Mechatronics Engineering

Lab 6

Pulse Width Modulation (PWM)

Open-Loop Speed Control of DC Motor

GUC German University in Cairo

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Tutorial Contents

- Pulse Width Modulation (PWM)
- DC Motors
  - Controlling the Direction : H-Bridge
  - Controlling the Speed: PWM signals
- PWM module in ATmega328P
  - Different Modes
    - Fast PWM
    - Phase Correct PWM
  - Associated Registers
- Analog-to-Digital Converter (ADC) in ATmega328P
- Lab 6 Validation

Tutorial Contents

- **Pulse Width Modulation (PWM)**

- DC Motors
  - Controlling the Direction : H-Bridge
  - Controlling the Speed: PWM signals

- PWM module in ATmega328P
  - Different Modes
    - Fast PWM
    - Phase Correct PWM
  - Associated Registers

- Analog-to-Digital Converter (ADC) in ATmega328P

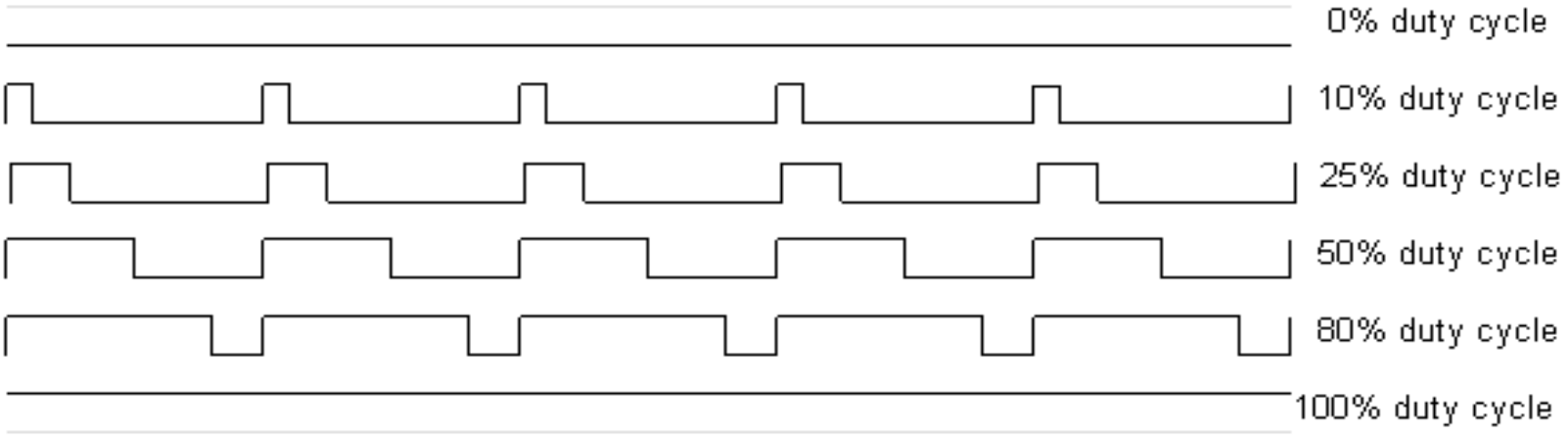
- Lab 6 Validation

Pulse- Width Modulation (PWM)

- A Pulse-Width Modulated (PWM) signal is a digital square wave, where the frequency is constant, but that fraction of the time the signal is HIGH (known as the duty cycle) can be varied between 0 and 100%.

- PWM has several uses, among which are the following.
  - Dimming an LED
  - Providing variable speed control for motors.

- In this Lab, the usage of PWM signals in the speed control of DC motors will be discussed.



Tutorial Contents

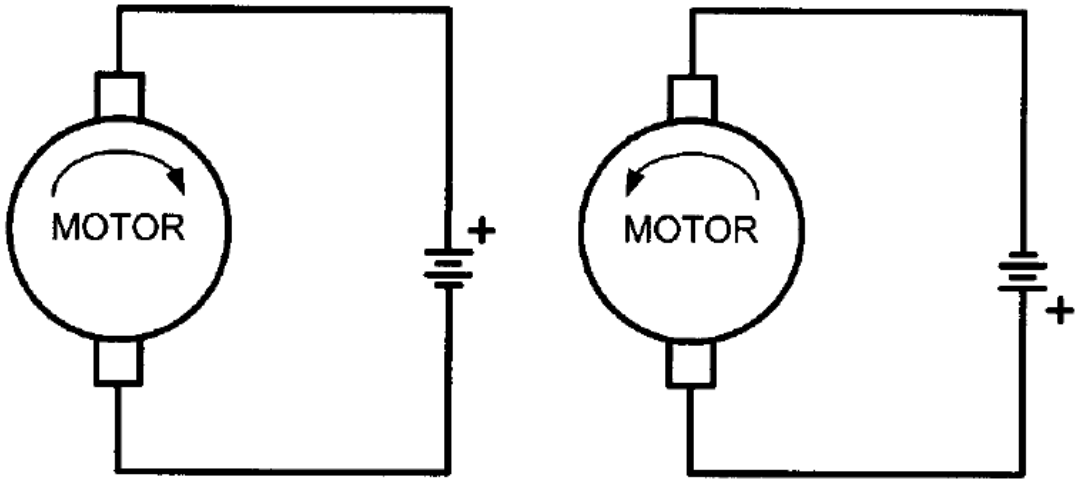
- Pulse Width Modulation (PWM)
- **DC Motors**
  - Controlling the Direction : H-Bridge
  - Controlling the Speed: PWM signals
- PWM module in ATmega328P
  - Different Modes
    - Fast PWM
    - Phase Correct PWM
  - Associated Registers
- Analog-to-Digital Converter (ADC) in ATmega328P
- Lab 6 Validation

DC Motors

- A Direct Current (DC) motor is a widely used device to transform electrical pulses to mechanical movement.

- Connecting the motor terminals to a DC voltage source makes the motor rotate in one direction. If you reverse the polarity of the voltage source, the DC motor rotates in the opposite direction.

- The DC motor has a voltage rating. If the DC voltage source provides this full rating, the motor rotates with full speed. If the source provides less voltage, the speed decrease proportionally.

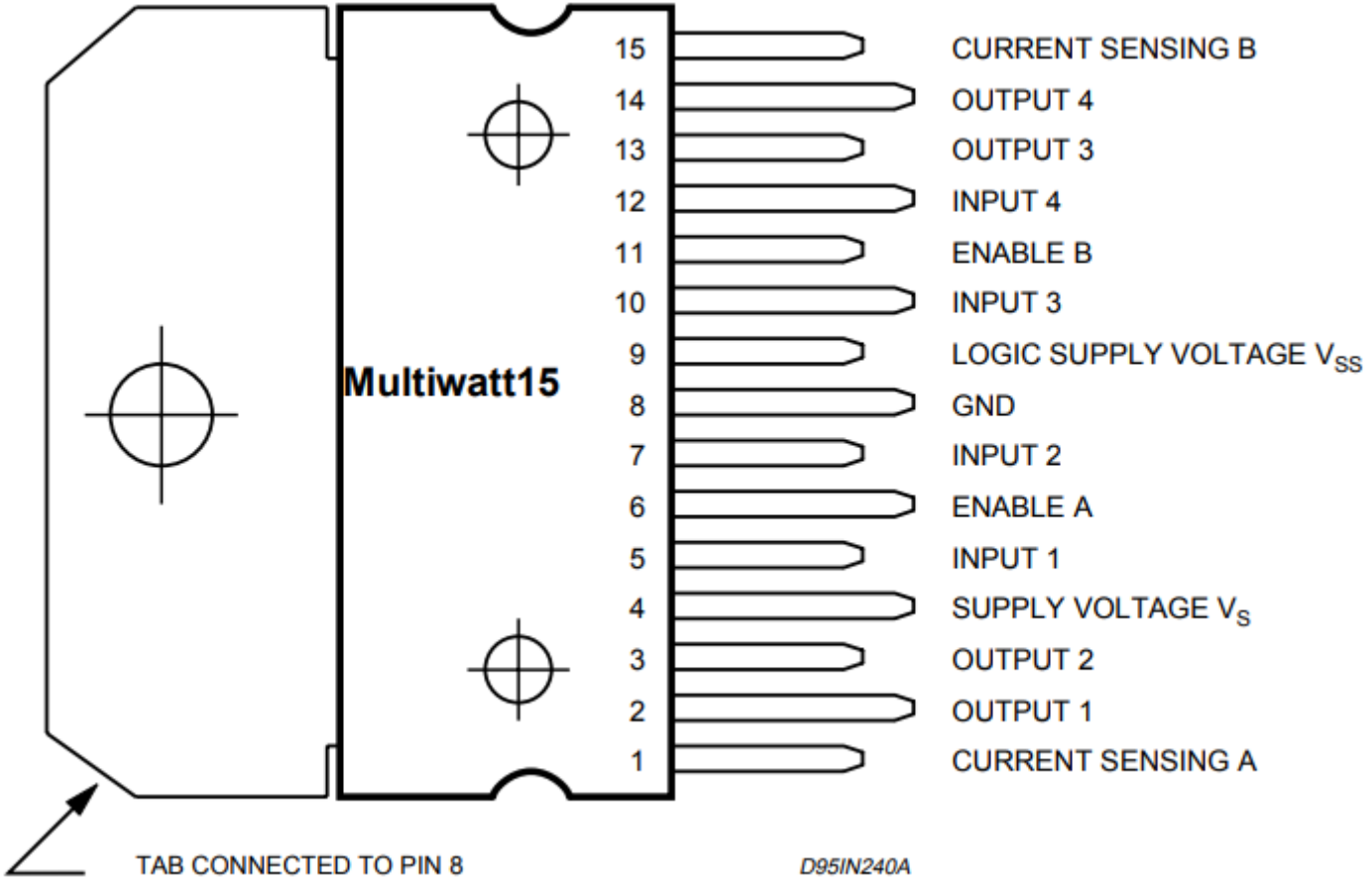


Tutorial Contents

- Pulse Width Modulation (PWM)
- **DC Motors**
  - **Controlling the Direction : H-Bridge**
  - Controlling the Speed: PWM signals
- PWM module in ATmega328P
  - Different Modes
    - Fast PWM
    - Phase Correct PWM
  - Associated Registers
- Analog-to-Digital Converter (ADC) in ATmega328P
- Lab 6 Validation

DC Motors: Controlling the Direction using H-Bridge

- We can control the DC motor's direction of rotation using an H-Bridge L298 chip.



DC Motors: Controlling the Direction using H-Bridge

- We can control the DC motor's direction of rotation using an H-Bridge L298 chip.

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V_S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

DC Motors: Controlling the Direction using H-Bridge

- The L298 chip is a dual full bridge designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals.

- The chip provides two channels – channel A and B. This means it can operate two DC motors simultaneously.

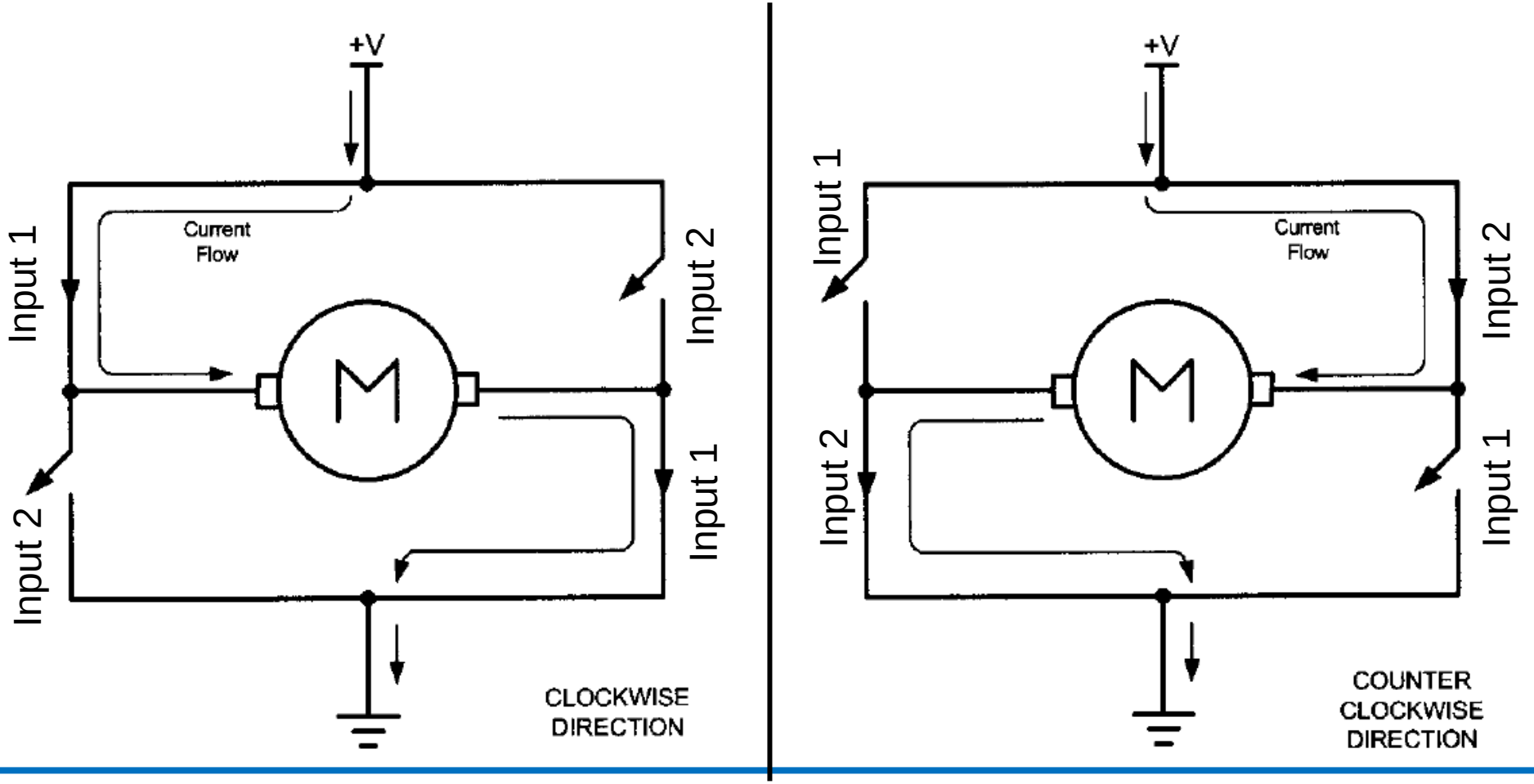
- The maximum current rating for each channel is 2 Amperes. Also, the 2 channels can be connected together to provide a maximum current rating of 4 Amperes.

- Thus, the directions can be controlled according to the following table.

Input 1 and Input 2	Action
00	Motor is free
01	Motor rotates clockwise
10	Motor rotates counter-clockwise
11	Motor brakes

DC Motors: Controlling the Direction using H-Bridge

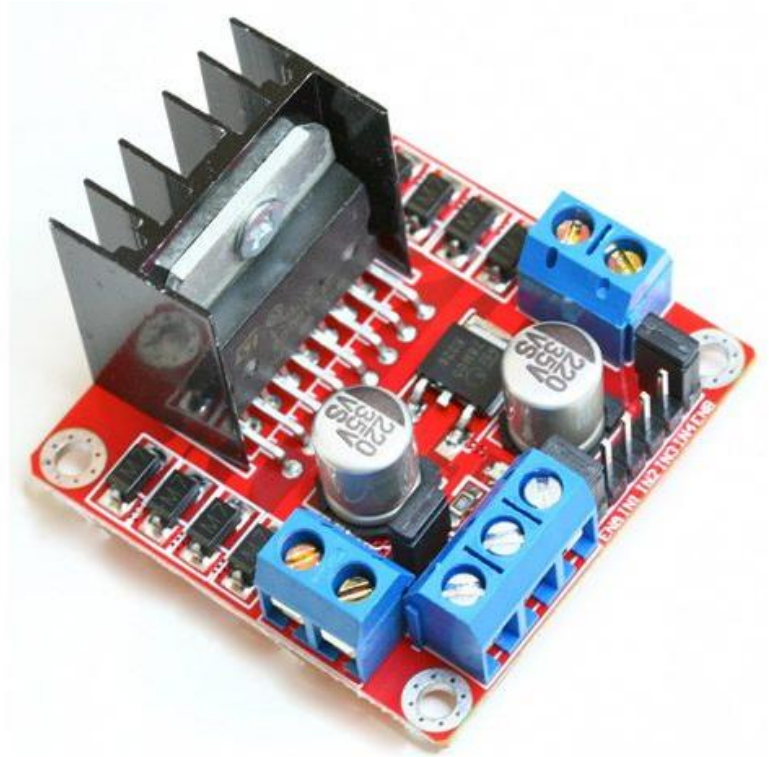
- The concept operation of the H-Bridge can be illustrated in this diagram.



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

DC Motors: Controlling the Direction using H-Bridge

- TThe DC motor driver shown here is based on the L298 chip.
- It takes a rated logical supply voltage V_{ss} of 5 Volts.
- The circuit can drive motors with supply voltages between 9 and 35 Volts.
- Note that the logic supply voltage and the motor supply voltage sources should be different due to the different current ratings.
- The motor driver circuitry includes decoupling capacitors, fly-back diodes and a heat sink. The heat sink protects the IC from eventually burning up from stall currents.



GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Tutorial Contents

- Pulse Width Modulation (PWM)
- **DC Motors**
  - Controlling the Direction : H-Bridge
  - **Controlling the Speed: PWM signals**
- PWM module in ATmega328P
  - Different Modes
    - Fast PWM
    - Phase Correct PWM
  - Associated Registers
- Analog-to-Digital Converter (ADC) in ATmega328P
- Lab 6 Validation

DC Motors: Controlling the Speed using PWM

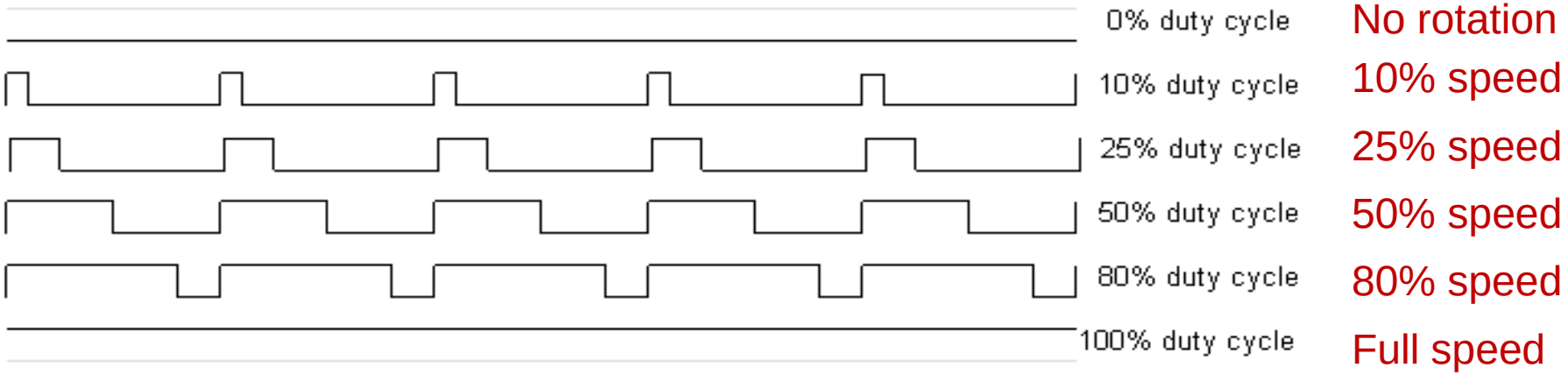
- The PWM signal can be input to the Enable pin of the corresponding channel in the H-Bridge. This will vary the speed of the motor according to the duty cycle.

- The duty cycle is defined as the fraction of the time period for which the PWM is signal HIGH. It can be expressed as a percentage.

$$D\% = \frac{T_{ON}}{T_{ON} + T_{OFF}} \times 100$$

- The duty cycle can also be expressed as an 8-bit digital word.

$$D = \frac{T_{ON}}{T_{ON} + T_{OFF}} \times 255$$



GUC German University in Cairo

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Tutorial Contents

- Pulse Width Modulation (PWM)
- DC Motors
  - Controlling the Direction : H-Bridge
  - Controlling the Speed: PWM signals
- **PWM module in ATmega328P**
  - Different Modes
    - Fast PWM
    - Phase Correct PWM
  - Associated Registers
- Analog-to-Digital Converter (ADC) in ATmega328P
- Lab 6 Validation

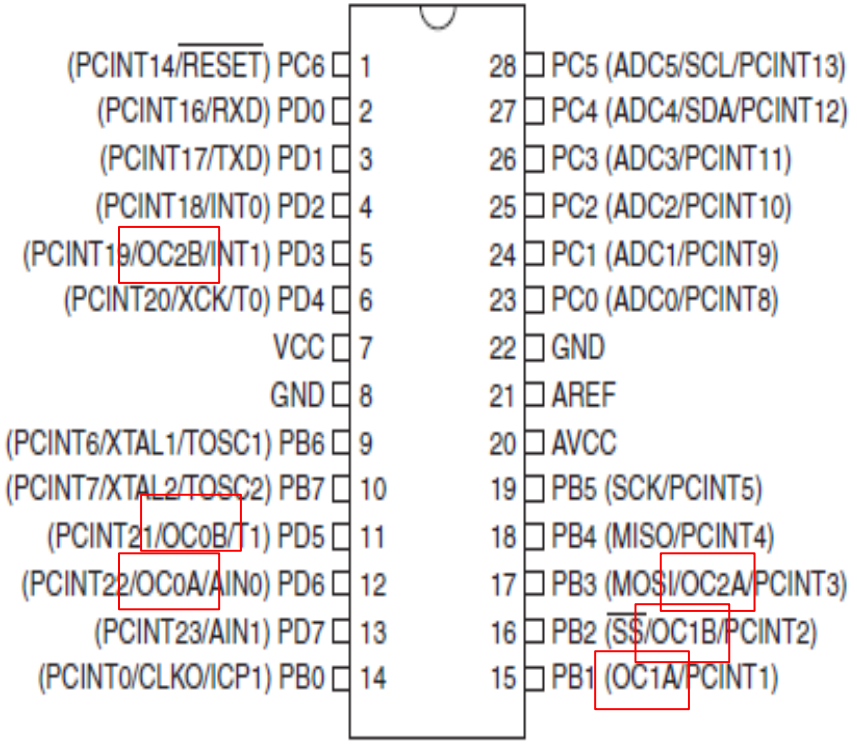
PWM Module in ATmega328P

- There are 6 PWM outputs available in ATmega328P; where 2 PWM outputs are associated with each of Timer 0, 1 and 2.
  - Associated with Timer 0: OC0A and OC0B
  - Associated with Timer 1: OC1A and OC1B
  - Associated with Timer 2: OC2A and OC2B

- There are 3 different PWM modes available.
  - Fast PWM
  - Phase Correct PWM
  - Phase and Frequency Correct PWM

- The benefit of using the built-in PWM feature is that it gives us the option of programming the time period and duty cycle of the generated PWM signal while relieving the CPU to do other things as well.

- In this Lab, we will use Timer 0 together with Fast PWM and Phase Correct PWM modes.



GUC German University in Cairo

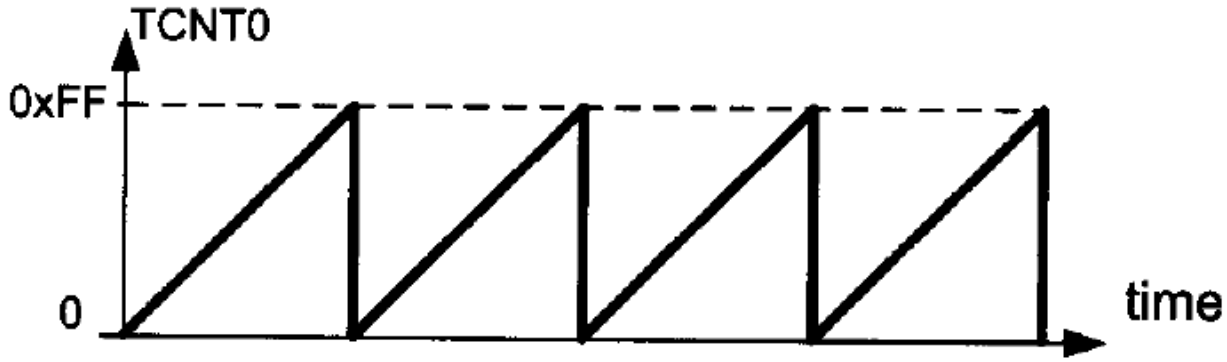
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Tutorial Contents

- Pulse Width Modulation (PWM)
- DC Motors
  - Controlling the Direction : H-Bridge
  - Controlling the Speed: PWM signals
- **PWM module in ATmega328P**
  - **Different Modes**
    - **Fast PWM**
    - Phase Correct PWM
  - Associated Registers
- Analog-to-Digital Converter (ADC) in ATmega328P
- Lab 6 Validation

Fast PWM Mode in ATmega328P

- In the Fast PWM mode, Timer 0 counts up from BOTTOM to TOP normally like it does in the Normal Mode. When it reaches its limit TOP, the timer overflows to BOTTOM and the timer overflow flag TOV0 is set HIGH.



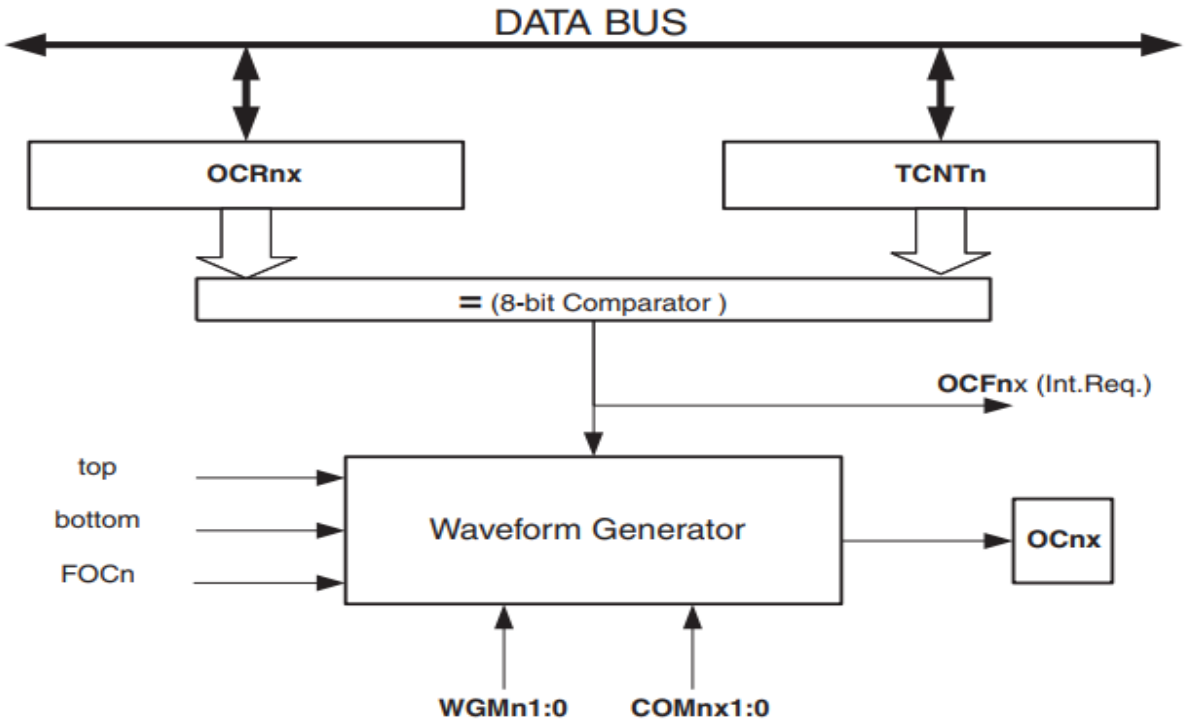
- These are some useful definitions used in the datasheet.

Parameter	Definition
BOTTOM	The counter reaches the BOTTOM when it becomes 0x00.
MAX	The counter reaches its MAXimum when it becomes 0xFF (decimal 255).
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF (MAX) or the value stored in the OCR0A register. The assignment is dependent on the mode of operation.

- The high frequency of the Fast PWM mode makes it well-suited for power regulation, rectification, and DAC applications.

Fast PWM Mode in ATmega328P

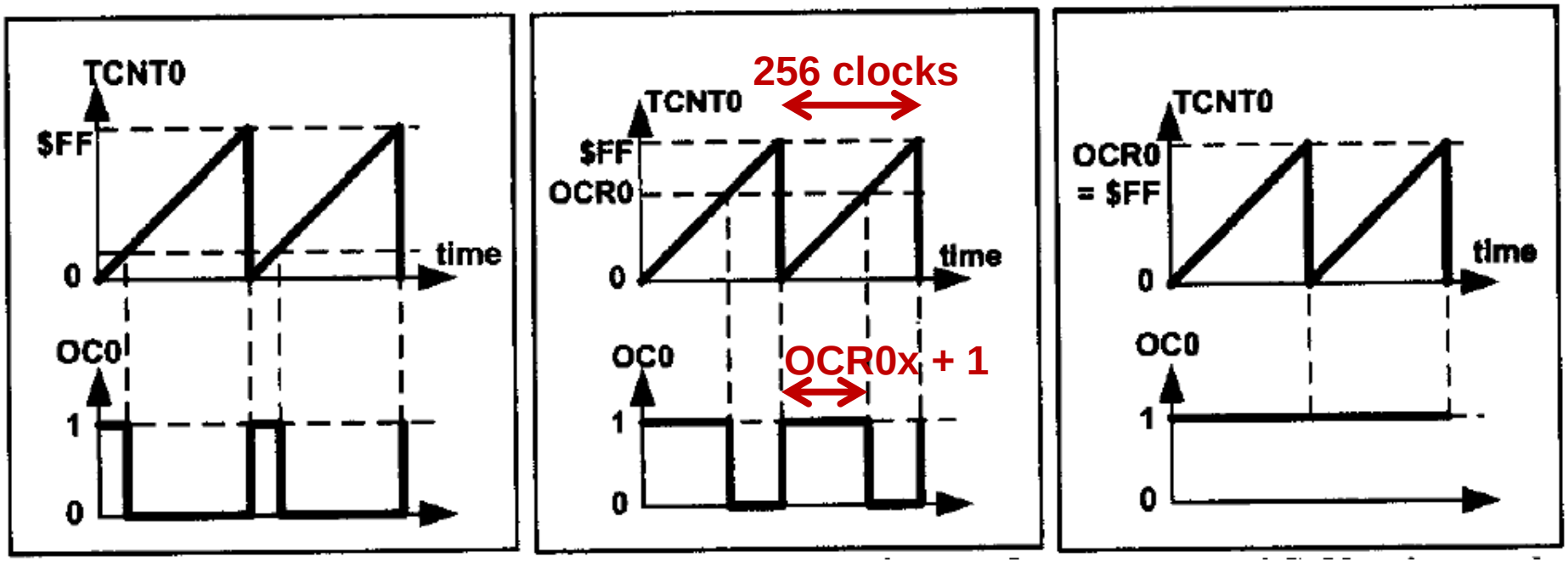
- When a match is detected between TCNT0 and OCR0x, the OCF0x flag is set and signal is send to the Waveform Generator. The Waveform Generator then changes the state of the OC0x pin (the state is determined by the selected mode). When the TCNTn register passes the TOP value (0xFF or OCR0A) it simply overflows (or overruns) back to 0, at the same time the OCF0x flag is set.



Fast PWM Mode in ATmega328P

- There are two available modes – inverted and non-inverted modes. We will use the non-inverted mode.

- Whenever the match between TCNT0 and OCR0x occurs, the Waveform Generator clears OC0x.



Fast PWM Mode in ATmega328P

- Let us deal only with the case where TOP = 0xFF which is the MAX.

- The frequency of the generated PWM signal in the Fast PWM mode is:

$$f_{gen} = \frac{f_{osc}}{256 \times N_{prescaler}}$$

where f_{osc} is the crystal oscillator frequency (16 MHz) and $N_{prescaler}$ is the pre-scaler value chosen.

- In the non-inverted Fast PWM mode, the duty cycle is calculated from:

$$D\% = \frac{OCR0x + 1}{256} \times 100$$

where $D\%$ is the required duty cycle in percent and $OCR0x$ is the register value in the Output Compare Register (OCR0x).

- **Example:** Calculate the value of the OCR0x register to generate a PWM wave of 75% duty cycle in the non-inverted Fast PWM mode.

$$D\% = \frac{OCR0x + 1}{256} \times 100 \rightarrow 75 = \frac{OCR0x + 1}{256} \times 100$$

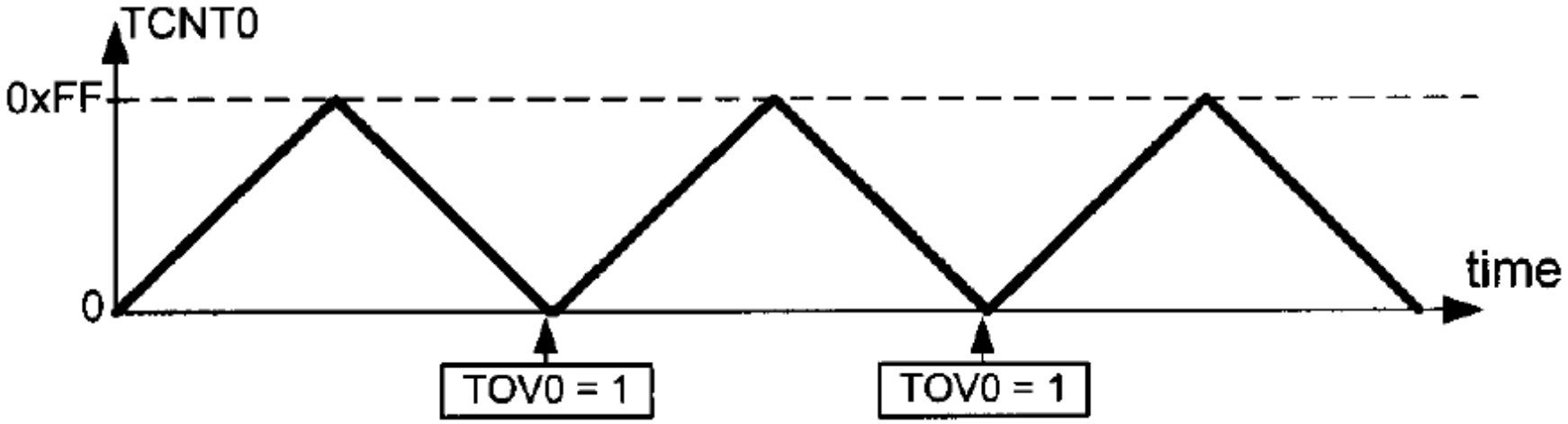
$$\therefore OCR0x = 191$$

Tutorial Contents

- Pulse Width Modulation (PWM)
- DC Motors
  - Controlling the Direction : H-Bridge
  - Controlling the Speed: PWM signals
- **PWM module in ATmega328P**
  - **Different Modes**
    - Fast PWM
    - **Phase Correct PWM**
  - Associated Registers
- Analog-to-Digital Converter (ADC) in ATmega328P
- Lab 6 Validation

Phase Correct PWM Mode in ATmega328P

- In the Phase Correct PWM mode, Timer 0 counts up from BOTTOM to TOP and then counts down till BOTTOM. When Timer 0 counts up and down back to BOTTOM, the timer overflow flag TOV0 is set HIGH.



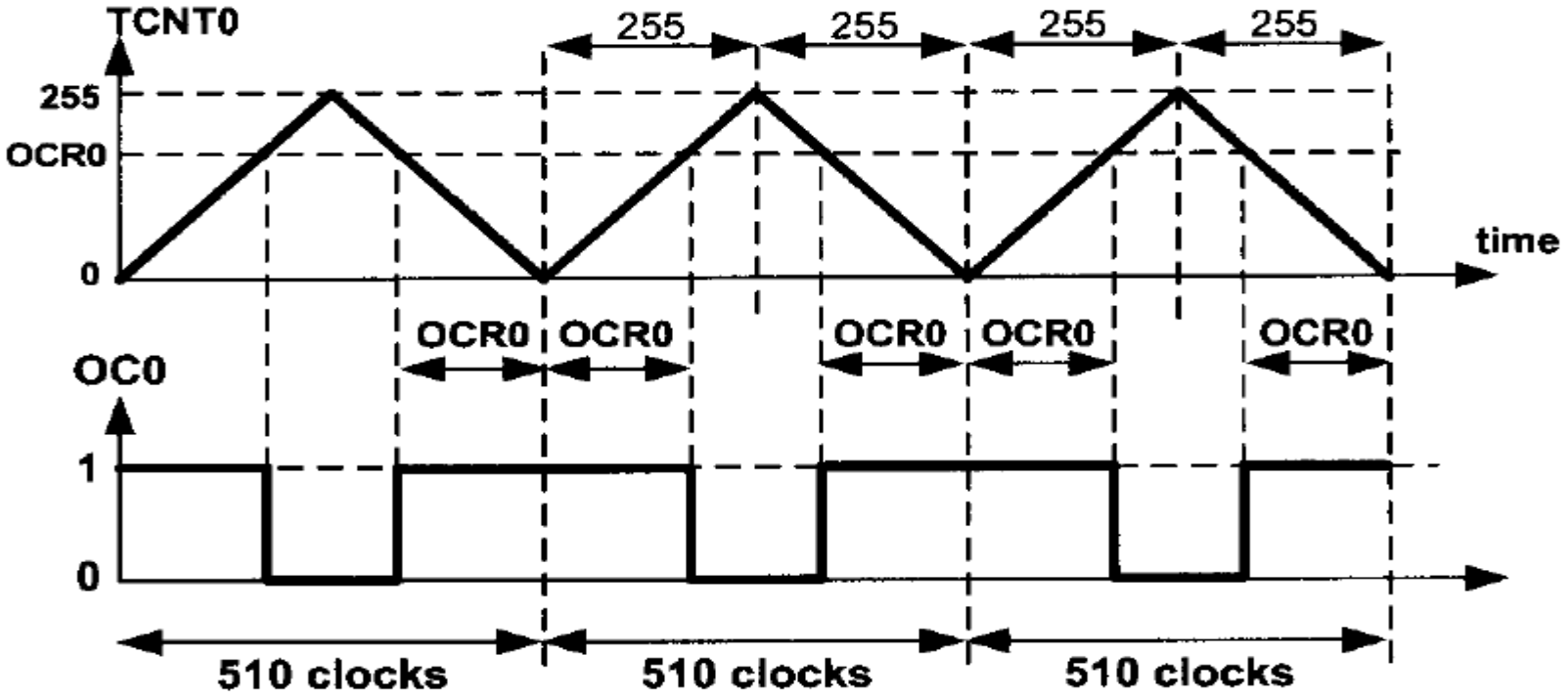
- Due to the double-slope operation, the operating frequency of the Phase Correct PWM mode can be half that of the Fast PWM mode that uses single-slope operation.

- However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

Phase Correct PWM Mode in ATmega328P

- Here as well, there are two available modes – inverted and non-inverted modes. We will use the non-inverted mode.

- Whenever the match between TCNT0 and OCR0x occurs, the Waveform Generator clears OC0x when counting up and sets OC0x when counting down.



Phase Correct PWM Mode in ATmega328P

- Let us deal only with the case where TOP = 0xFF which is the MAX.

- The frequency of the generated PWM signal in the Phase Correct PWM mode is:

$$f_{gen} = \frac{f_{osc}}{510 \times N_{prescaler}}$$

where f_{osc} is the crystal oscillator frequency (16 MHz) and $N_{prescaler}$ is the pre-scaler value chosen.

- In the non-inverted Phase Correct PWM mode, the duty cycle is calculated from:

$$D\% = \frac{OCR0x}{255} \times 100$$

where $D\%$ is the required duty cycle in percent and $OCR0x$ is the register value in the Output Compare Register (OCR0x).

- **Example:** Calculate the value of the OCR0x register to generate a PWM wave of 20% duty cycle in the non-inverted Fast PWM mode.

$$D\% = \frac{OCR0x}{255} \times 100 \rightarrow 20 = \frac{OCR0x}{255} \times 100$$

$$\therefore OCR0x = 51$$

Tutorial Contents

- Pulse Width Modulation (PWM)
- DC Motors
  - Controlling the Direction : H-Bridge
  - Controlling the Speed: PWM signals
- **PWM module in ATmega328P**
  - Different Modes
    - Fast PWM
    - Phase Correct PWM
  - **Associated Registers**
- Analog-to-Digital Converter (ADC) in ATmega328P
- Lab 6 Validation

PWM Module in ATmega328P: Associated Registers

TCCR0A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 1:0 – WGM01:0: Waveform Generation Mode**

Combined with the WGM02 bit found in the TCCR0B register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see Table 14-8. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), clear timer on compare match (CTC) mode, and two types of pulse width modulation (PWM) modes (see Section 14.7 "Modes of Operation" on page 78).

Table 14-8. Waveform Generation Mode Bit Description

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on(1)(2)
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, phase correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, phase correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

PWM Module in ATmega328P: Associated Registers

TCCR0A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 – COM0A1:0: Compare Match Output A Mode**

These bits control the output compare pin (OC0A) behavior. If one or both of the COM0A1:0 bits are set, the OC0A output overrides the normal port functionality of the I/O pin it is connected to. However, note that the data direction register (DDR) bit corresponding to the OC0A pin must be set in order to enable the output driver.

When OC0A is connected to the pin, the function of the COM0A1:0 bits depends on the WGM02:0 bit setting. Table 14-2 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 14-3. Compare Output Mode, Fast PWM Mode(1)

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal port operation, OC0A disconnected. WGM02 = 1: Toggle OC0A on compare match.
1	0	Clear OC0A on compare match, set OC0A at BOTTOM, (non-inverting mode).
1	1	Set OC0A on compare match, clear OC0A at BOTTOM, (inverting mode).

PWM Module in ATmega328P: Associated Registers

TCCR0A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 – COM0A1:0: Compare Match Output A Mode**

These bits control the output compare pin (OC0A) behavior. If one or both of the COM0A1:0 bits are set, the OC0A output overrides the normal port functionality of the I/O pin it is connected to. However, note that the data direction register (DDR) bit corresponding to the OC0A pin must be set in order to enable the output driver.

When OC0A is connected to the pin, the function of the COM0A1:0 bits depends on the WGM02:0 bit setting. Table 14-2 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 14-4. Compare Output Mode, Phase Correct PWM Mode(1)

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal port operation, OC0A disconnected. WGM02 = 1: Toggle OC0A on compare match.
1	0	Clear OC0A on compare match when up-counting. Set OC0A on compare match when down-counting.
1	1	Set OC0A on compare match when up-counting. Clear OC0A on compare match when down-counting.

PWM Module in ATmega328P: Associated Registers

TCCR0B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 2:0 – CS02:0: Clock Select**

The three clock select bits select the clock source to be used by the Timer/Counter.

Table 14-9. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(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 T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

PWM Module in ATmega328P: Associated Registers

TCNT0 – Timer/Counter Register

Bit	7	6	5	4	3	2	1	0	
0x26 (0x46)	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Timer/Counter register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNT0 register blocks (removes) the compare match on the following timer clock. Modifying the counter (TCNT0) while the counter is running, introduces a risk of missing a compare match between TCNT0 and the OCR0x registers.

OCR0A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
0x27 (0x47)	OCR0A[7:0]								OCR0A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The output compare register A contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC0A pin.

PWM Module in ATmega328P: Associated Registers

TIMSK0 – Timer/Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6E)	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – OCIE0A: Timer/Counter0 Output Compare Match A Interrupt Enable**

When the OCIE0A bit is written to one, and the I-bit in the status register is set, the Timer/Counter0 compare match A interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter0 occurs, i.e., when the OCF0A bit is set in the Timer/Counter 0 interrupt flag register – TIFR0.

TIFR0 – Timer/Counter 0 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x15 (0x35)	–	–	–	–	–	OCF0B	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – OCF0A: Timer/Counter 0 Output Compare A Match Flag**

The OCF0A bit is set when a compare match occurs between the Timer/Counter0 and the data in OCR0A – output compare register0. OCF0A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0A is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0A (Timer/Counter0 compare match interrupt enable), and OCF0A are set, the Timer/Counter0 compare match interrupt is executed.

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Tutorial Contents

- Pulse Width Modulation (PWM)
- DC Motors
  - Controlling the Direction : H-Bridge
  - Controlling the Speed: PWM signals
- PWM module in ATmega328P
  - Different Modes
    - Fast PWM
    - Phase Correct PWM
  - Associated Registers
- **Analog-to-Digital Converter (ADC) in ATmega328P**
- Lab 6 Validation

Analog-to-Digital Converter (ADC) in ATmega328P

Code to initialize the ADC in the ATmega328P:

```
void init_ADC()
{
    ADCSRA |= 0b00000111; //Set prescalar to 128
    DIDR0  |= 0b11111111; //disable digital input on analog input channels
    ADCSRA |= 0b10000000; //Enable the ADC module (set ADEN)
}
```

Code to read analog signal value:

```
unsigned int read_adc(unsigned char channel)
{
    ADMUX = (ADMUX & 0b11110000) | (channel & 0b00001111); //Set the Mux to select the ADC channel
    ADCSRA |= 0b01000000; // Start conversion (set ADSC bit)
    while(ADCSRA & 0b01000000);// wait for conversion
    return ADC; //return the 10-bit value
}
```

Tutorial Contents

- Pulse Width Modulation (PWM)
- DC Motors
  - Controlling the Direction : H-Bridge
  - Controlling the Speed: PWM signals
- PWM module in ATmega328P
  - Different Modes
    - Fast PWM
    - Phase Correct PWM
  - Associated Registers
- Analog-to-Digital Converter (ADC) in ATmega328P
- **Lab 6 Validation**

Pulse-Width Modulation (PWM) in ATmega328P

Lab 6 Validation: (to be submitted in your Lab next week)

It is required to build a circuit with 2 push buttons, a potentiometer and a DC motor. The push buttons control the motor direction and the potentiometer controls the motor speed.

Circuit Operation:

- When the user changes the potentiometer, the speed of the motor changes accordingly. For example, when the potentiometer is on maximum resistance (i.e. zero input voltage), motor stops. When it is on minimum resistance (i.e. maximum voltage), motor is at full speed.
- When the user presses the 1st push button, the motor rotates in one direction. When the user pressed the 2nd push button, the motor rotates in the opposite one.

Design Requirements:

- Use AVR Embedded board.
- Use Embedded C for programming.
- The 2 push buttons are connected to PD0-PD1.
- The potentiometer is connected to ADC0.
- Include a reset push button in your hardware implementation.
- Do not use pre-defined ADC-related or PWM-related functions.

Pulse-Width Modulation (PWM) in ATmega328P

Lab 6 Validation: (to be submitted in your Lab next week)

Here is the pseudo-code of the software approach to be used.

1. Input-Output Configuration:
    a) Configure PD0-PD1 as inputs for the push buttons.
    b) Configure PD2-PD3 as outputs to define the motor direction for H-Bridge.
    c) Configure OCA0 which corresponds to PD6 as an output for PWM.

2. Initialize the on-chip ADC .

3. Phase Correct PWM Initialization:
    a) Set the bits WGM00-WGM01= 01 in the TCCR0A register in order to select the Phase Correct PWM mode of operation with TOP = 0xFF.
    b) Set bits COM0A0-COM0A1 = 10 in the TCCR0A register in order to use the non-inverting mode.
    c) We will set the PWM frequency to 4 kHz. So, we need a pre-scaler of 8.

$$f_{gen} = \frac{f_{osc}}{510 \times N_{prescaler}} = 4000 \rightarrow N_{prescaler} \approx 8$$

So, set bits CS02-00 = 010 to set 8 as pre-scaler.

Pulse-Width Modulation (PWM) in ATmega328P

Lab 6 Validation: (to be submitted in your Lab next week)

4. Select the motor direction (PD2 and PD3) according to which push button is pressed.

5. Read the potentiometer analog input through the ADC (as done in Lab 5).

6. Set the OCR0A register = potentiometer value/4 in order to set the required duty cycle of the PWM signal.

   255/1023 ≈ 1/4

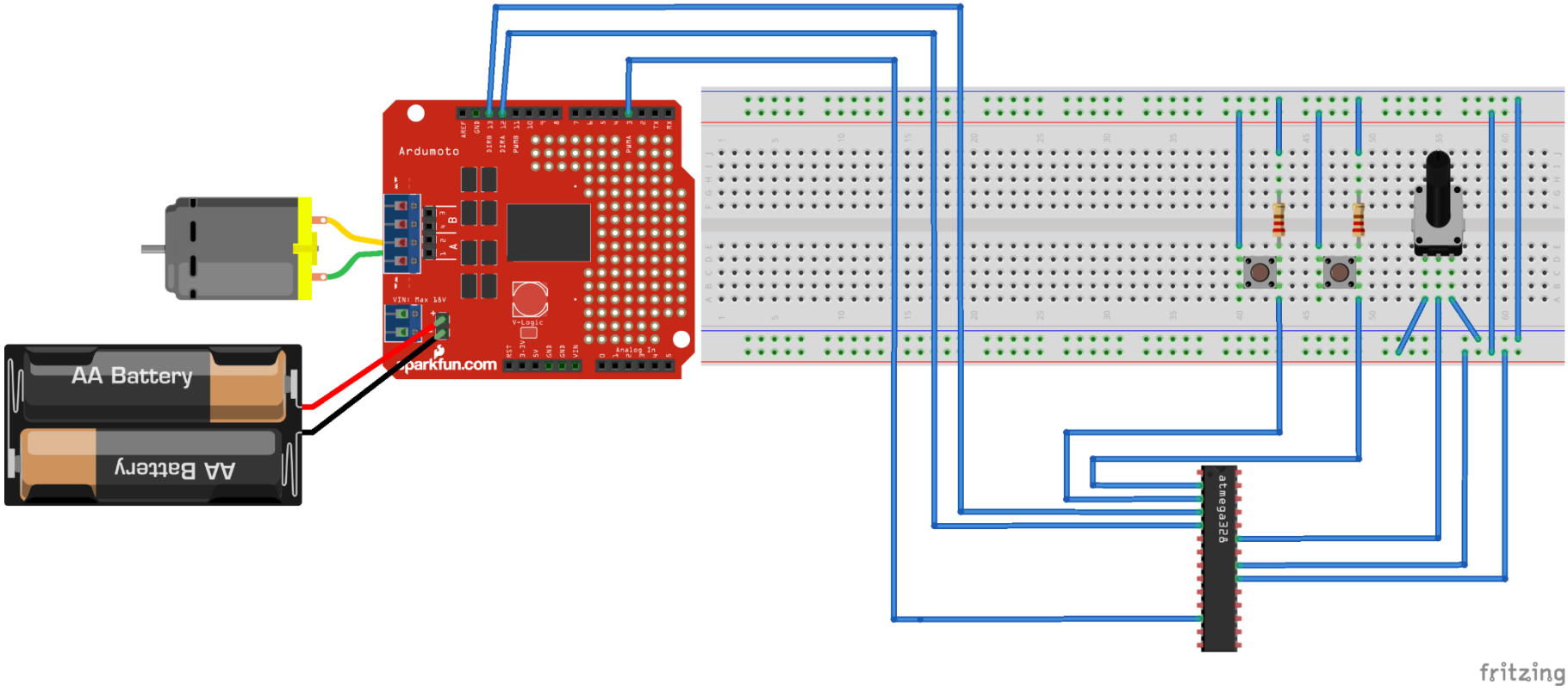
7. Go to step 4 and repeat.

Recommended: Run the Phase Correct PWM mode using Timer 1.

Pulse-Width Modulation (PWM) in ATmega328P

Lab 6 Validation: (to be submitted in your Lab next week)

Here are the hardware connections.



GUC German University in Cairo

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science