10's PWM frequency to 31 Hz (31250/1024 = 31)
setPwmFrequency(10, 1024);
```



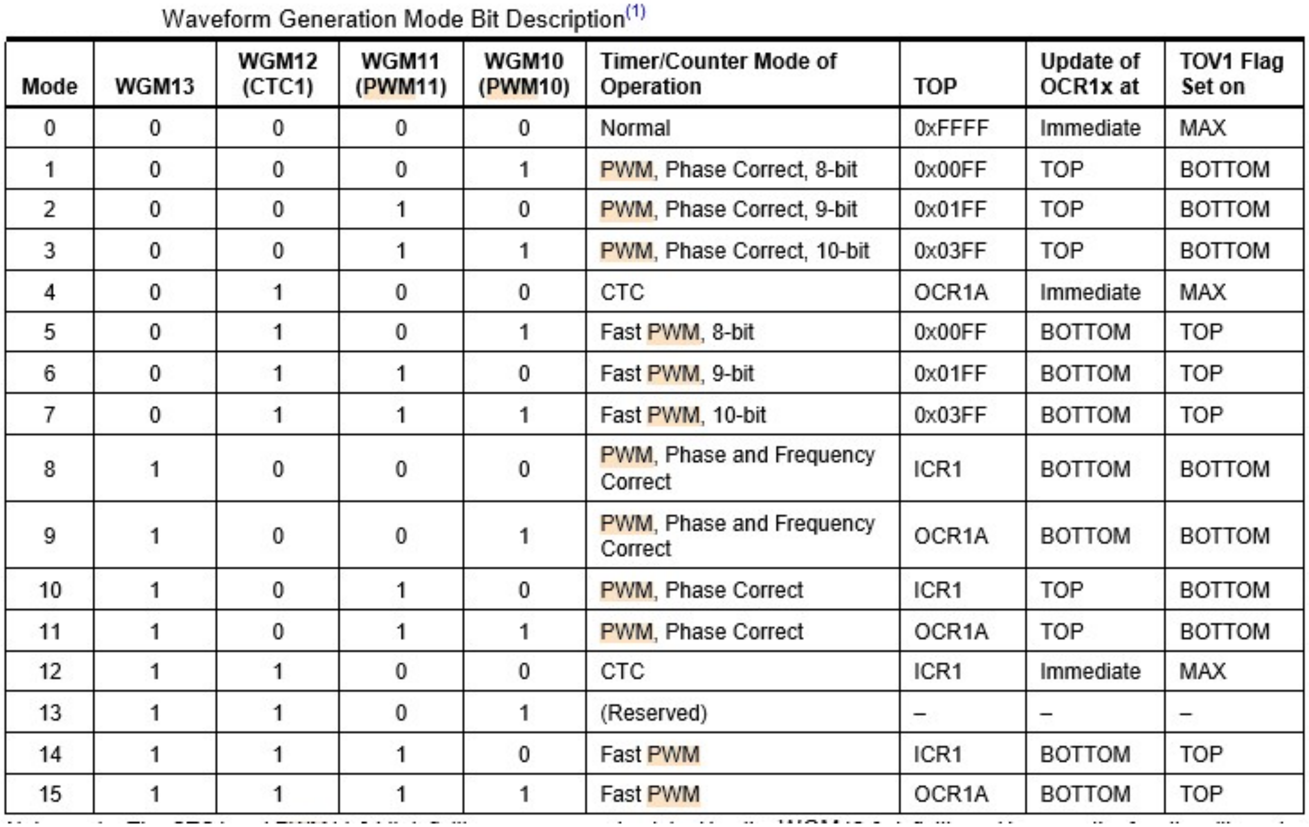
```
void setPwmFrequency(int pin, int divisor) {
 byte mode;
 if(pin == 5 || pin == 6 || pin == 9 || pin == 10) {
  switch(divisor) {
   case 1: mode = 0x01; break;
   case 8: mode = 0x02; break;
   case 64: mode = 0x03; break;
   case 256: mode = 0x04; break;
   case 1024: mode = 0x05; break;
   default: return;
  }
```



```
  if(pin == 5 || pin == 6) {
   TCCR0B = TCCR0B & 0b11111000 | mode;
  } else {
   TCCR1B = TCCR1B & 0b11111000 | mode;
  }
 } else if(pin == 3 || pin == 11) {
  switch(divisor) {
   case 1: mode = 0x01; break;
   case 8: mode = 0x02; break;
   case 32: mode = 0x03; break;
   case 64: mode = 0x04; break;
   case 128: mode = 0x05; break;
   case 256: mode = 0x06; break;
   case 1024: mode = 0x07; break;
   default: return;
  }
  TCCR2B = TCCR2B & 0b11111000 | mode;
 }
}
```

Chapter 6 : Pulse Width Modulation (PWM)

- Register

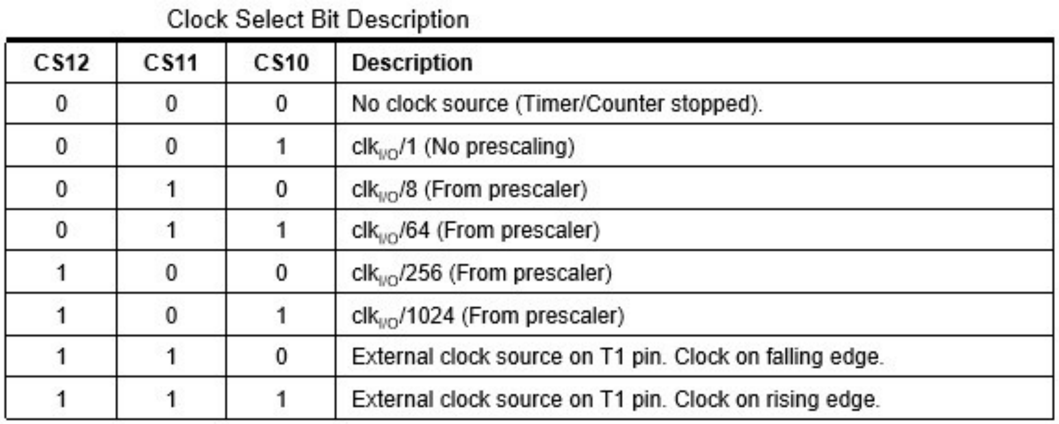


Waveform Generation Mode Bit Description[1]

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Chapter 6 : PWM

- function setPwmFrequency(int pin, int divisor)

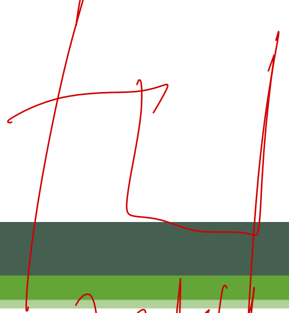


Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/1 (No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

```
void setPwmFrequency(int pin, int divisor) {
  byte mode;
  if(pin == 5 || pin == 6 || pin == 9 || pin == 10) {
    switch(divisor) {
      case 1: mode = 0x01; break;
      case 8: mode = 0x02; break;
      case 64: mode = 0x03; break;
      case 256: mode = 0x04; break;
      case 1024: mode = 0x05; break;
      default: return;
    }
    if(pin == 5 || pin == 6) {
      TCCR0B = TCCR0B & 0b11111000 | mode;
    } else {
      TCCR1B = TCCR1B & 0b11111000 | mode;
    }
  } else if(pin == 3 || pin == 11) {
    switch(divisor) {
      case 1: mode = 0x01; break;
      case 8: mode = 0x02; break;
      case 32: mode = 0x03; break;
      case 64: mode = 0x04; break;
      case 128: mode = 0x05; break;
      case 256: mode = 0x06; break;
      case 1024: mode = 0x07; break;
      default: return;
    }
    TCCR2B = TCCR2B & 0b11111000 | mode;
  }
}
```

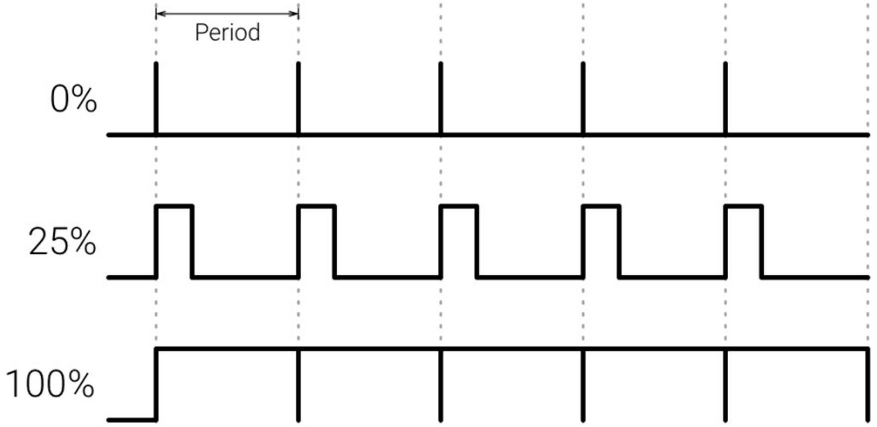
Pin 5 : 62500/8 = 7.8 KHz



7.8 KHz

Chapter 6 : Pulse Width Modulation (PWM)

- Example 6.5 :
- Use PIN 5 as PWM
- Set Pulse Width to 50%
- Set divisor to 8, measure the waveform
- Set divisor to 256, measure the waveform



- What is frequency of the PWM for each divisor

divisor is 8, PWM = ... Hz
divisor is 256, PWM = ... Hz

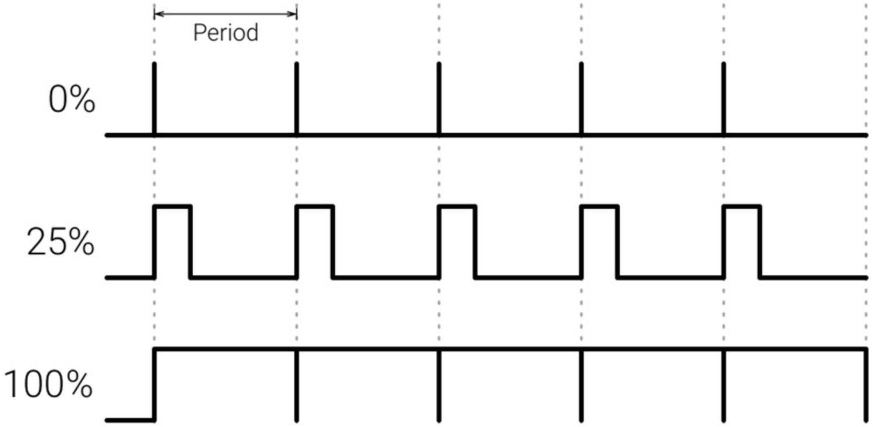
62500/256 = 244. Hz

Pin 9: 31250 / 8 = 3.9 kHz

31250/256 = 122 Hz

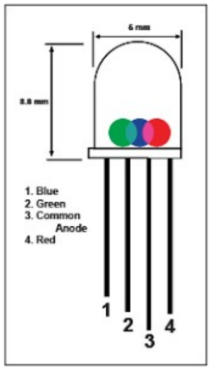
Chapter 6 : Pulse Width Modulation (PWM)

- Example 6.6 :
- Use PIN 9 as PWM
- Set divisor to 8, measure the waveform
- Set divisor to 256, measure the waveform

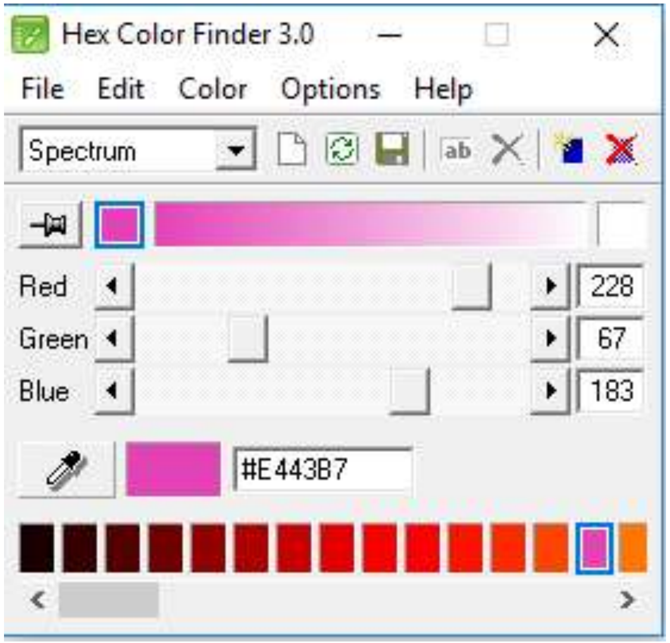


- What is frequency of the PWM for each divisor

divisor is 8, PWM = ... Hz
divisor is 256, PWM = ... Hz



Chapter 6 : Pulse Width Modulation (PWM)



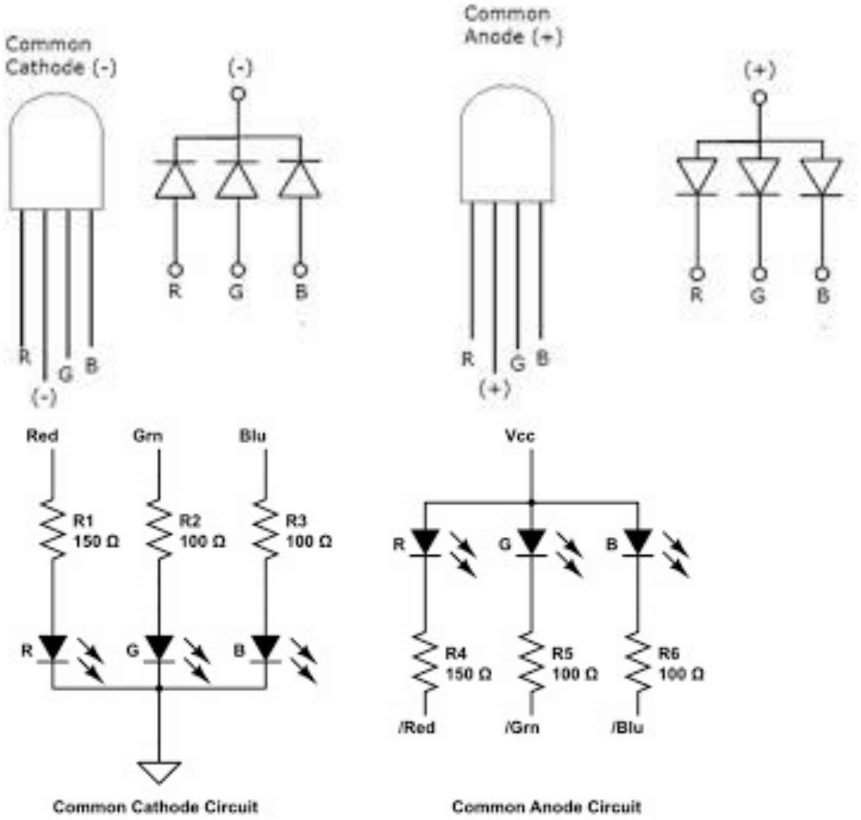
Spectrum

Red 228

Green 67

Blue 183

#E443B7



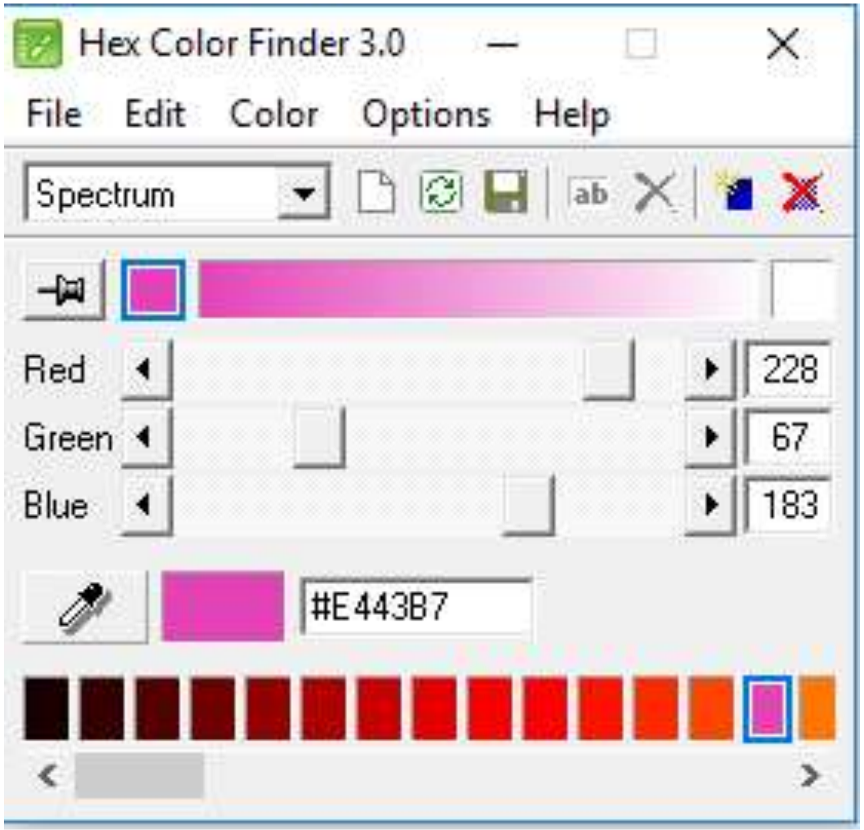
R : 228 : 255-228 = 27

G : 67 : 255-67 = 188

B : 183 : 255-183 = 72

CC CA

Chapter 6 : Pulse Width Modulation (PWM)

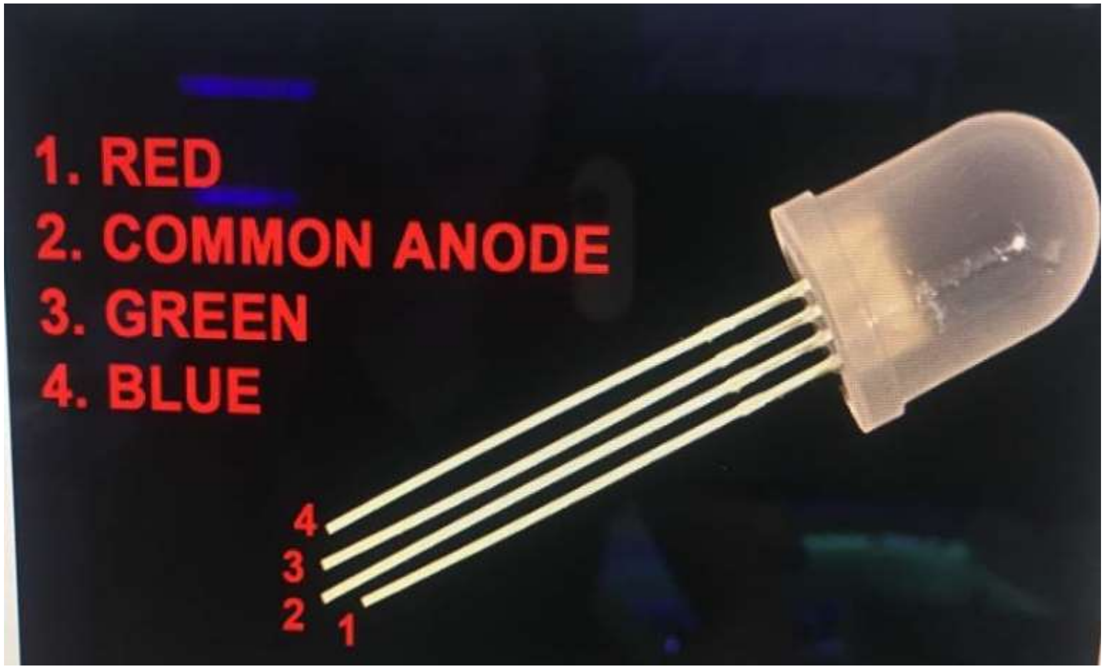


Red 228

Green 67

Blue 183

#E443B7



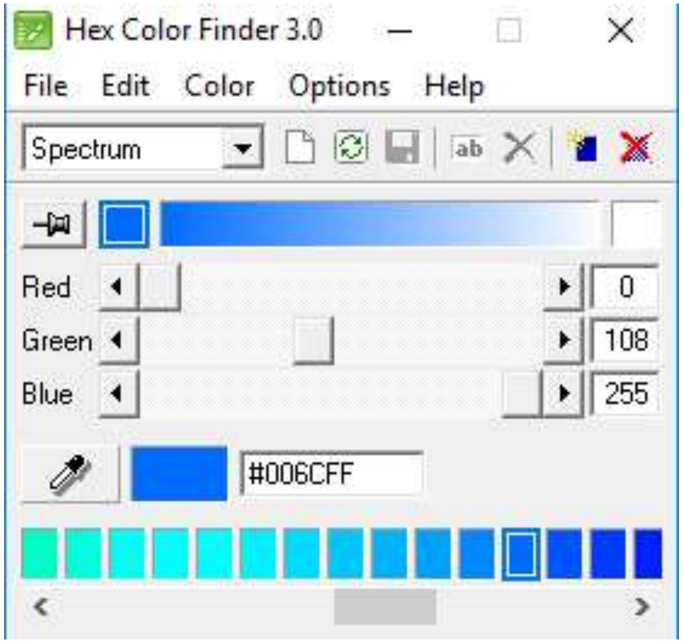
Use R between 200-300 ohm for limiting the current to each LED

Chapter 6 : Pulse Width Modulation (PWM)

Example 6.7: Use RGB module

- Perform RGB LED module as color indicated
  - Pin for R …….., Value 0
  - Pin for G …….., Value 108
  - Pin for B …….., Value 255

Can you obtained color as indicated?



Red 0

Green 108

Blue 255

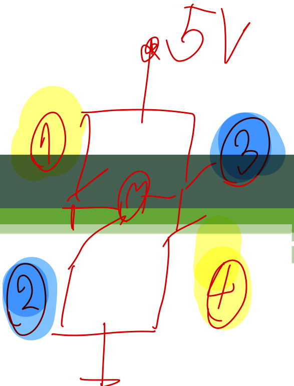
#006CFF

PWM ⇒ Counter Speed

0 — 100%

Lowest 0

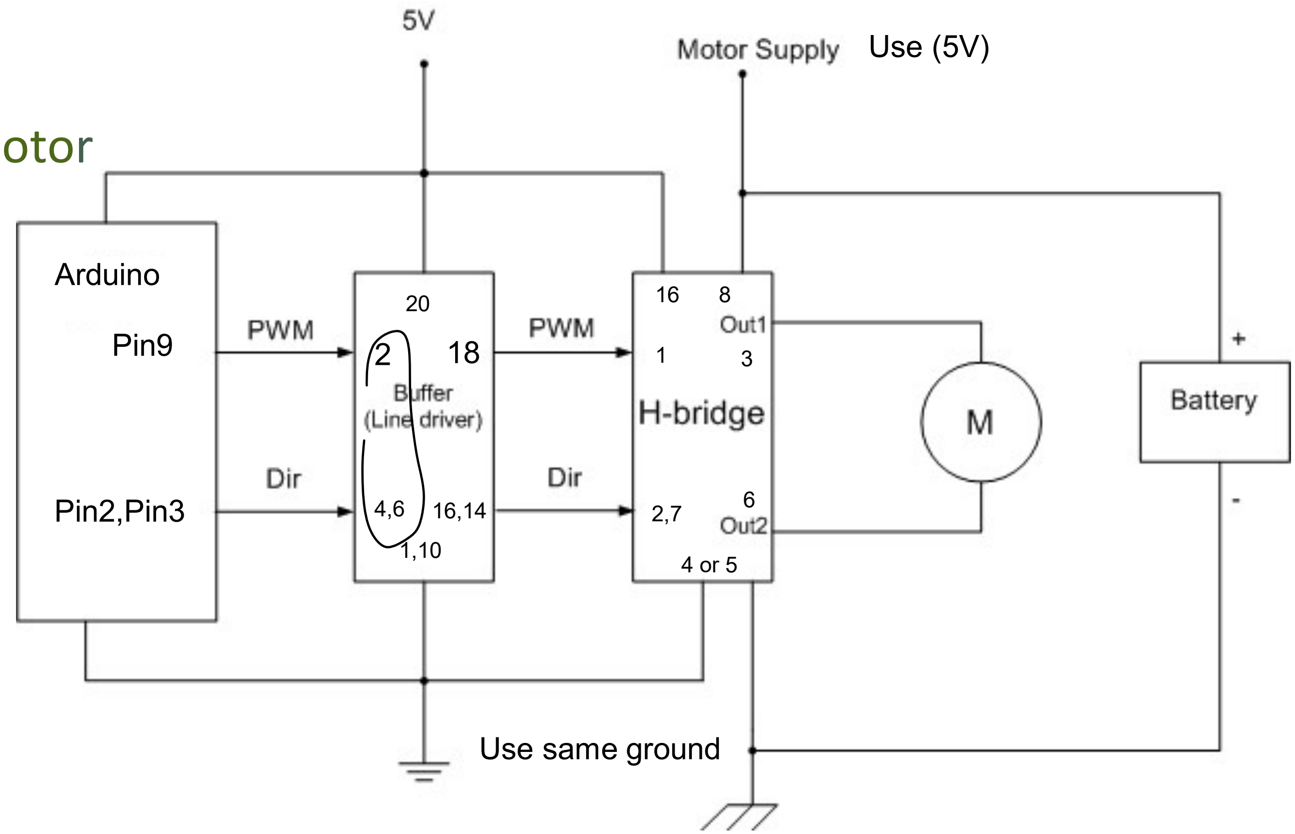
Highest 255



Chapter 6 : Pulse Width Modulation (PWM)

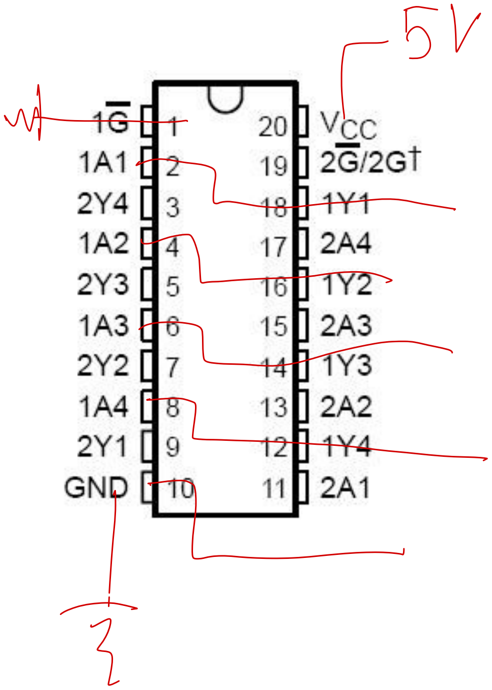
Example 6.8: Speed control of motor

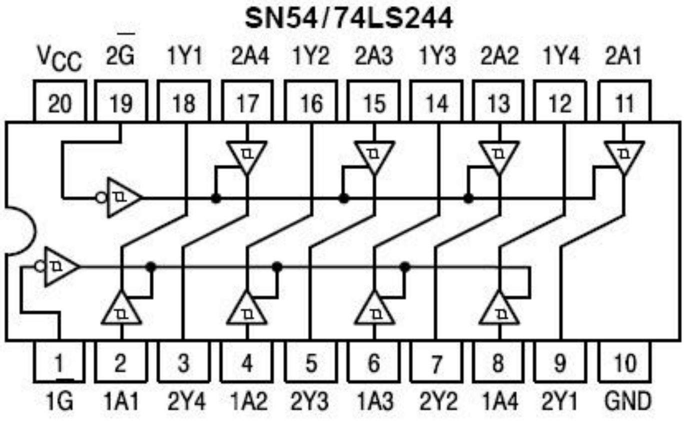
- Set f = 3.9 KHz
- Press “a” from KB, set motor 100% CW
- Press “b” from KB, set motor 60% CW
- Press “s” from KB, set motor 0%
- Press “d” from KB, set motor 100% CCW
- Press “e” from KB, set motor 60% CCW



Chapter 6 : Pulse Width Modulation (PWM)

Line driver L 74Ls244 (buffer)



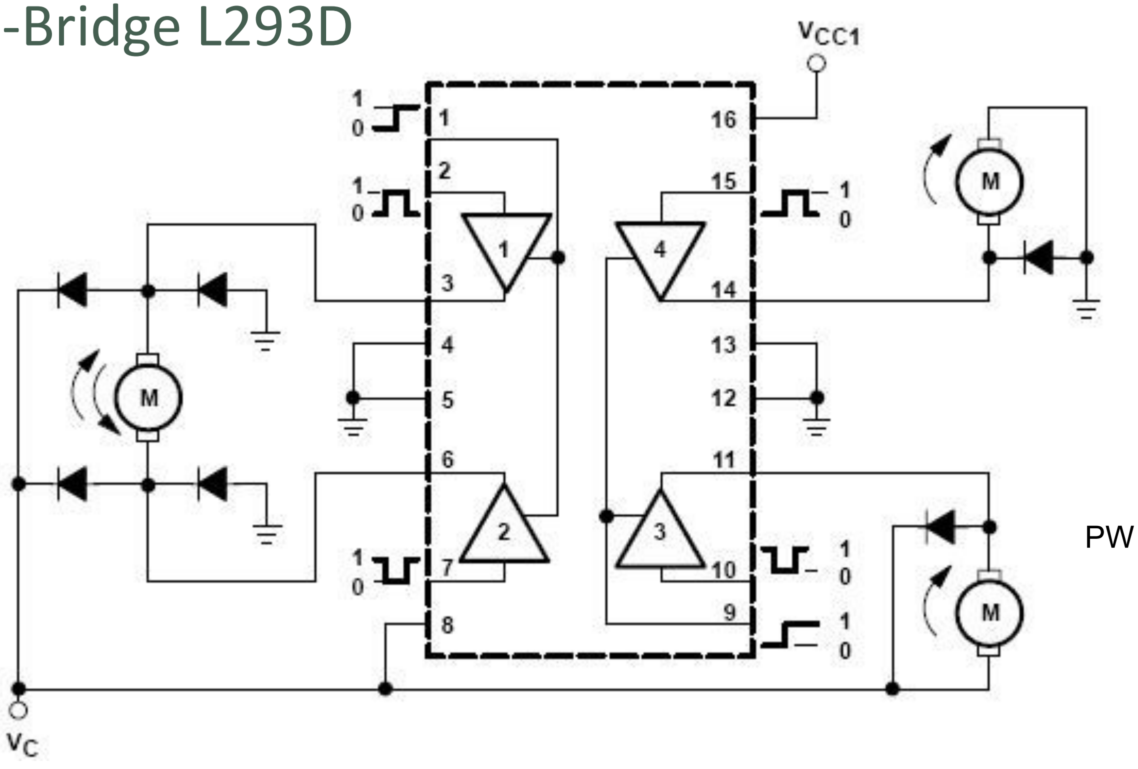


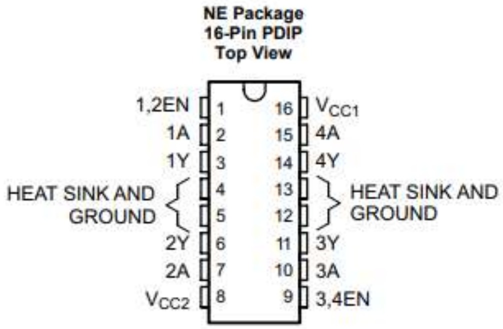
SN54/74LS244

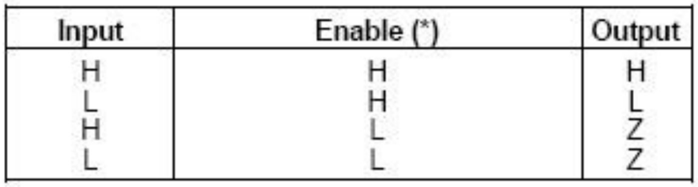
INPUTS		OUTPUT
$\overline{1G}$, $\overline{2G}$	D	
L	L	L
L	H	H
H	X	(Z)

Chapter 6 : Pulse Width Modulation (PWM)

H-Bridge L293D

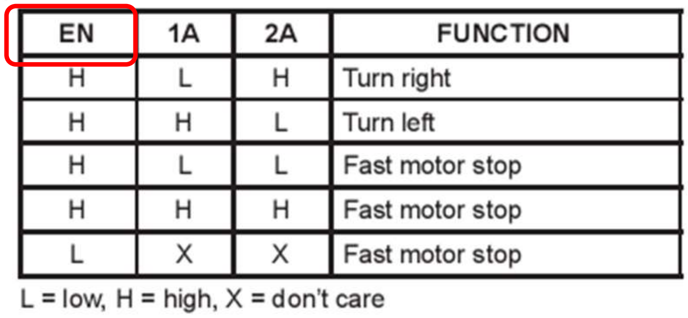






Input	Enable (*)	Output
H	H	H
L	H	L
H	L	Z
L	L	Z

PWM



EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

ASSUMPTION UNIVERSITY OF THAILAND

SINCE 1969

มหาวิทยาลัยอัสสัมชัญ