PWM Interfacing on eYFi-Mega Board

e-Yantra Team

Embedded Real-Time Systems Lab
Indian Institute of Technology-Bombay

IIT Bombay
July 4, 2022

Agenda for Discussion

1. Introduction
   - Pulse Width Modulation
   - Duty Cycle
   - Timers in AVR
2. PWM Generation in AVR
   - Timer/Counter (TCNTn)
   - Output Compare Register
   - PWM Signal
   - Required Functions

Pulse Width Modulation

Pulse Width Modulation

1. Pulse Width Modulation (PWM), is a method of transmitting information on a series of pulses

Pulse Width Modulation

1. Pulse Width Modulation (PWM), is a method of transmitting information on a series of pulses
2. The data that is being transmitted is encoded on the width of these pulses to control the amount of power being sent to a load

Pulse Width Modulation

1. Pulse Width Modulation (PWM), is a method of transmitting information on a series of pulses
2. The data that is being transmitted is encoded on the width of these pulses to control the amount of power being sent to a load
3. Examples: Electric stoves, Lamp dimmers, and Robotic Servos

Pulse Width Modulation

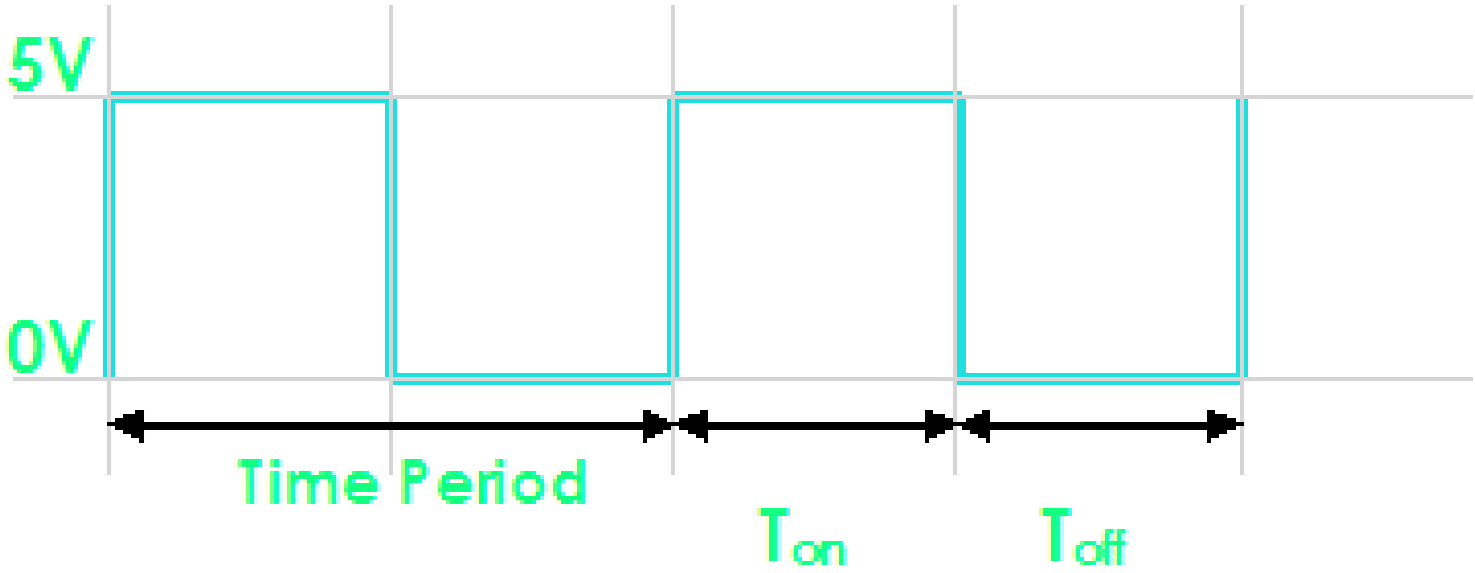
1. Pulse Width Modulation (PWM), is a method of transmitting information on a series of pulses
2. The data that is being transmitted is encoded on the width of these pulses to control the amount of power being sent to a load
3. Examples: Electric stoves, Lamp dimmers, and Robotic Servos

Pulse Width Modulation

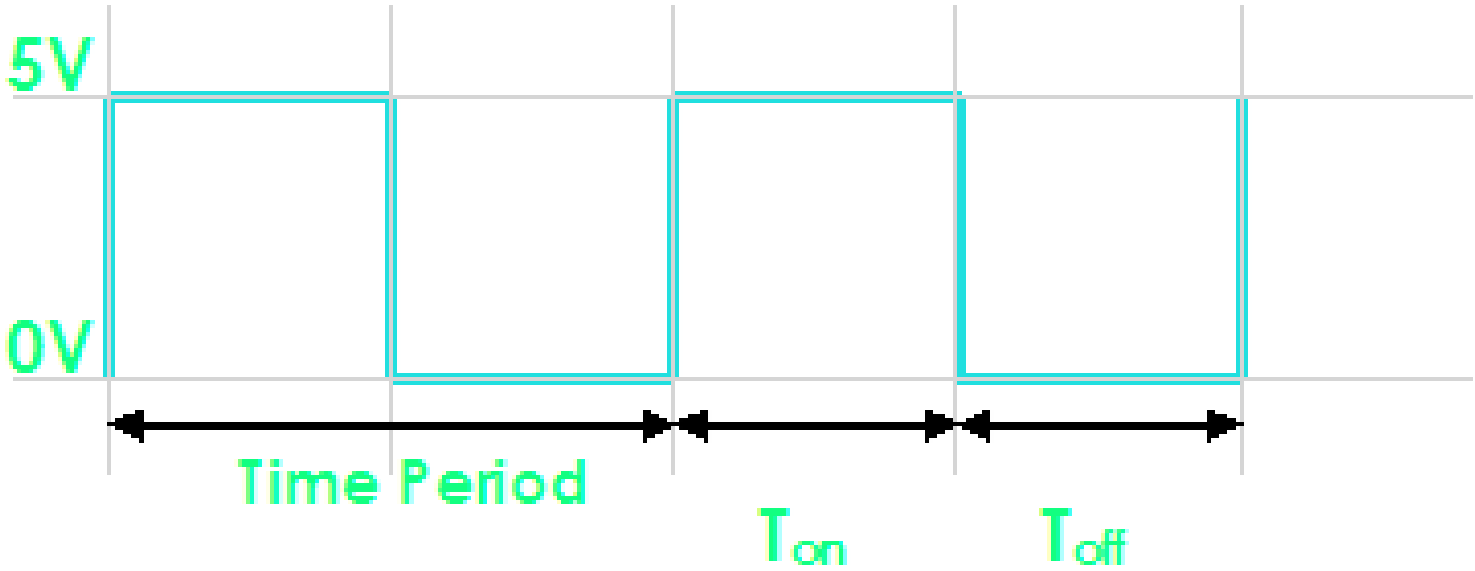
1. Pulse Width Modulation (PWM), is a method of transmitting information on a series of pulses
2. The data that is being transmitted is encoded on the width of these pulses to control the amount of power being sent to a load
3. Examples: Electric stoves, Lamp dimmers, and Robotic Servos

Duty Cycle

Duty Cycle

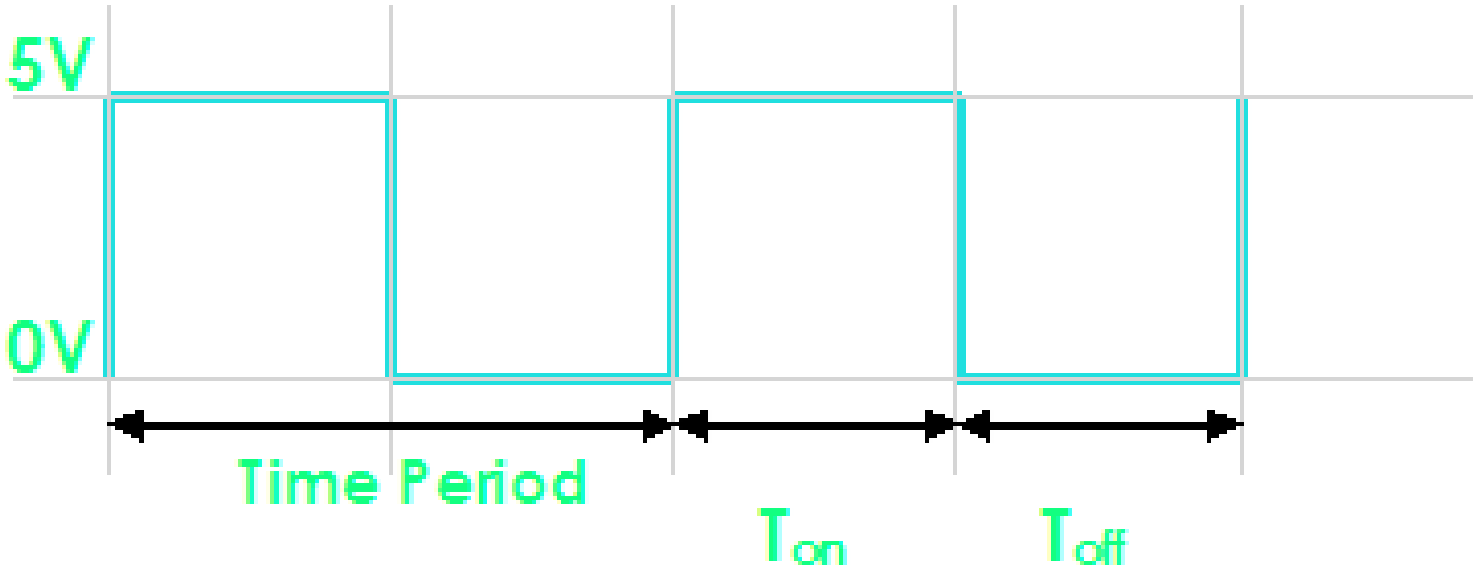


Duty Cycle



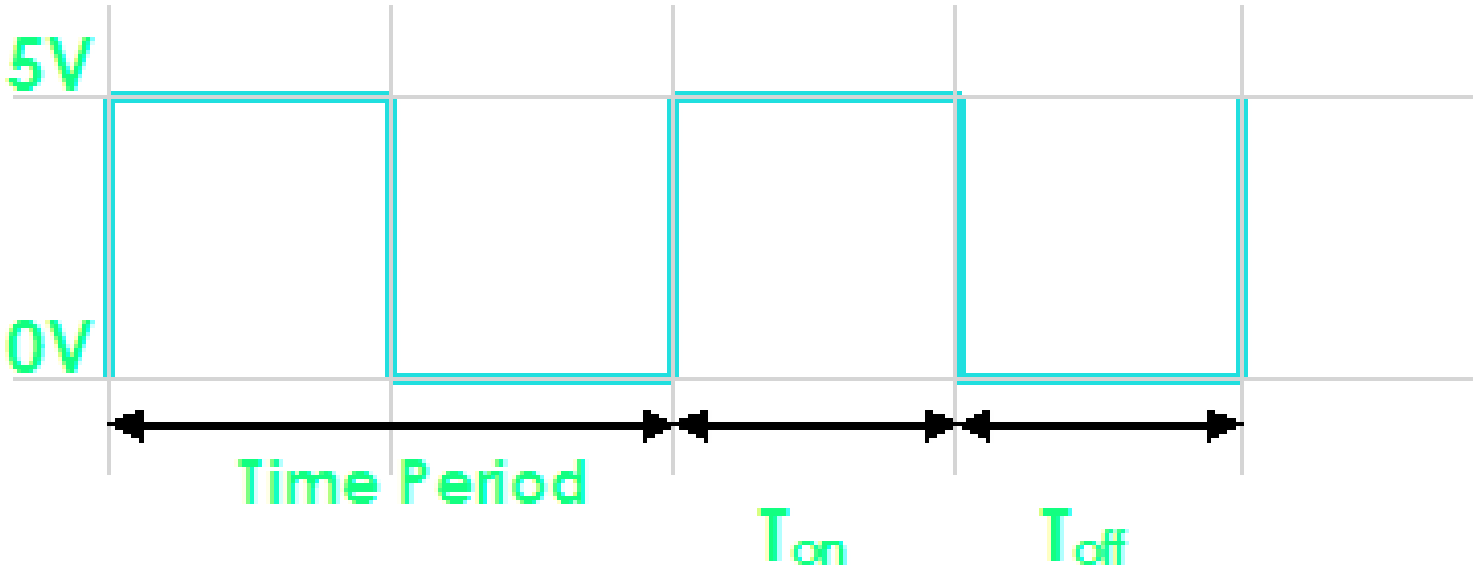
- The signal remains "ON" for some time and "OFF" for some time.

Duty Cycle



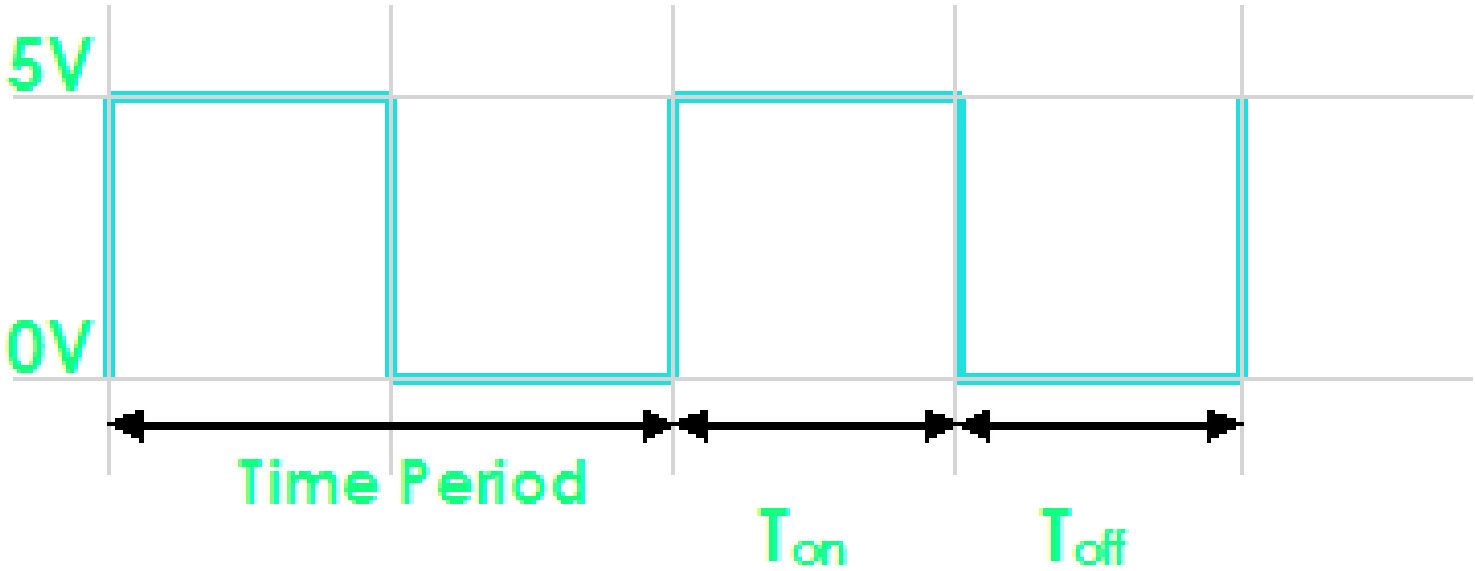
- The signal remains "ON" for some time and "OFF" for some time.
- Ton = Time the output remains high.

Duty Cycle



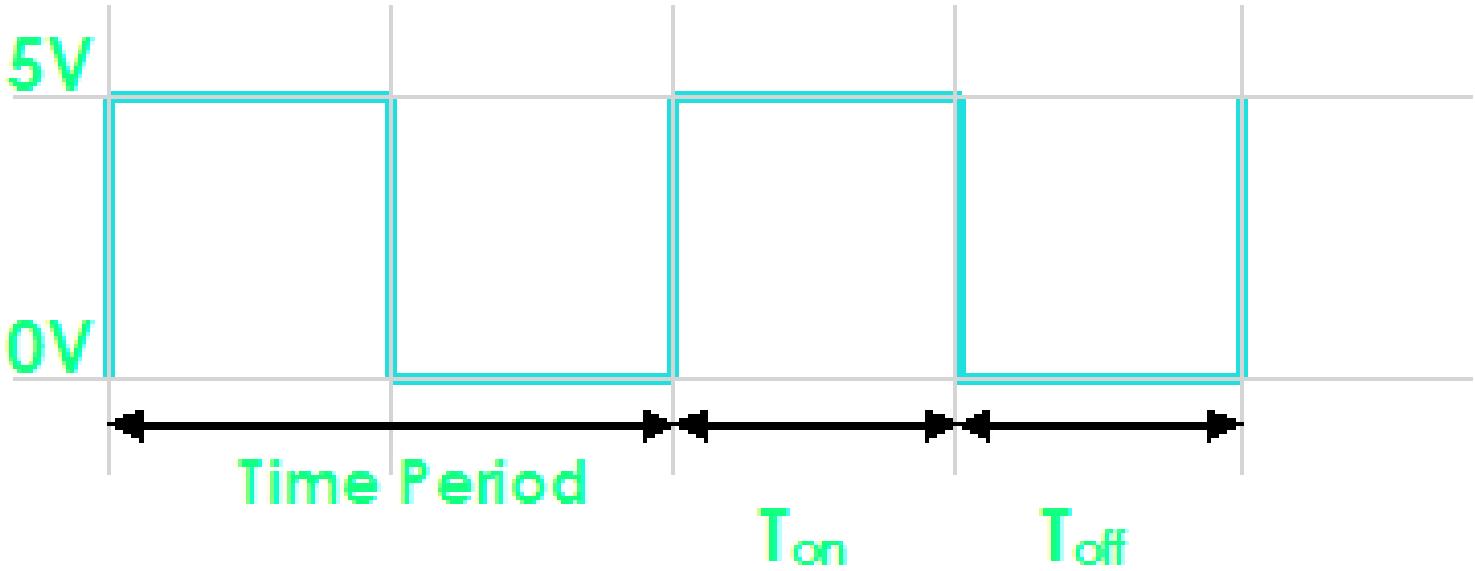
- The signal remains "ON" for some time and "OFF" for some time.
- Ton = Time the output remains high.
- Toff = Time the output remains Low.

Duty Cycle



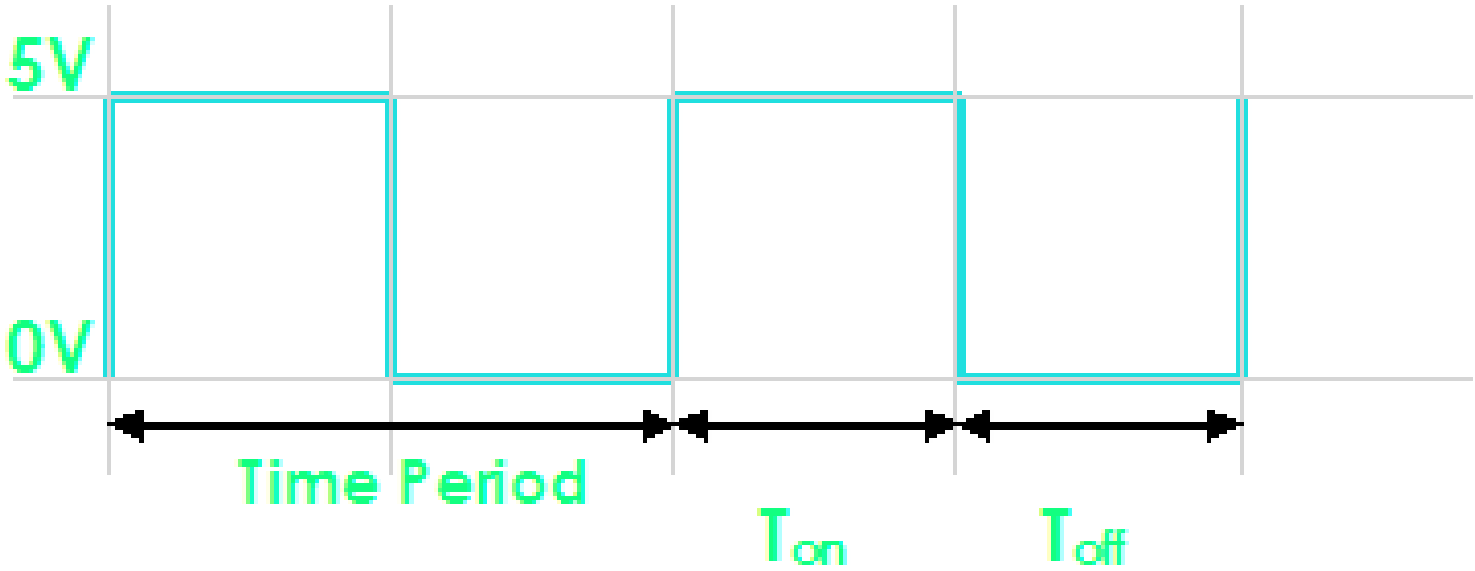
- The signal remains "ON" for some time and "OFF" for some time.
- Ton = Time the output remains high.
- Toff = Time the output remains Low.
- When output is high the voltage is 5v

Duty Cycle



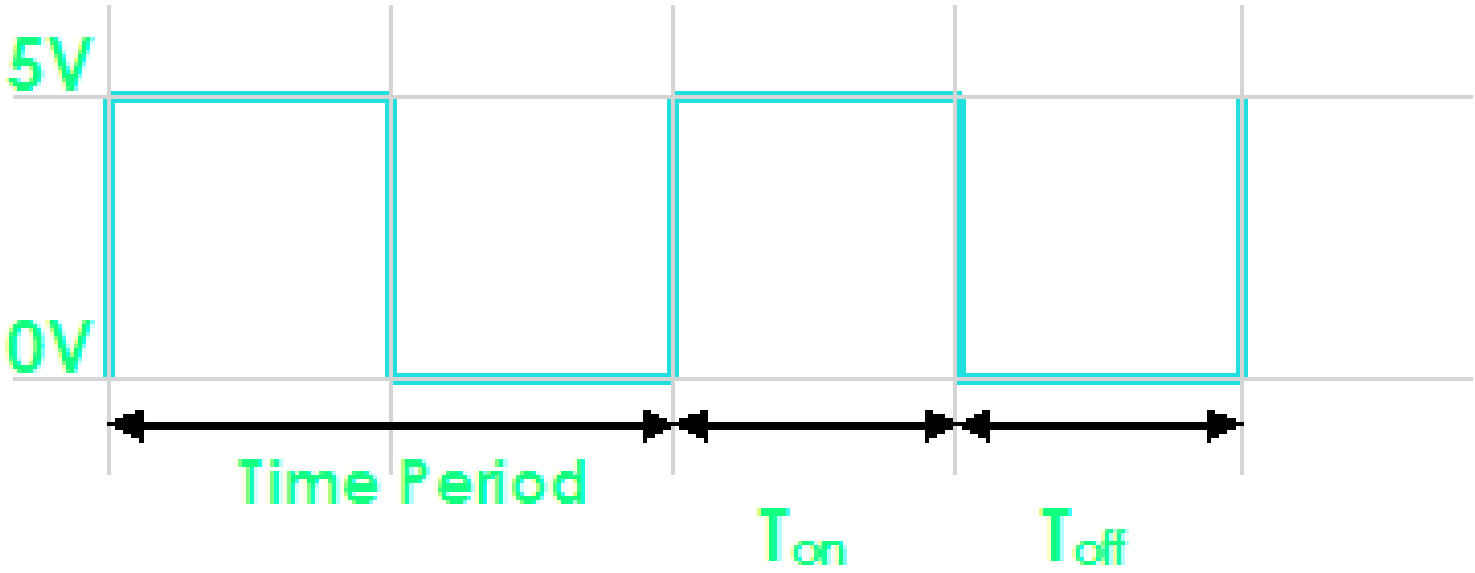
- The signal remains "ON" for some time and "OFF" for some time.
- Ton = Time the output remains high.
- Toff = Time the output remains Low.
- When output is high the voltage is 5v
- When output is low the voltage is 0v

Duty Cycle



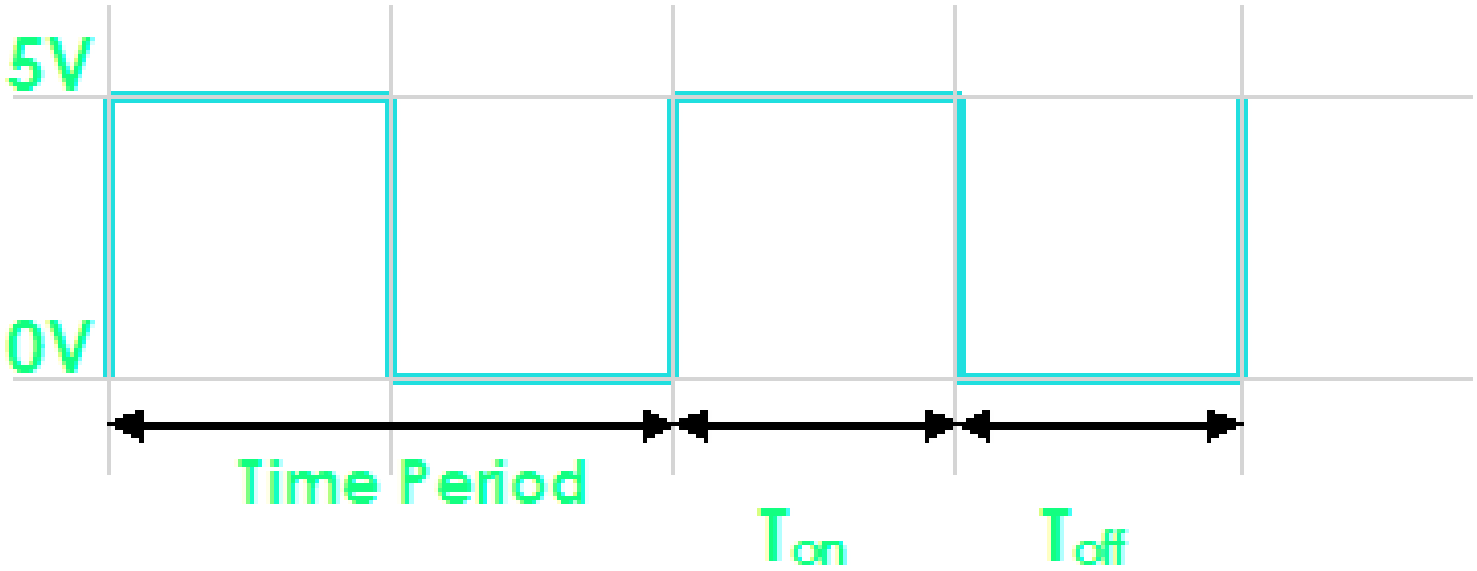
- The signal remains "ON" for some time and "OFF" for some time.
- Ton = Time the output remains high.
- Toff = Time the output remains Low.
- When output is high the voltage is 5v
- When output is low the voltage is 0v
- Time Period(T) = Ton + Toff

Duty Cycle



- The signal remains "ON" for some time and "OFF" for some time.
- Ton = Time the output remains high.
- Toff = Time the output remains Low.
- When output is high the voltage is 5v
- When output is low the voltage is 0v
- Time Period(T) = Ton + Toff
- Duty Cycle = Ton*100/(Ton + Toff)

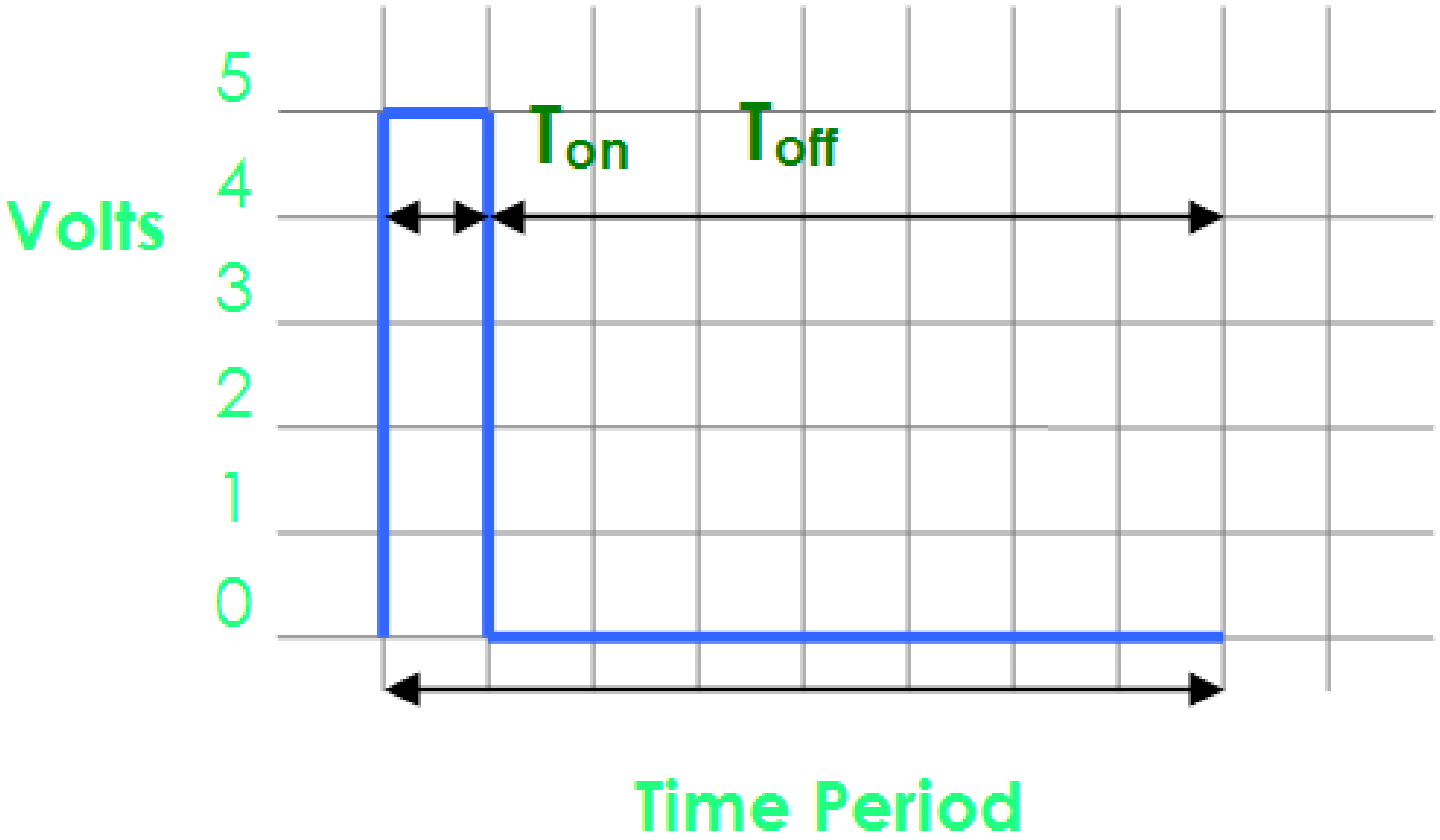
Duty Cycle



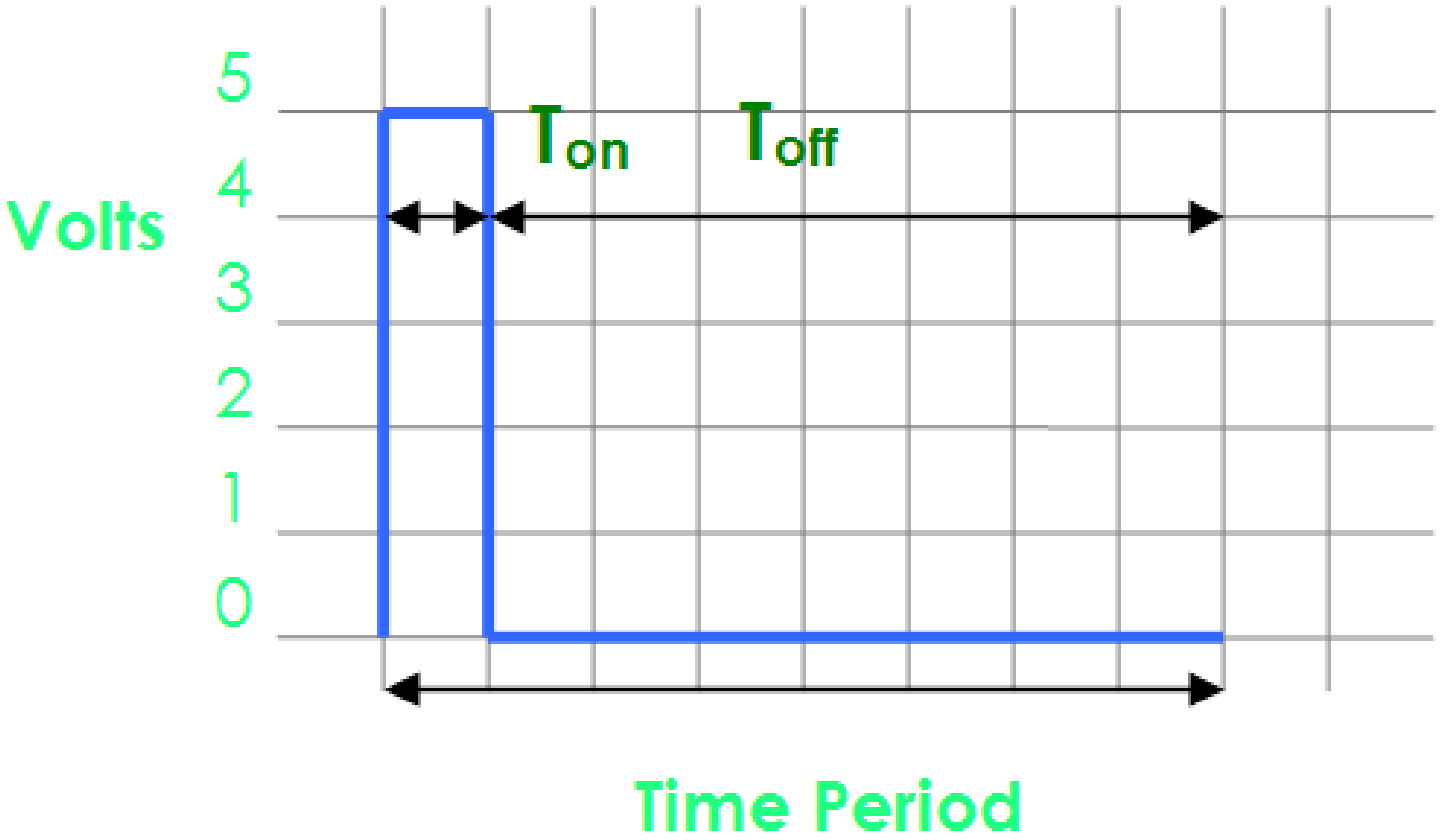
- The signal remains "ON" for some time and "OFF" for some time.
- Ton = Time the output remains high.
- Toff = Time the output remains Low.
- When output is high the voltage is 5v
- When output is low the voltage is 0v
- Time Period(T) = Ton + Toff
- Duty Cycle = Ton*100/(Ton + Toff)
- Duty Cycle = 50%

Duty Cycle (Contd..)

Duty Cycle (Contd..)

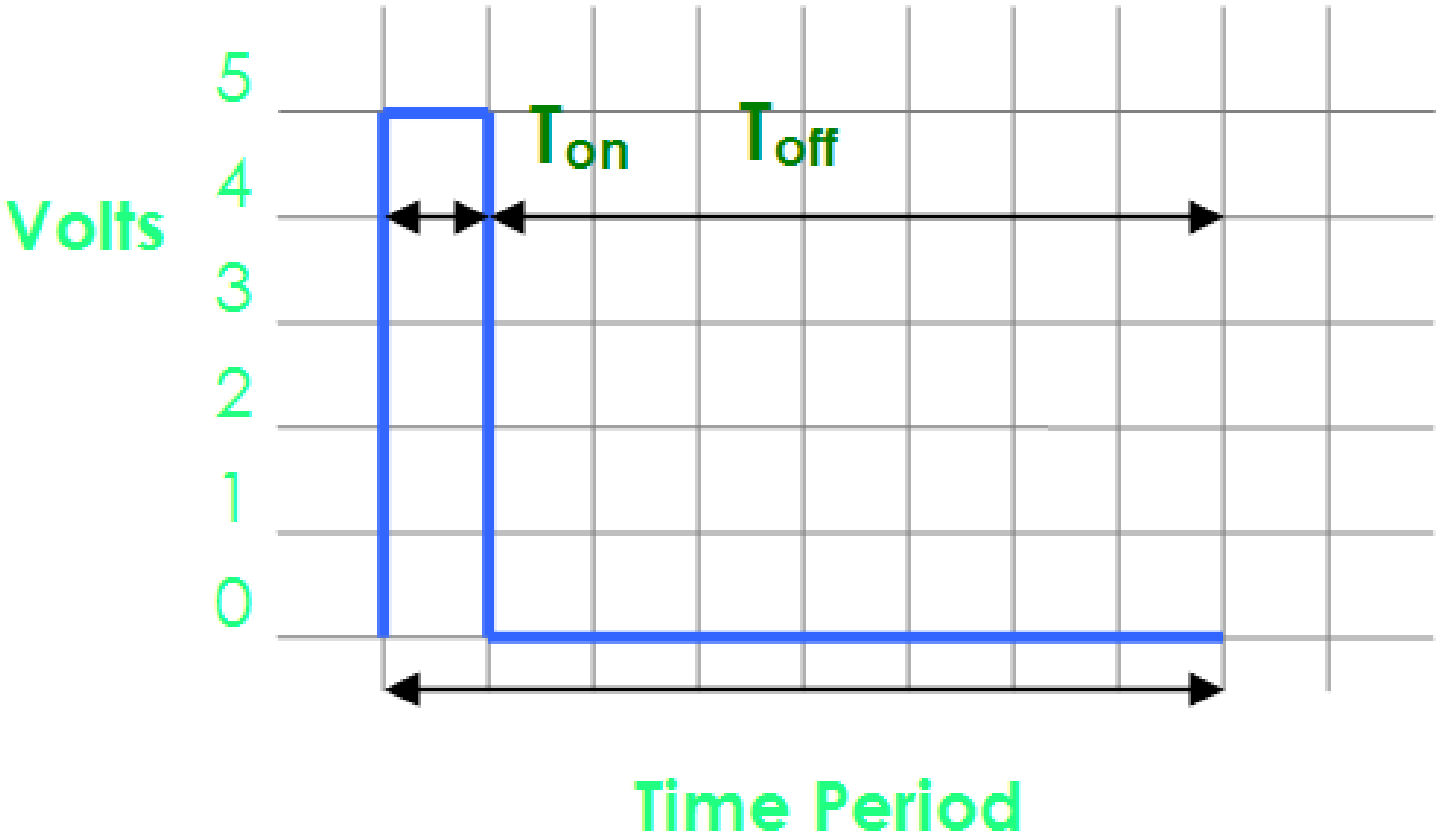


Duty Cycle (Contd..)



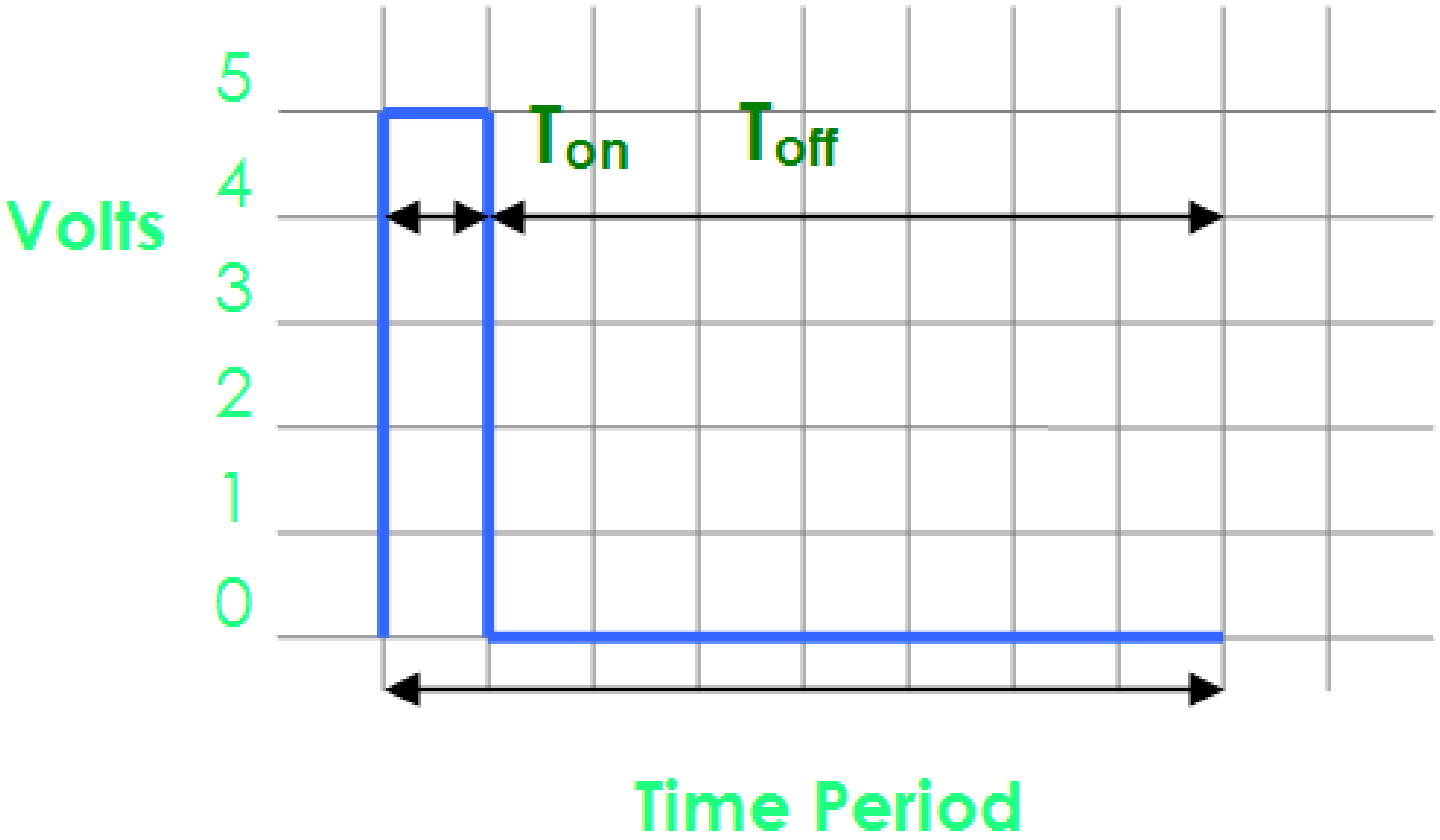
- Ton = Time the output remains high = 1

Duty Cycle (Contd..)



- Ton = Time the output remains high = 1
- Toff = Time the output remains Low = 7

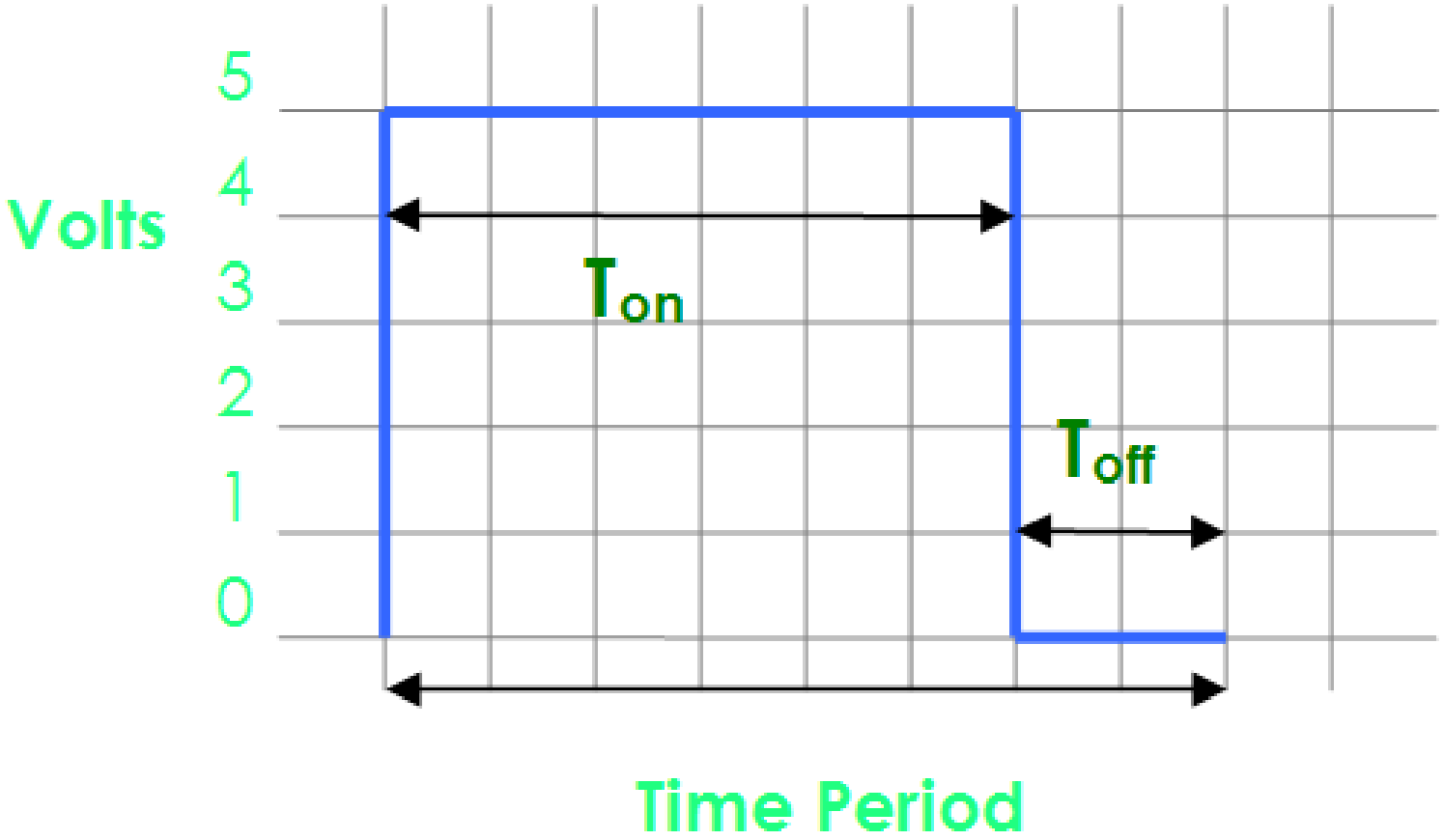
Duty Cycle (Contd..)



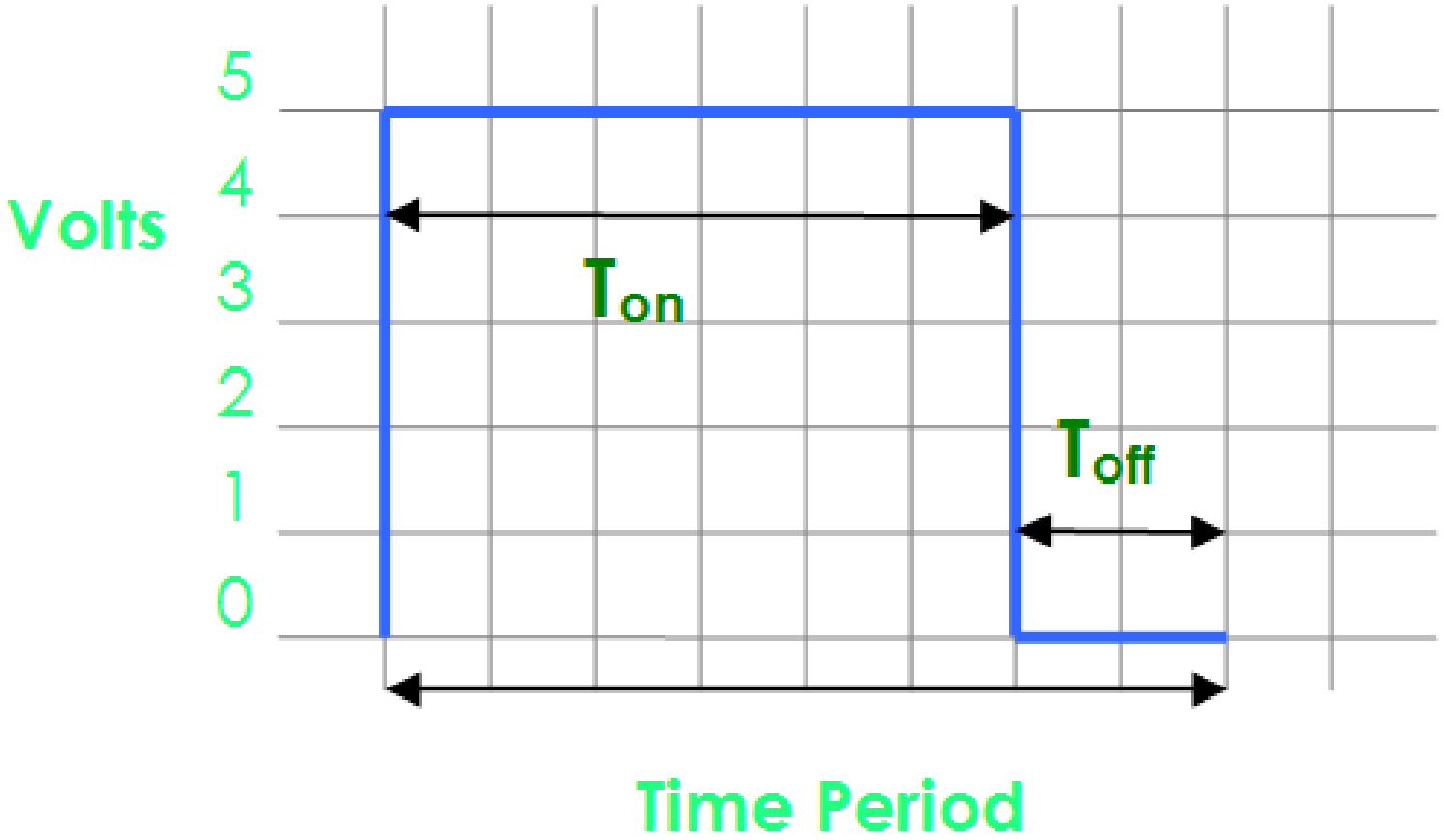
- Ton = Time the output remains high = 1
- Toff = Time the output remains Low = 7
- Duty Cycle = 12.5%

Duty Cycle (Contd..)

Duty Cycle (Contd..)

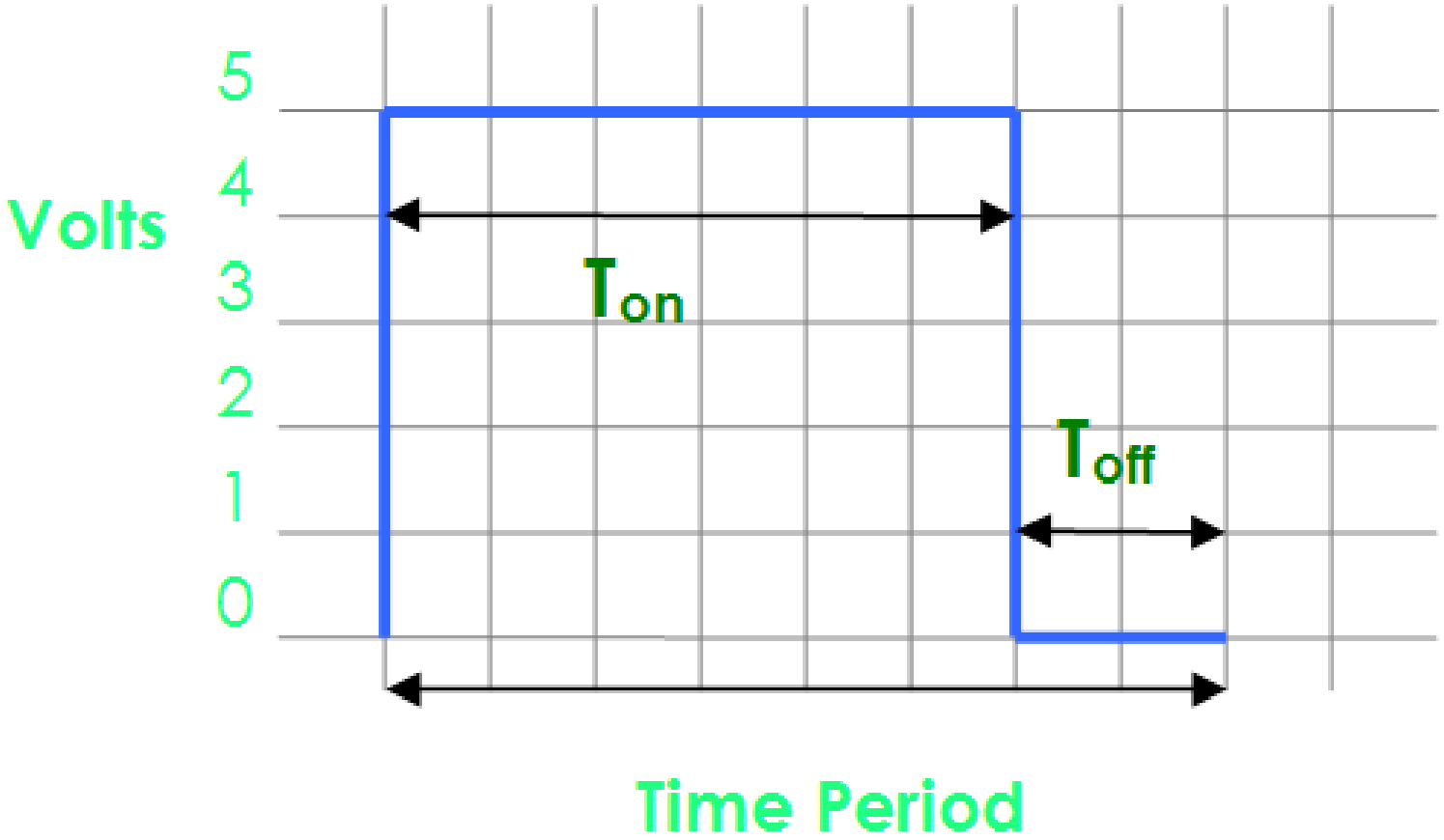


Duty Cycle (Contd..)



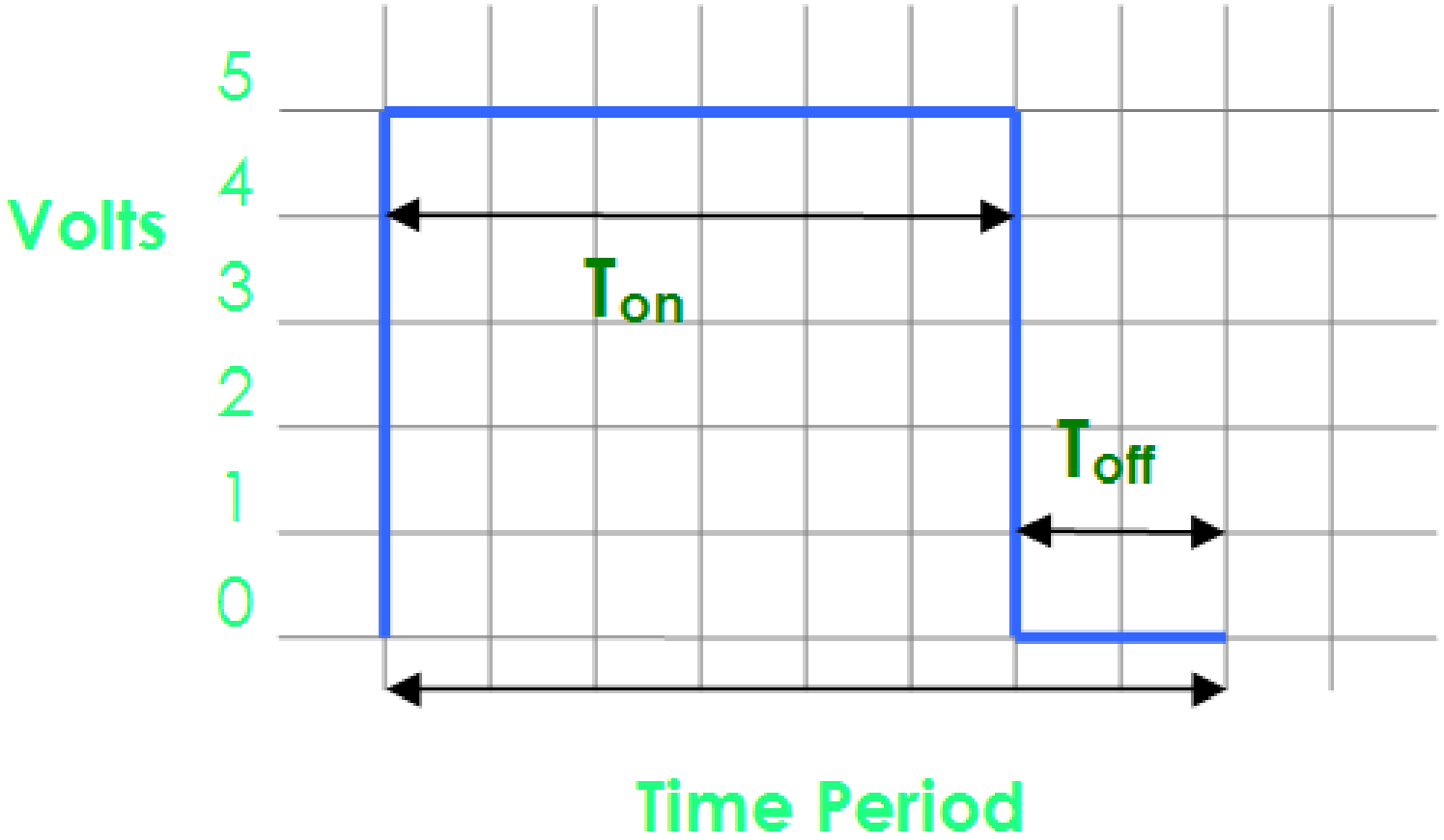
- Ton = Time the output remains High = 6

Duty Cycle (Contd..)



- Ton = Time the output remains High = 6
- Toff = Time the output remains Low = 2

Duty Cycle (Contd..)



- Ton = Time the output remains High = 6
- Toff = Time the output remains Low = 2
- Duty Cycle = 75%

Timers in AVR

Timers in AVR

1. The AVR microcontroller ATmega2560 has

Timers in AVR

1. The AVR microcontroller ATmega2560 has
   - Two 8-bit timers (Timer 0 and Timer 2) and

Timers in AVR

1. The AVR microcontroller ATmega2560 has
   - Two 8-bit timers (Timer 0 and Timer 2) and
   - Four 16-bit timers (Timer 1, 3, 4 and 5)

Timers in AVR

1. The AVR microcontroller ATmega2560 has
   - Two 8-bit timers (Timer 0 and Timer 2) and
   - Four 16-bit timers (Timer 1, 3, 4 and 5)
2. Timers have different channels

Timers in AVR

1. The AVR microcontroller ATmega2560 has
   - Two 8-bit timers (Timer 0 and Timer 2) and
   - Four 16-bit timers (Timer 1, 3, 4 and 5)
2. Timers have different channels
   - Each 8-bit timer has 2 channels and

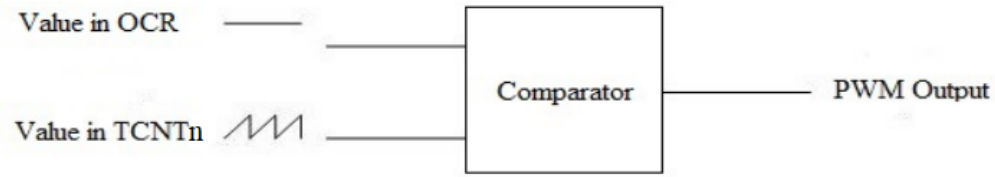
Timers in AVR

1. The AVR microcontroller ATmega2560 has
   - Two 8-bit timers (Timer 0 and Timer 2) and
   - Four 16-bit timers (Timer 1, 3, 4 and 5)
2. Timers have different channels
   - Each 8-bit timer has 2 channels and
   - Each 16-bit timer has 3 channels

PWM Generation in AVR

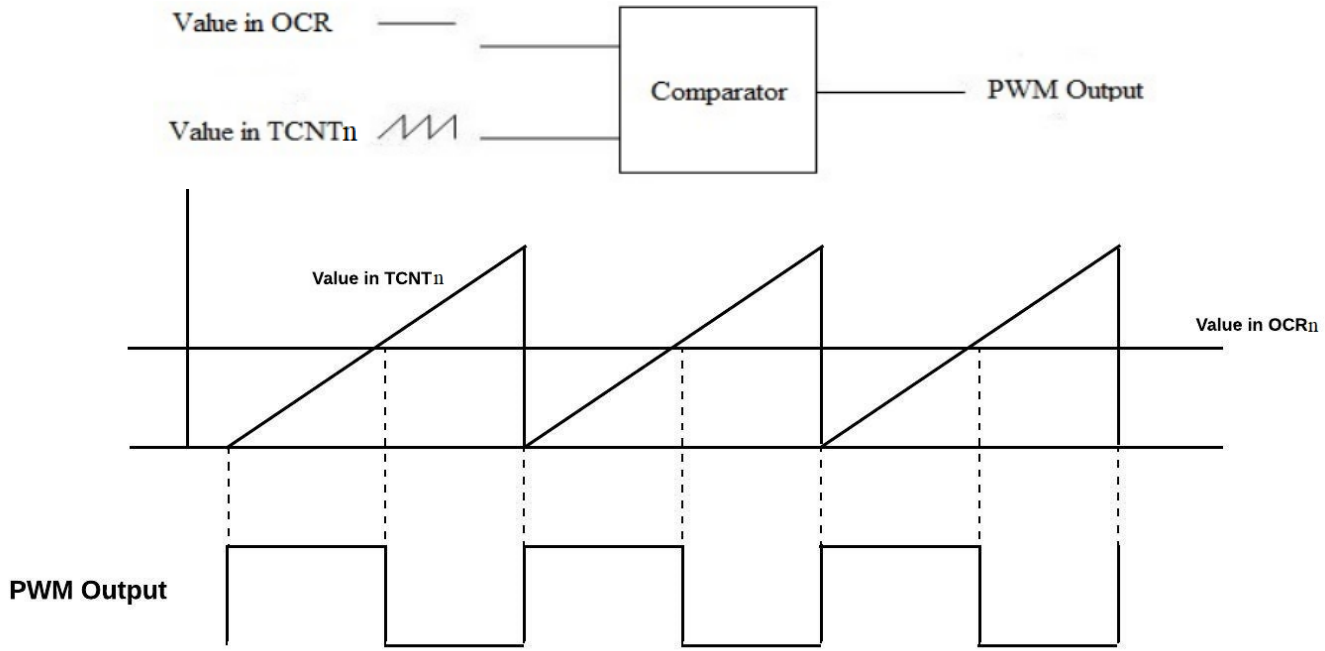
PWM Generation in AVR

Pulse width waveform generation:



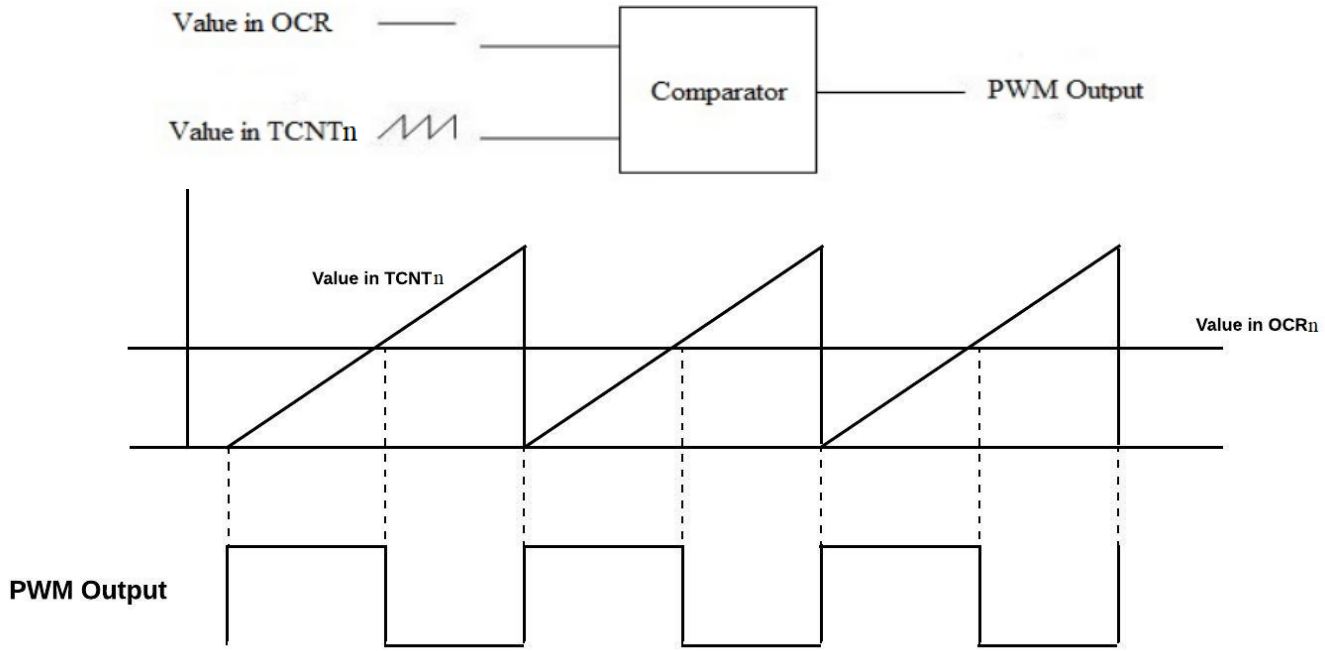
PWM Generation in AVR

Pulse width waveform generation:



PWM Generation in AVR

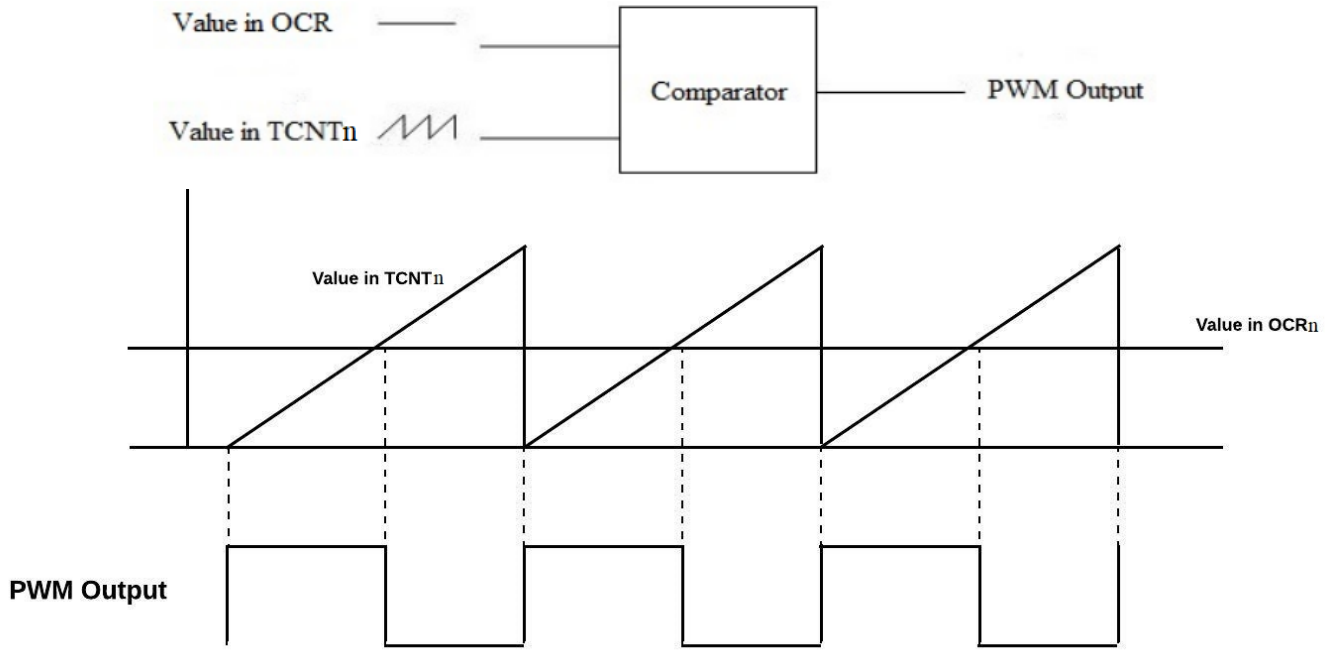
Pulse width waveform generation:



Its generation involves the use of following registers:

PWM Generation in AVR

Pulse width waveform generation:

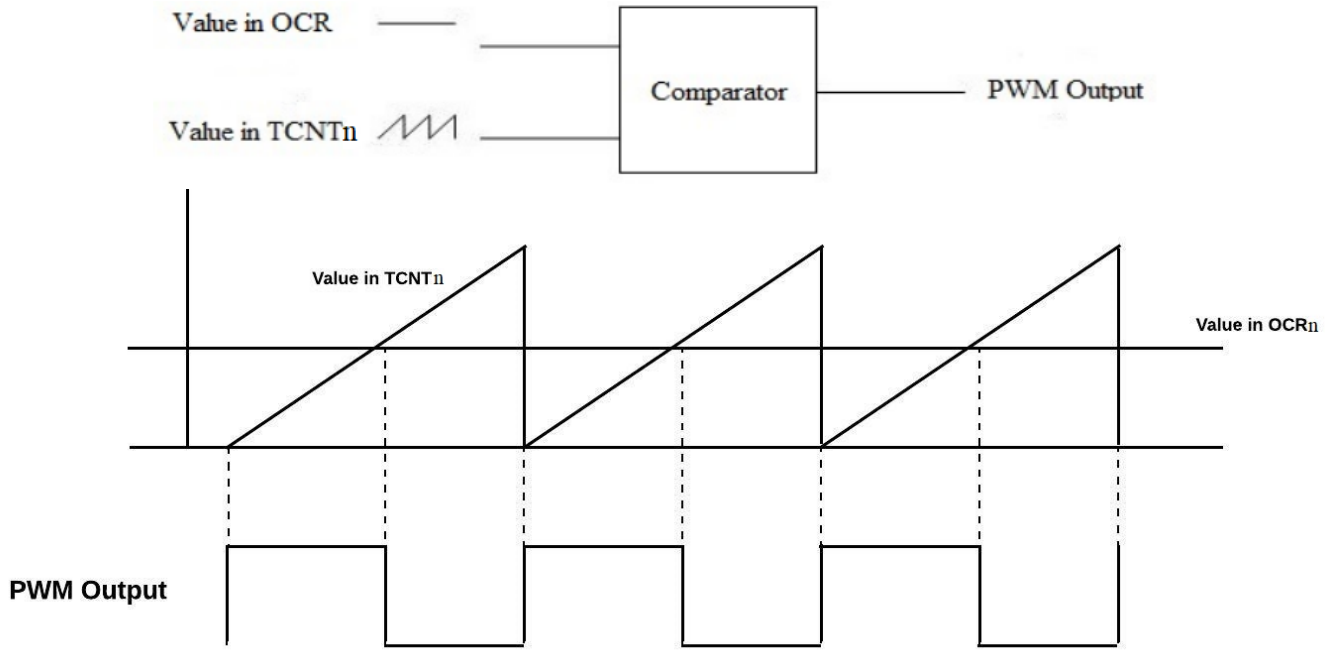


Its generation involves the use of following registers:

- Timer/Counter register n (TCNTn)

PWM Generation in AVR

Pulse width waveform generation:

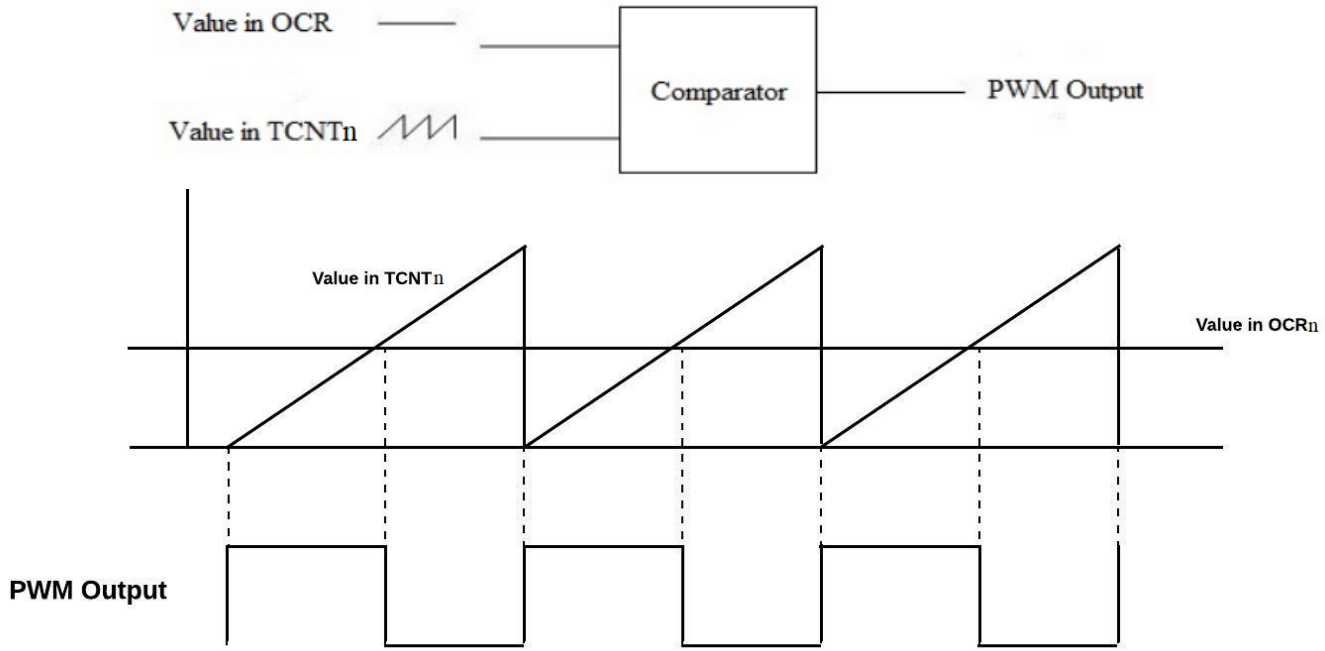


Its generation involves the use of following registers:

- Timer/Counter register n (TCNTn)
- Output Compare register (OCRnA, OCRnB and/or OCRnC)

PWM Generation in AVR

Pulse width waveform generation:



Its generation involves the use of following registers:

- Timer/Counter register n (TCNTn)
- Output Compare register (OCRnA, OCRnB and/or OCRnC)
- Timer/Counter Control registers (TCCRnA and TCCRnB)

Timer/Counter (TCNTn)

Timer/Counter (TCNTn)

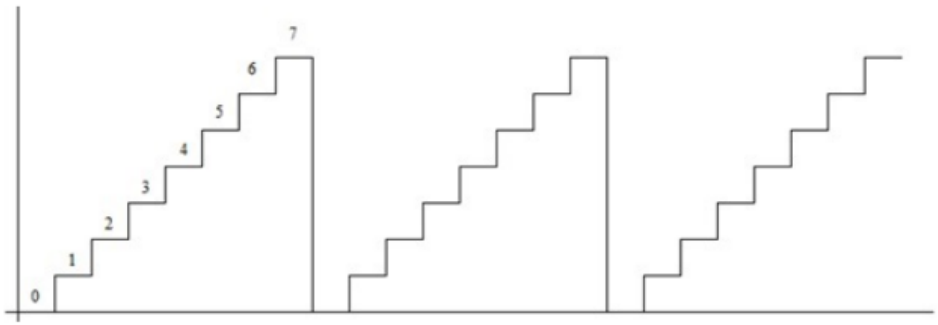
1. The Timer/Counter is a register that increments its value after every clock cycle.

Timer/Counter (TCNTn)

1. The Timer/Counter is a register that increments its value after every clock cycle.
2. The maximum value depends upon the resolution of Counter.

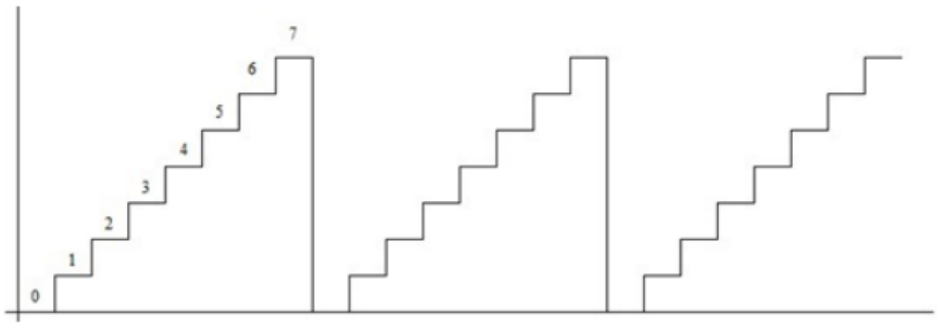
Timer/Counter (TCNTn)

1. The Timer/Counter is a register that increments its value after every clock cycle.
2. The maximum value depends upon the resolution of Counter.
3. For example, a 3 bit counter will have 8 values (i.e. 0-7). Its waveform will be seen as follows:



Timer/Counter (TCNTn)

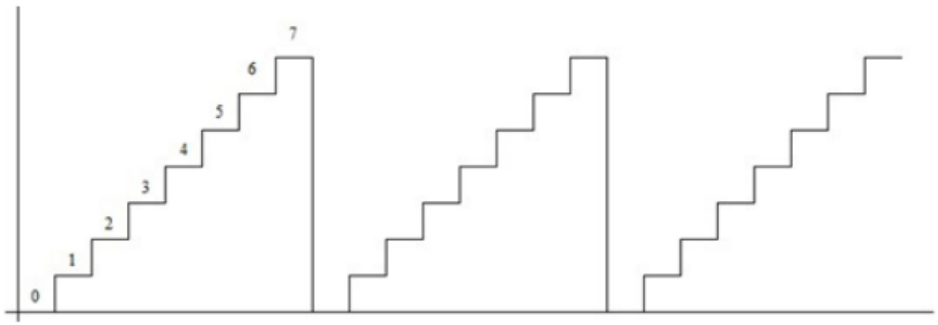
1. The Timer/Counter is a register that increments its value after every clock cycle.
2. The maximum value depends upon the resolution of Counter.
3. For example, a 3 bit counter will have 8 values (i.e. 0-7). Its waveform will be seen as follows:



4. For n-bit counter, maximum value = 2^n - 1.

Timer/Counter (TCNTn)

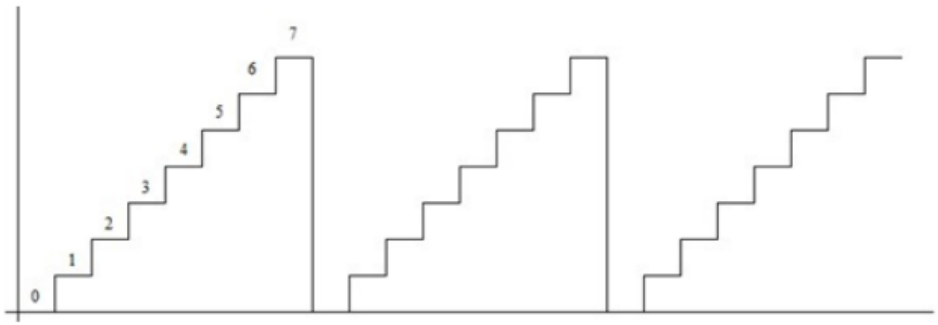
1. The Timer/Counter is a register that increments its value after every clock cycle.
2. The maximum value depends upon the resolution of Counter.
3. For example, a 3 bit counter will have 8 values (i.e. 0-7). Its waveform will be seen as follows:



4. For n-bit counter, maximum value = 2^n - 1.
5. For example, The Timer/Counter 5 is a 16 bit register.

Timer/Counter (TCNTn)

1. The Timer/Counter is a register that increments its value after every clock cycle.
2. The maximum value depends upon the resolution of Counter.
3. For example, a 3 bit counter will have 8 values (i.e. 0-7). Its waveform will be seen as follows:



4. For n-bit counter, maximum value = 2^n - 1.
5. For example, The Timer/Counter 5 is a 16 bit register.
6. We will use it in 8-bit mode, for PWM generation.

Output Compare Register (OCRnA, OCRnB and OCRnC)

Output Compare Register (OCRnA, OCRnB and OCRnC)

1. The value of the Timer/Counter is constantly compared with a reference value.

Output Compare Register (OCRnA, OCRnB and OCRnC)

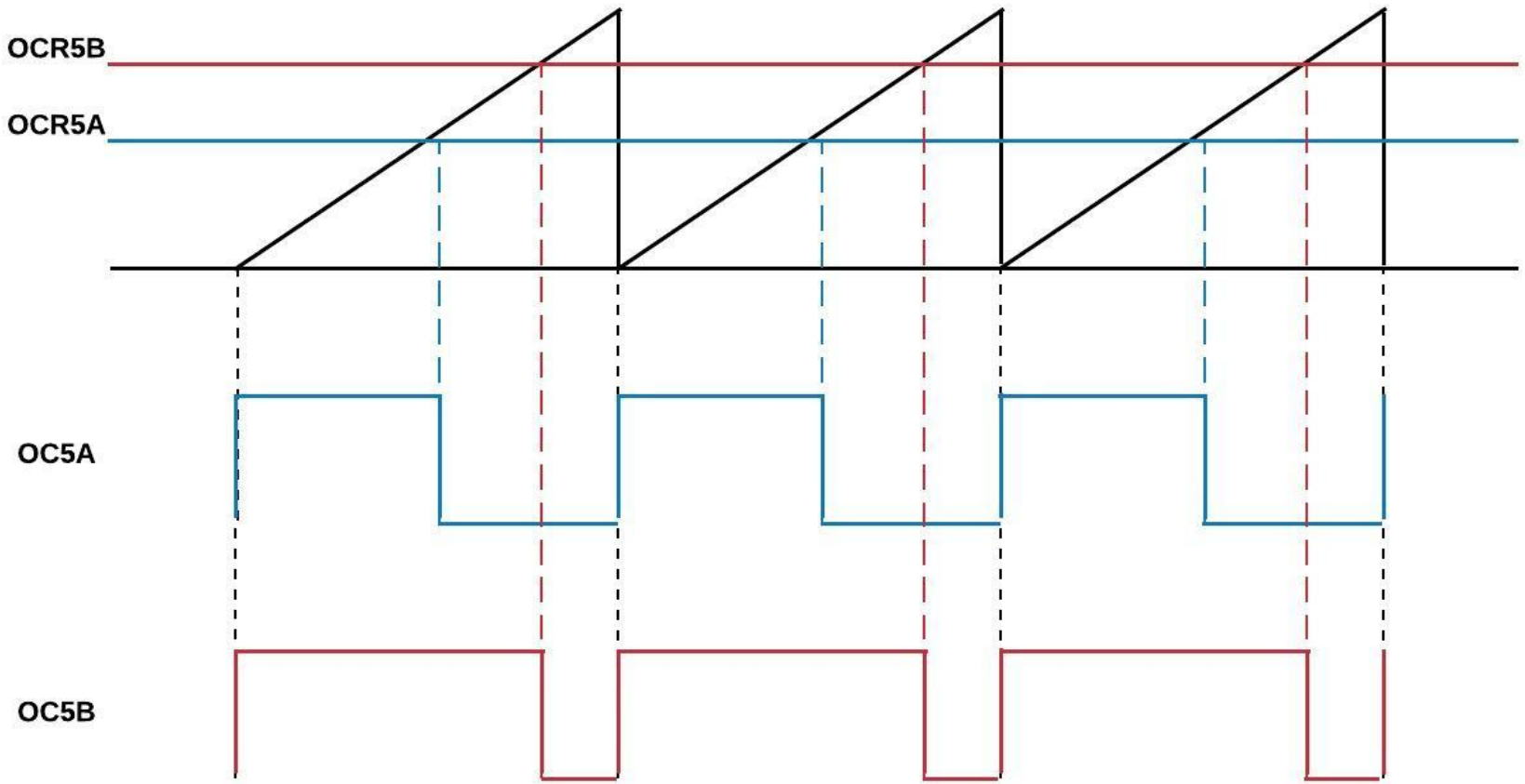
1. The value of the Timer/Counter is constantly compared with a reference value.
2. This reference value is given in the Output Compare Register (OCR).

Output Compare Register (OCRnA, OCRnB and OCRnC)

1. The value of the Timer/Counter is constantly compared with a reference value.
2. This reference value is given in the Output Compare Register (OCR).
3. For example, Output Compare Registers associated with Timer 5 for PWM generation: OCR5A, OCR5B and OCR5C.

PWM signal for pins OC5A OC5B

PWM signal for pins OC5A OC5B



Required Functions

Required Functions

In order to initialize PWM pin for eYFI Mega, we need to use the following functions:

Required Functions

In order to initialize PWM pin for eYFI Mega, we need to use the following functions:

- pinMode(pin_number, INPUT/OUTPUT);

Required Functions

In order to initialize PWM pin for eYFI Mega, we need to use the following functions:

- pinMode(pin_number, INPUT/OUTPUT);
- analogWrite(pin_number, 0 - 255);

Required Functions

In order to initialize PWM pin for eYFI Mega, we need to use the following functions:

- pinMode(pin_number, INPUT/OUTPUT);
- analogWrite(pin_number, 0 - 255);
  - Pins 2 - 13, 44 - 46 are PWM pins

Required Functions

In order to initialize PWM pin for eYFI Mega, we need to use the following functions:

- pinMode(pin_number, INPUT/OUTPUT);
- analogWrite(pin_number, 0 - 255);
  - Pins 2 - 13, 44 - 46 are PWM pins
  - 490 Hz (pins 4 and 13: 980 Hz)

Thank You!