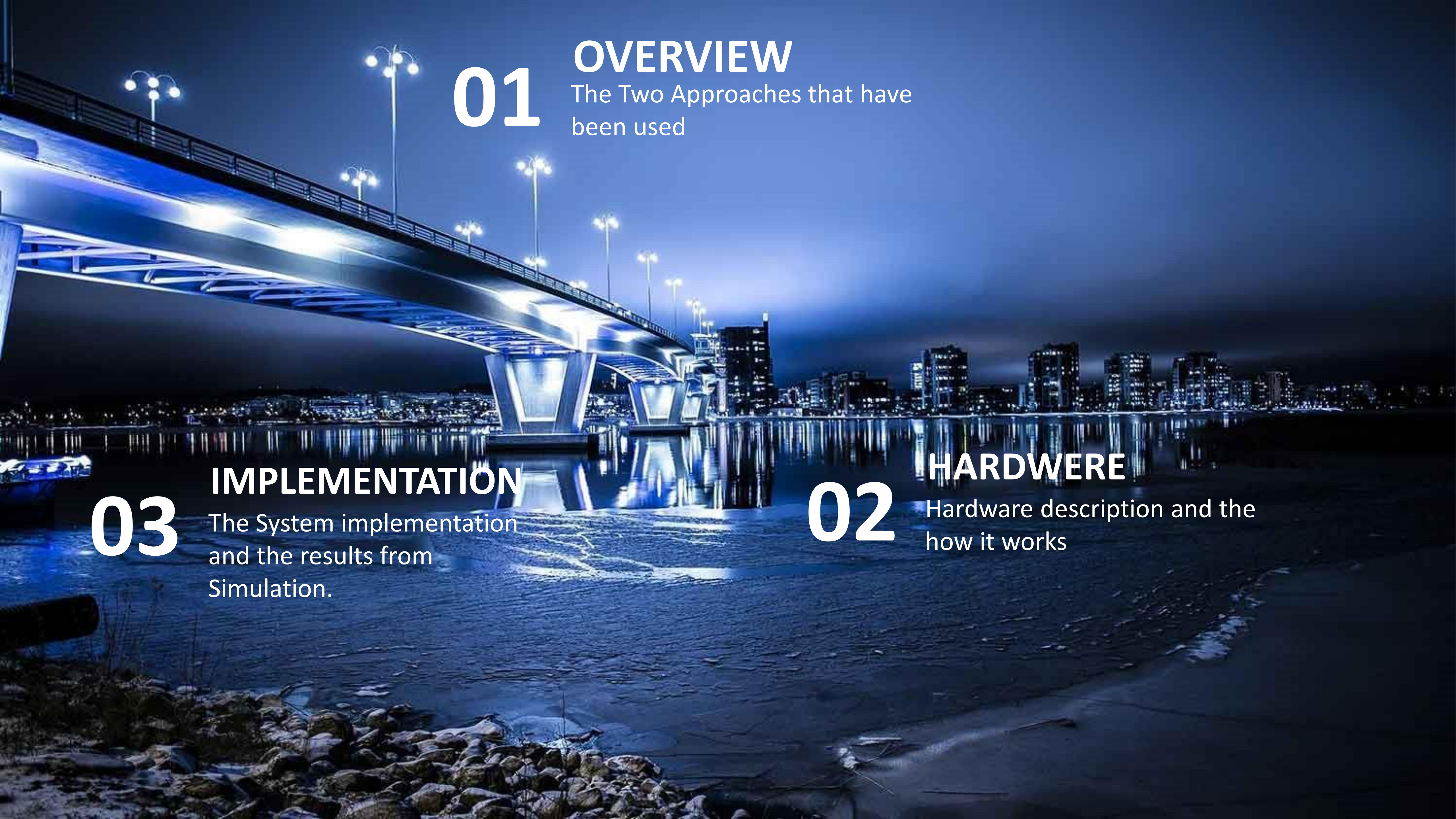
# SMART LIGHTING



## 01 OVERVIEW

The Two Approaches that have been used

## 02 HARDWERE

Hardware description and the how it works

## 03 IMPLEMENTATION

The System implementation and the results from Simulation.

# Internet Of Things

## IOT in Smart Systems

IoT automation is the **ability to control domestic appliances by electronically controlled, internet-connected systems**. It may include setting complex heating and lighting systems in advance and setting alarms and home security controls, all connected by a central hub and remote-controlled by a mobile app.

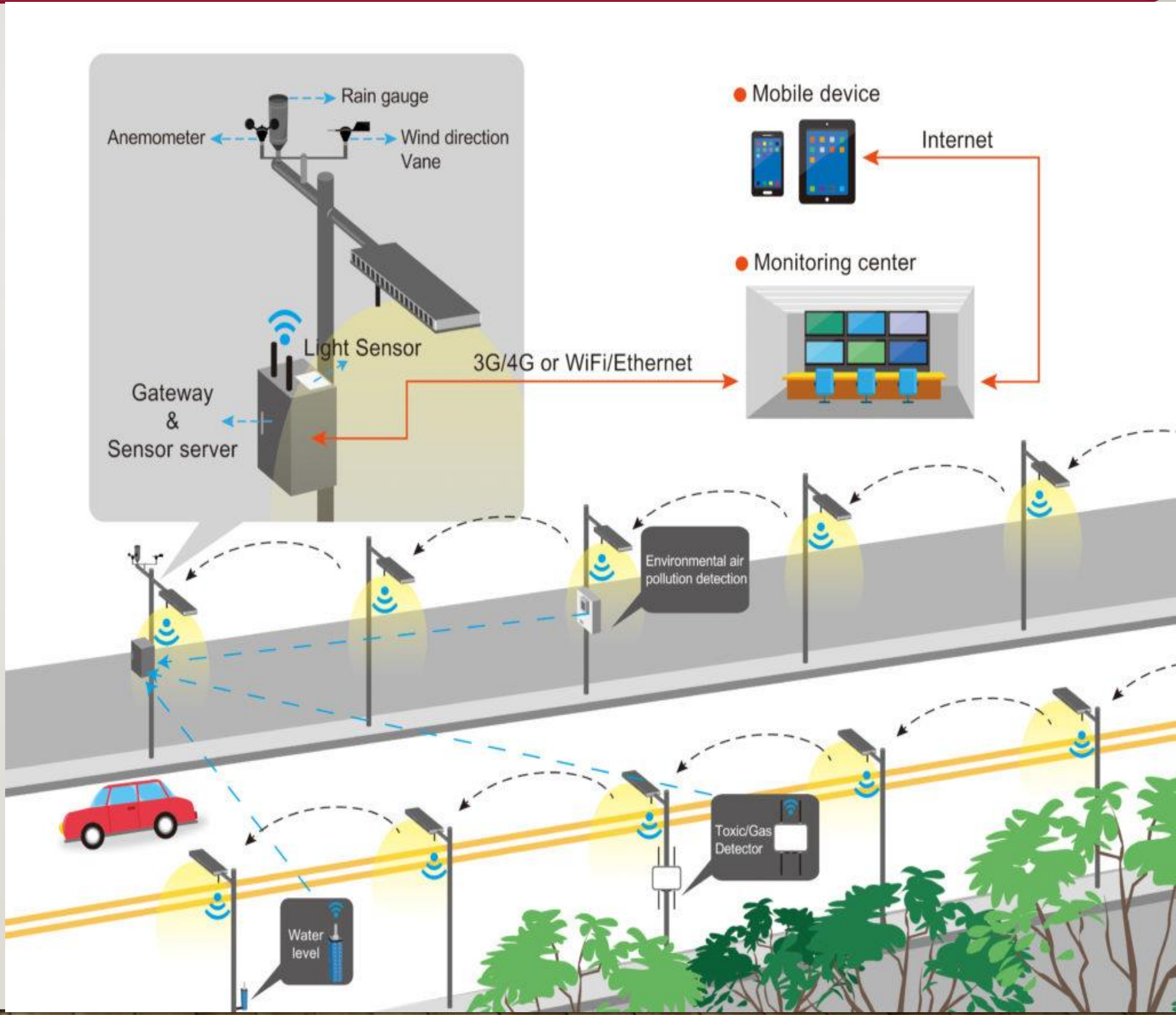
The IoT smart Light typically works via an internet-connected central hub that controls all the different devices.

IOT

# OVERVIEW

Accidents occurring in the night time are either caused by driver errors or malfunctioning street lights The malfunctions can be caused by fused bulbs or damaged wiring. This project reduces this error hence reducing accidents significantly.

Also the Manual control is prone to errors and leads to energy wastages and manually controlling during mid night is impracticable. Where the Smart Street Lighting System is an advance lighting system which is designed to increase the accuracy and efficiency for the street light by timed controlled Switching



# APPROACHES

## Light Sensor

Based on the Light Intensity outside we control the Light.

## Motion Detection

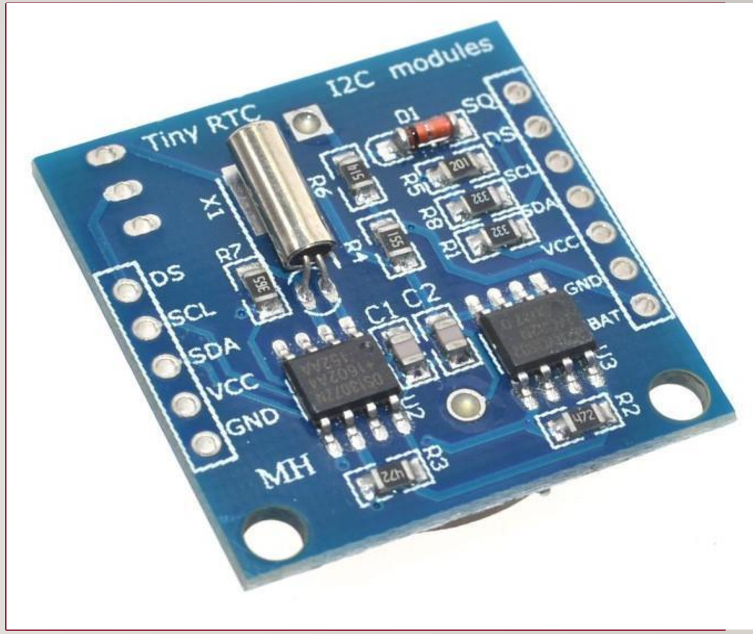
Based on the motion of a person or object.

## Real Time Clock ( RTC )

Based on the time of sunset or sunrise.

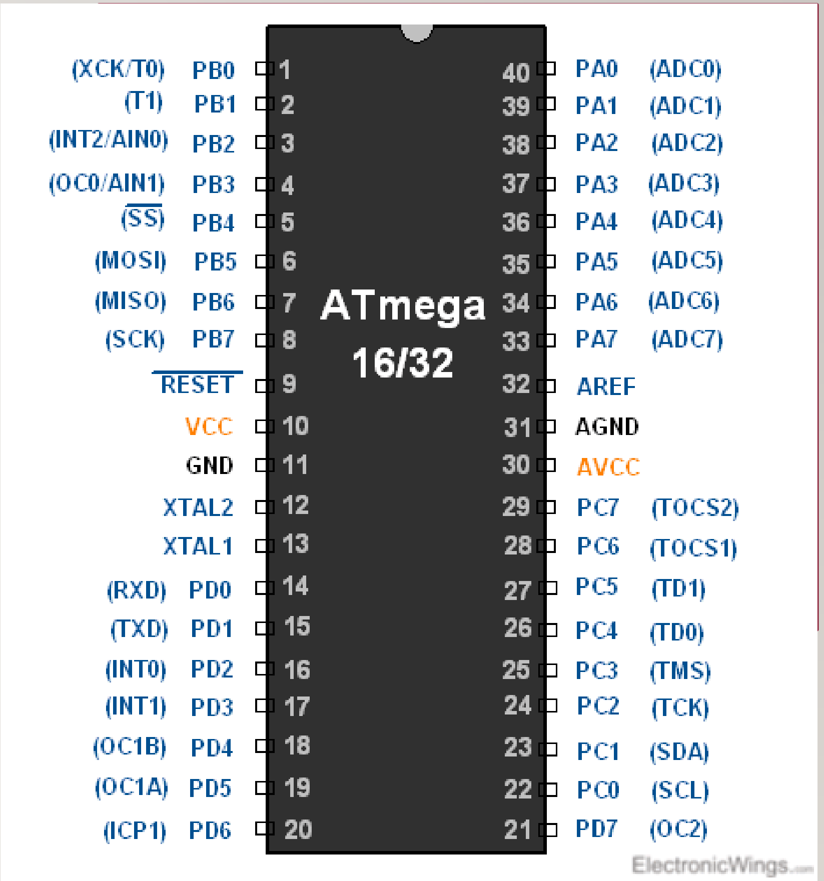
# Real Time Clock (RTC)

GENERAL DESCRIPTION The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I2 C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12- hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.



# ATMEGA32

The Atmel®AVR®ATmega32 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega32 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.



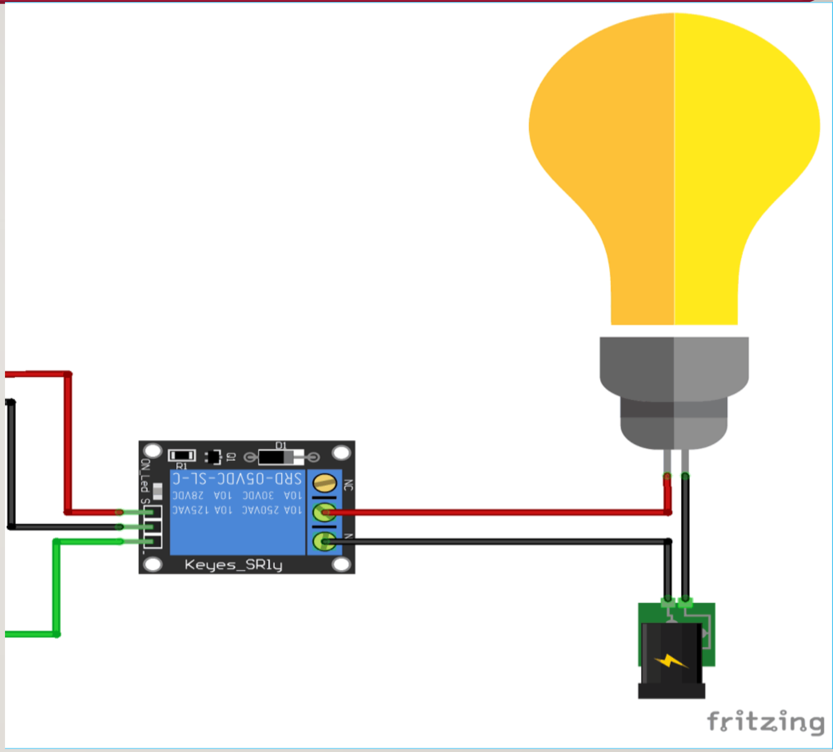
# System Implementation

1. We Can not connect the Light Bulb Directly to the Microcontroller as the Pins can deals only with 5v.

2. So We connect a RELAY Module to protect the Microcontroller Pin and the MC will act as a switch.

3. To Control the Light intensity of the Bulb we need to generate analog voltage.

4. MC only deals with digital values but using PWM we can generate Square wave that is equivalent to the analog voltage.

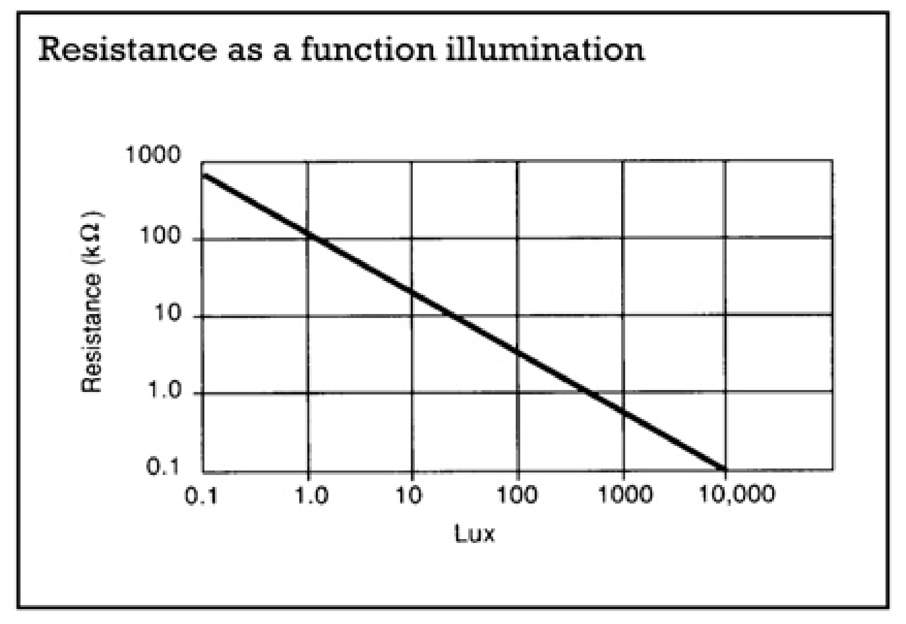


# Light Intensity (LDR Sensor)

Measures the intensity of the external light and when the degree of darkness is high, the LEDs goes to an ON state at night.( when the light intensity is increased the resistivity is increased hence the voltage drops).

This system works by sensing the intensity of light in its environment. The sensor that can be used to detect light is an LDR.

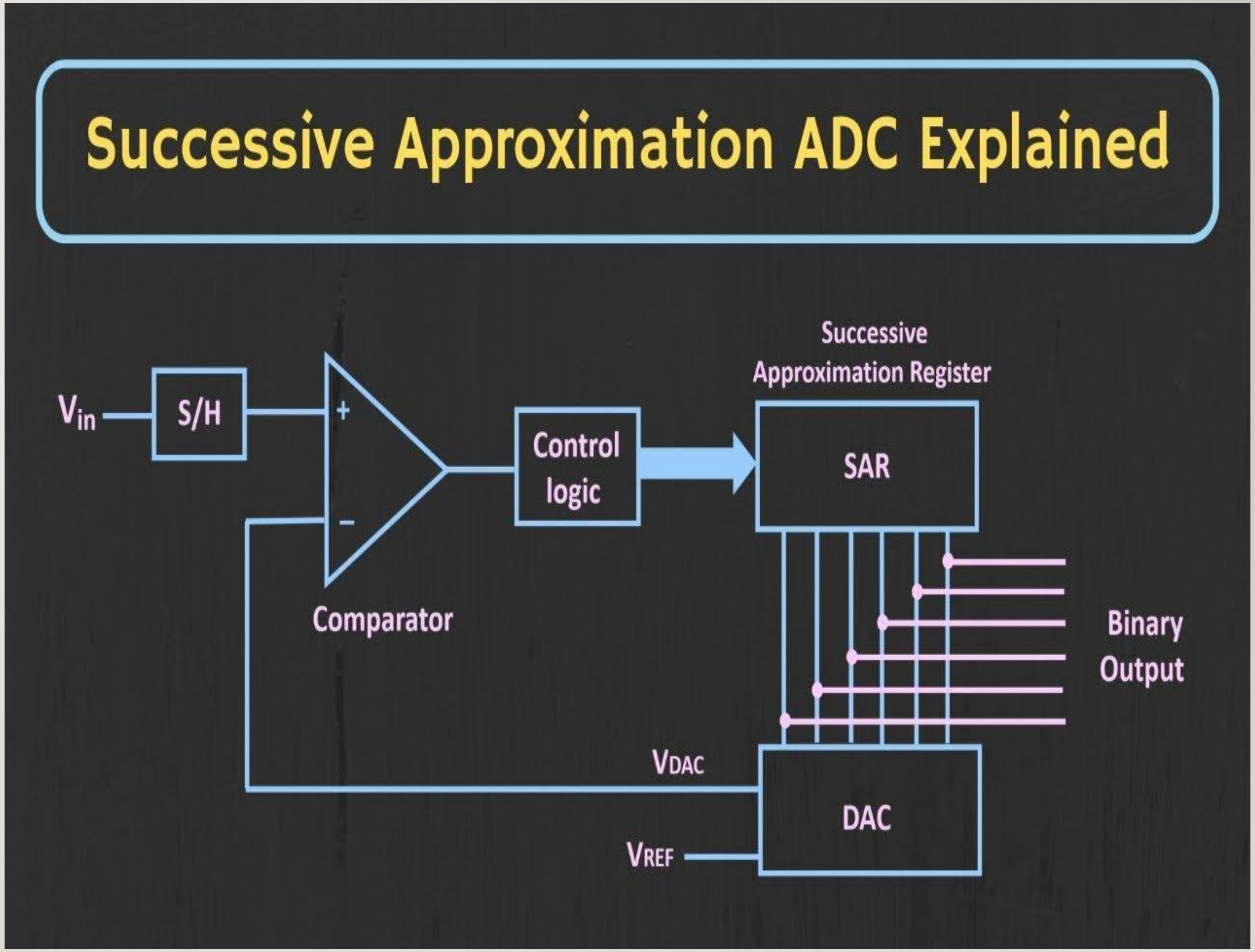
The LDR gives out an analog voltage when connected to VCC (5V), which varies in magnitude in direct proportion to the input light intensity on it. That is, the greater the intensity of light, the greater the corresponding voltage from the LDR will be



# ADC

*Successive Approximation ADC* (SAR ADC)

Its basic idea depends on a counter that counts on a clock signal called ***ADC Clock*** and every count it increases a ***compare analog signal voltage value*** that is compared to the input signal, the counter stop counting when the compare signal is approximately equals to the input signal. Then the counter value represent the digital value of the input signal.



# ADC

## ADC Resolution

**Number of the bits** that represent the output digital value.

## Reference Voltage

**Maximum Voltage** could be measured by the ADC, at this voltage all bits of the output value shall be 1

## ADC Step

The analog value needed to increase the digital value by 1

$$\boxed{\text{Step} = \frac{\text{Reference Voltage}}{2^{\text{ADC Resolution}}}}$$

## Conversion Time

**Time taken** by ADC to convert analog signal to digital signal

## ADC Clock

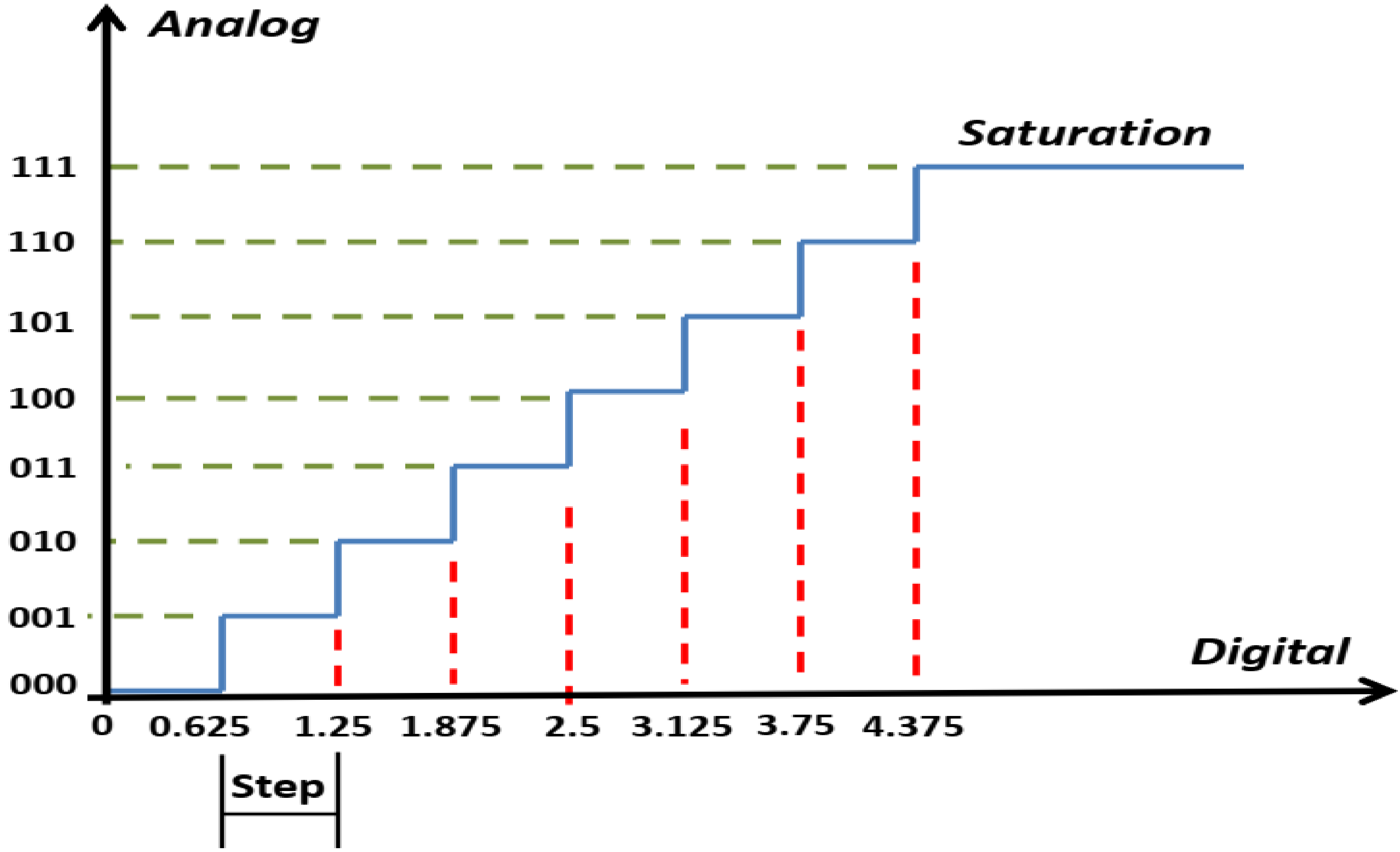
The input clock to the counter inside the ADC. The **higher** frequency of the ADC clock, the **lower** conversion time and **higher** power consumption.

# ADC

This graph shows the mapping between the analog values and digital values for ***3 bit ADC*** with reference voltage = 5V.

The 3 bit ADC has 8 possible digital values. So, simply the step = 8 / 5 = **0.625v**.

**Important Note:**
The ADC saturates at voltage =
Reference Voltage – 1 step.

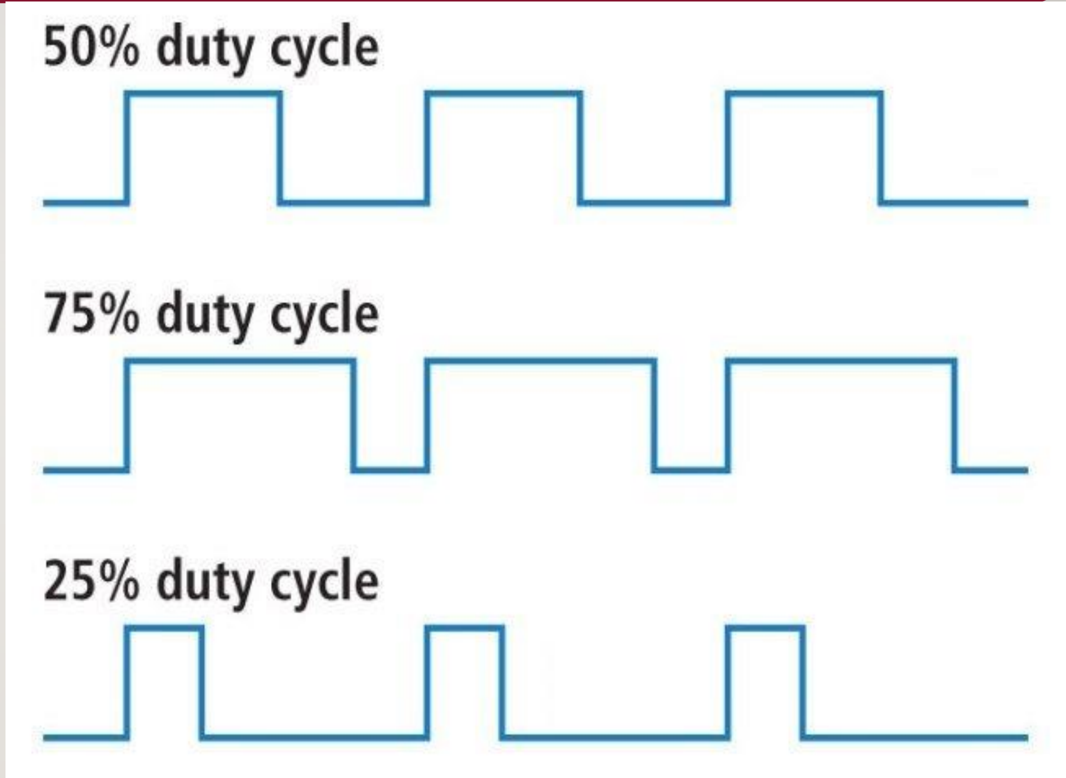


# Pulse Width Modulation (PWM)

- Pulse Width Modulation (PWM) is a method for changing how long a square wave stays "on".

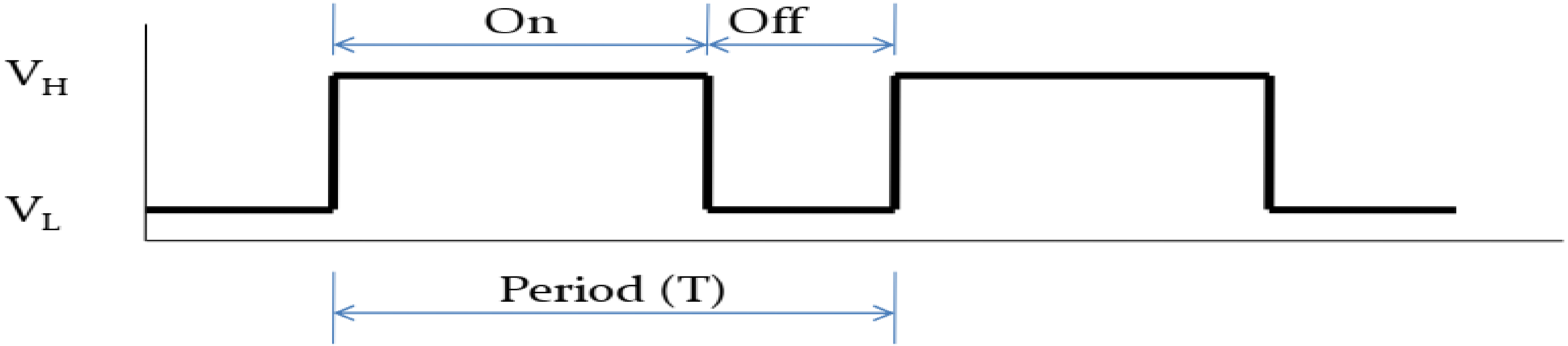
- The on-off behavior changes the average power of the signal.

- If signal toggles between on and off quicker than the load, then the load is not affected by the toggling.



# Pulse Width Modulation (PWM)

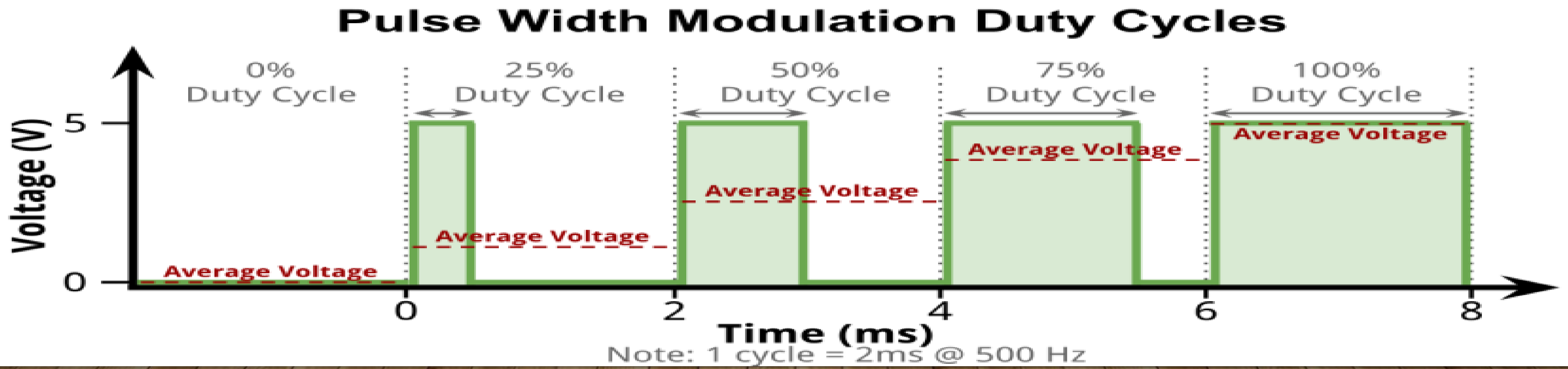
| | |
|---|---|
| Amplitude | The voltage difference between the on state and the off state. |
| Period | The repetition time for PWM.<br>Period = On_Time + Off_Time |
| Duty cycle | The percentage of the On_Time over the total time |



# Pulse Width Modulation (PWM)

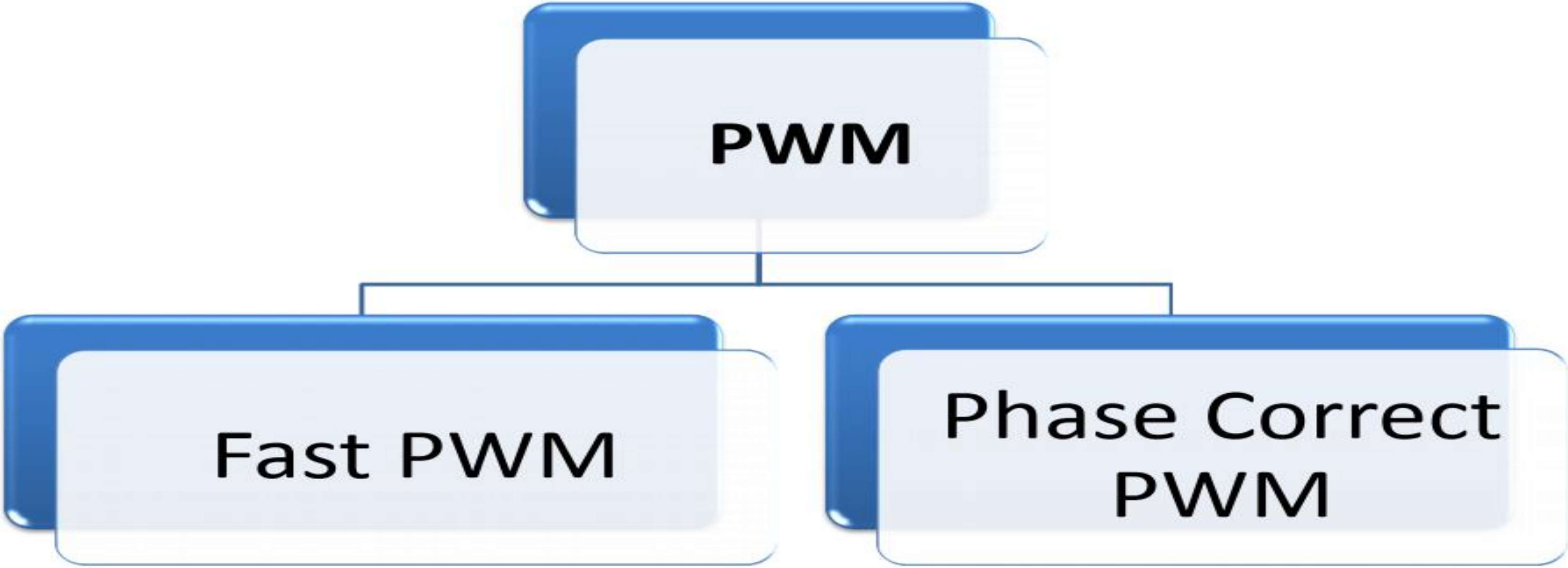
- The duty cycle is a percentage measurement of how long the signal stays on.
- ***The effective voltage*** of the PWM signal is called Root Mean Square (RMS) which equals to:

$$RMS = Amplitude \sqrt{Duty\ Cycle}$$



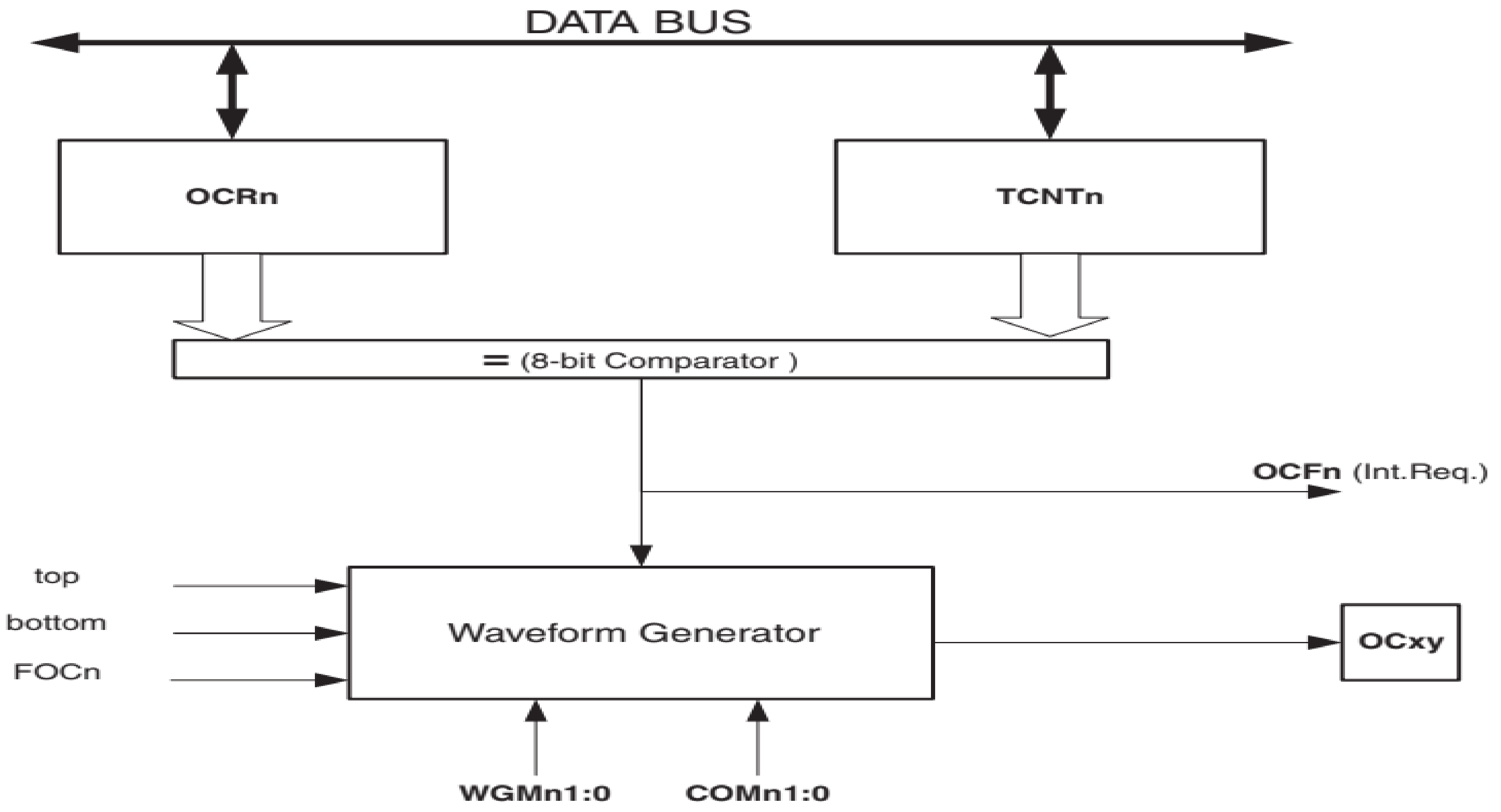
# Pulse Width Modulation (PWM)

**The AVR already has a PWM modes**



# Pulse Width Modulation (PWM)

**Figure 55.** Output Compare Unit, Block Diagram



# Pulse Width Modulation (PWM)

Figure 47. Phase Correct PWM Mode, Timing Diagram

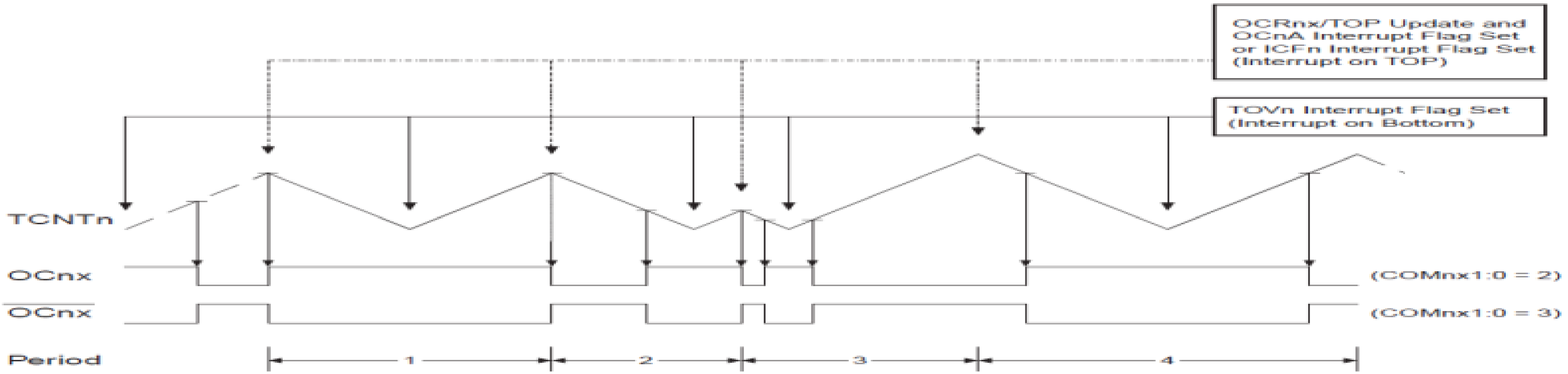
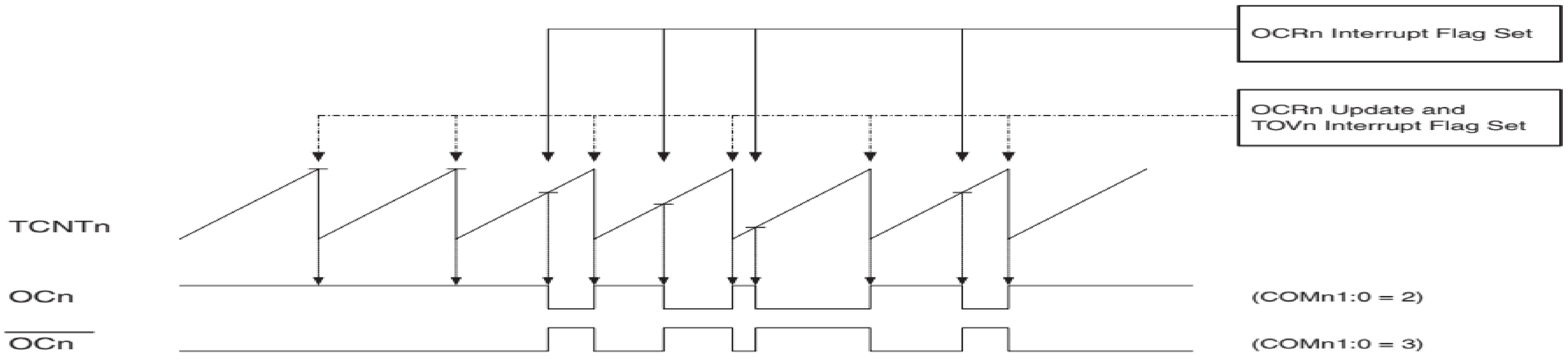



Figure 58. Fast PWM Mode, Timing Diagram



# Pulse Width Modulation (PWM)

## Fast PWM

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnPWM} = \frac{f_{\text{clk\_I/O}}}{N \cdot 256}$$

The N variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

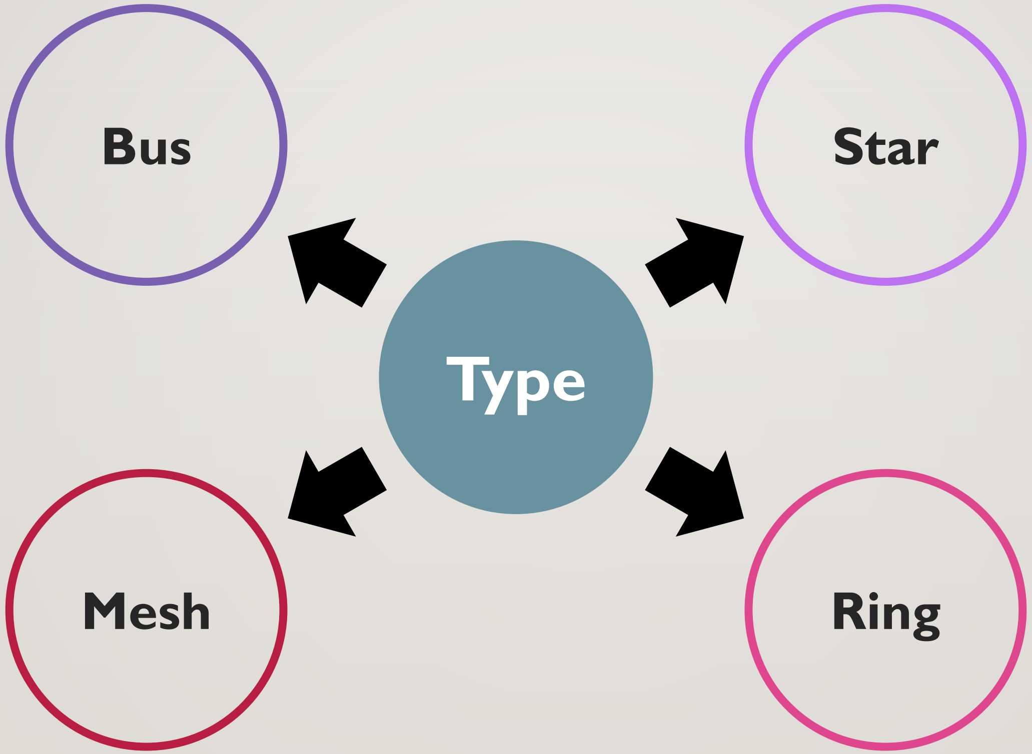
## Phase Correct PWM

PWM frequency for the output when using phase correct PWM can be calculated by the following equation:

$$f_{OCnPCPWM} = \frac{f_{\text{clk\_I/O}}}{N \cdot 510}$$

The N variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

# Network Topology



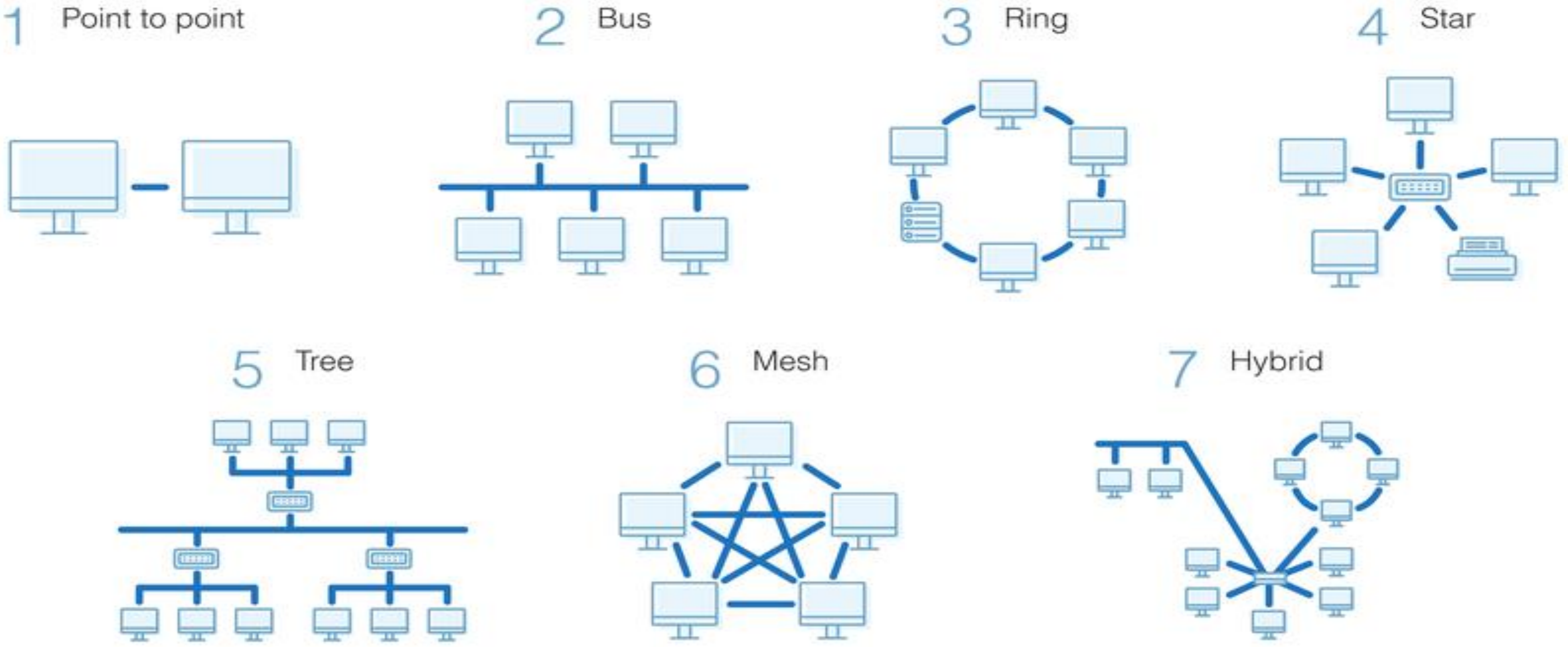

Bus topology is a network type in which every computer and network device is connected to a single cable.

In star topology, all the devices are connected to a single hub through a cable. This hub is the central node and all other nodes are connected to the central node

In a mesh topology, every device is connected to another device via a particular channel.

In this topology, it forms a ring connecting devices with its exactly two neighboring devices.

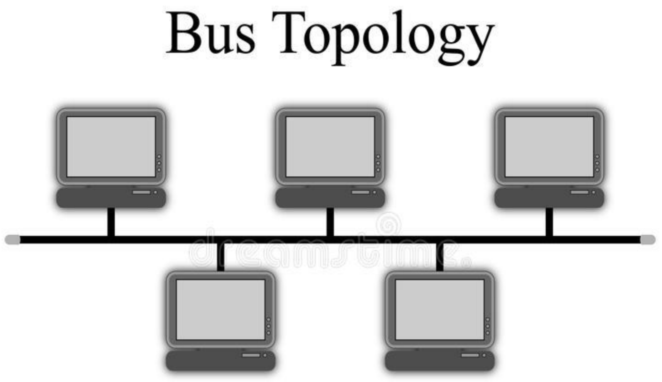
# Network Topology Types



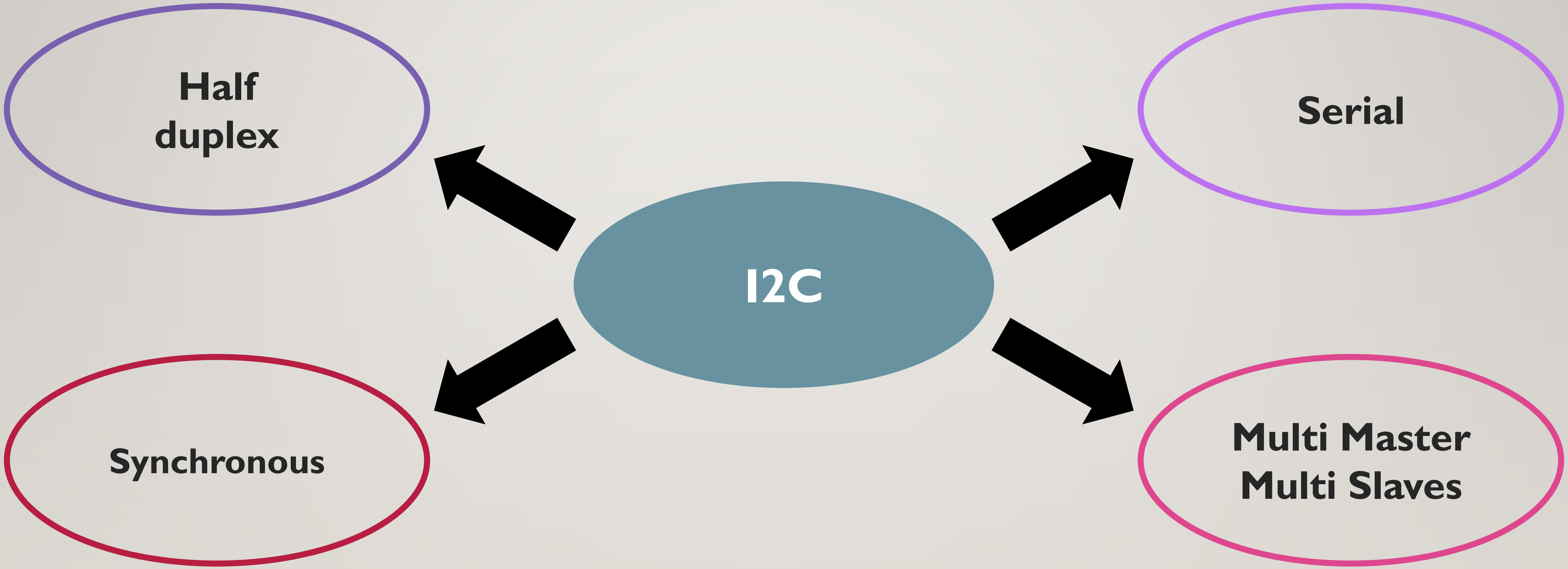
# Bus

## CSMA : Carrier Sense Multiple Access

Bus topology is a network type in which every computer and network device is connected to a single cable. It transmits the data from one end to another in a single direction. No bi-directional feature is in bus topology. It is a multi-point connection and a non-robust topology because if the backbone fails the topology crashes.



# I2C BUS


Half
duplex
Serial
I2C
Synchronous
Multi Master
Multi Slaves

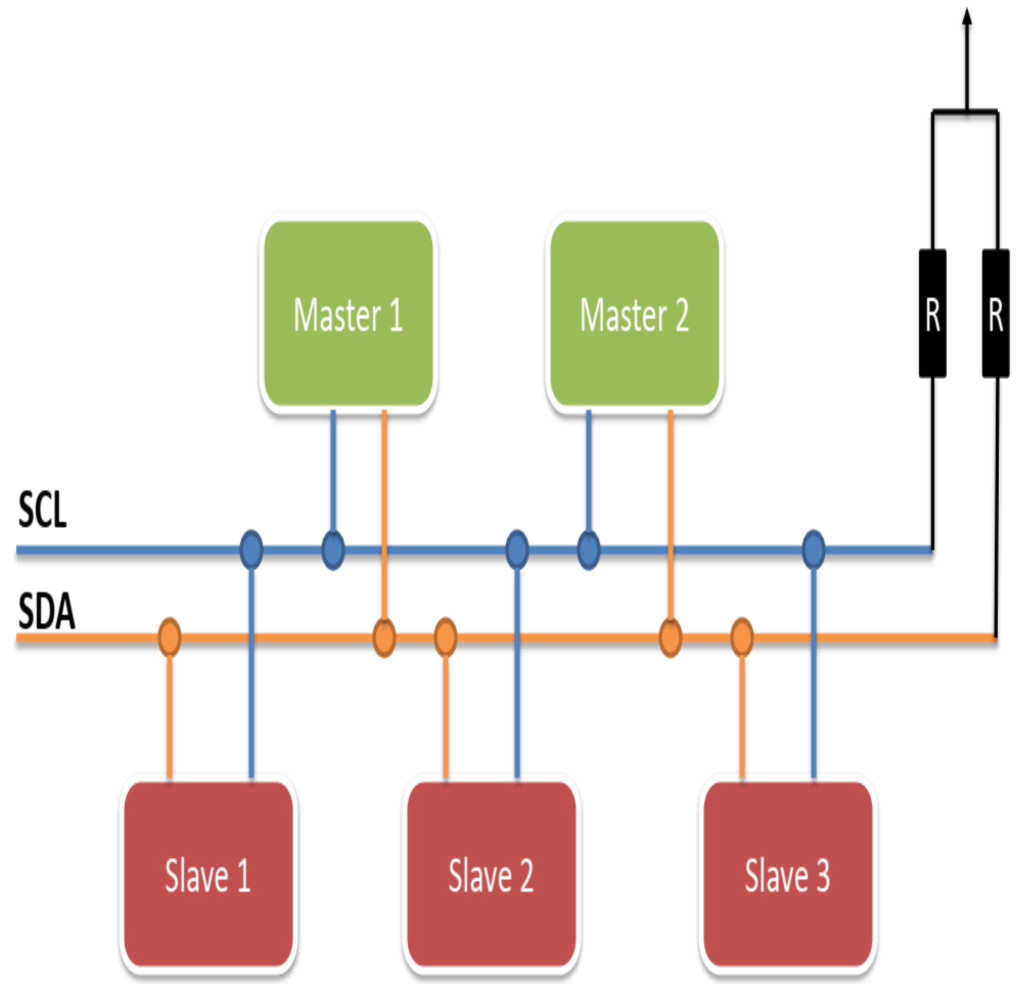
# RTC Connection With I2C

- **Inter-Integrated Circuit** **I²C** (Pronounced I Two C or I Squared C), is a serial communication protocol at which the devices are hooked up to the I2C bus with just ***two wires***.

- It is sometimes referred to as ***Two Wire Interface***, or the ***TWI***

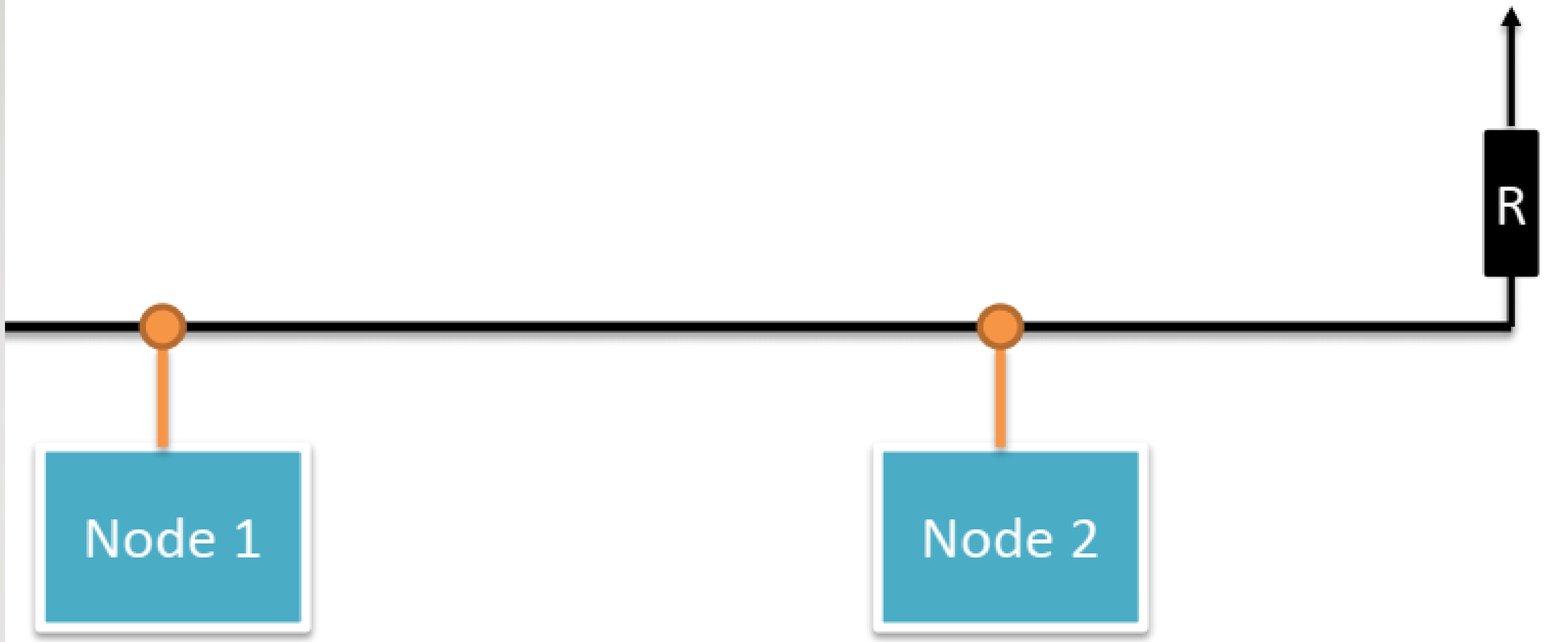
- Devices could be the CPU, IO Peripherals like ADC, or any other device which supports the I2C protocol.

- All the devices connected to the bus are classified as either being ***Master*** or ***Slave*** .



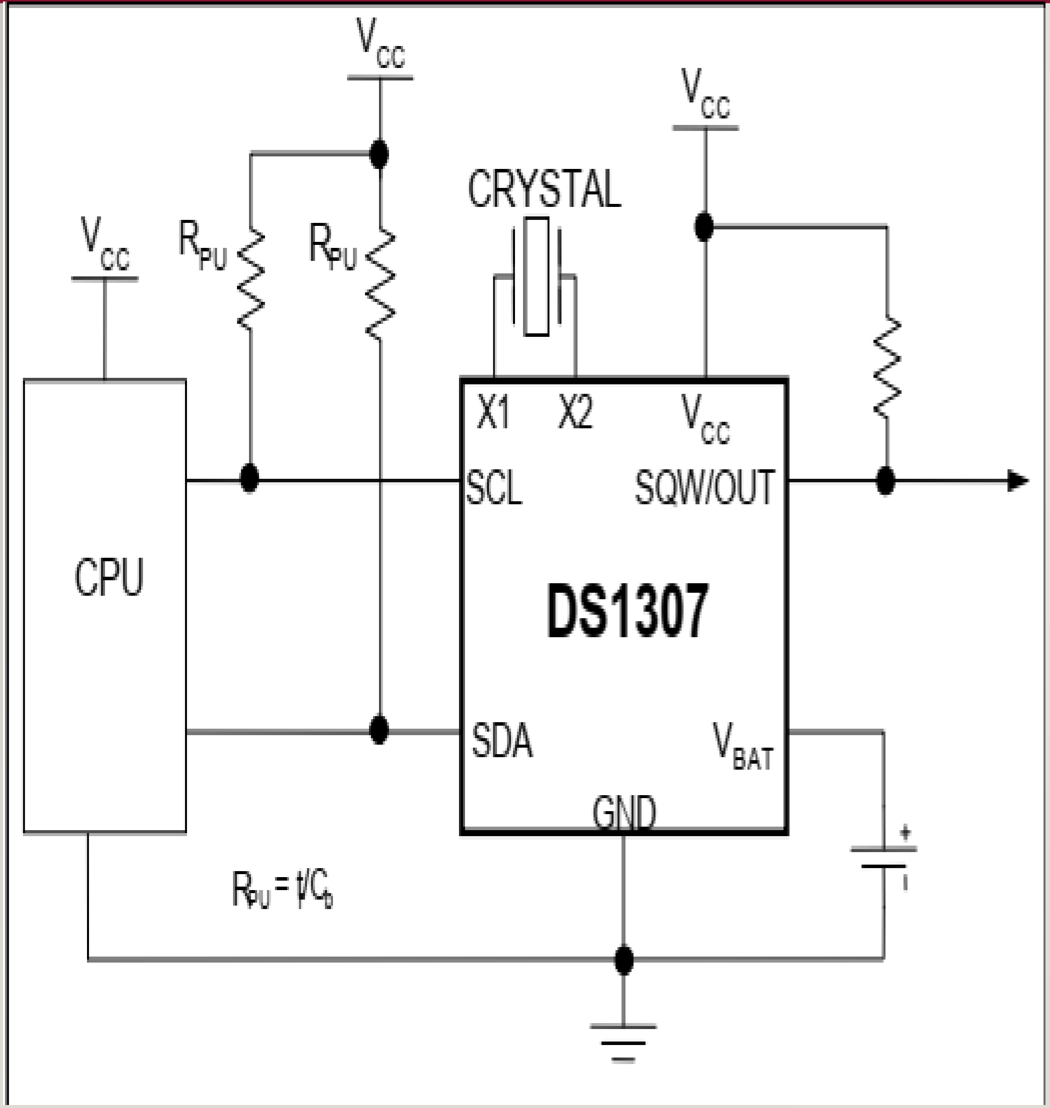
If the bus is open drain line, then any node that writes 0 the bus would be derived to GND, if all nodes writes 1 then the bus would be floating, that’s why we use a pull up resistor to connect the line to VCC which will be effective only if all nodes writes 1.

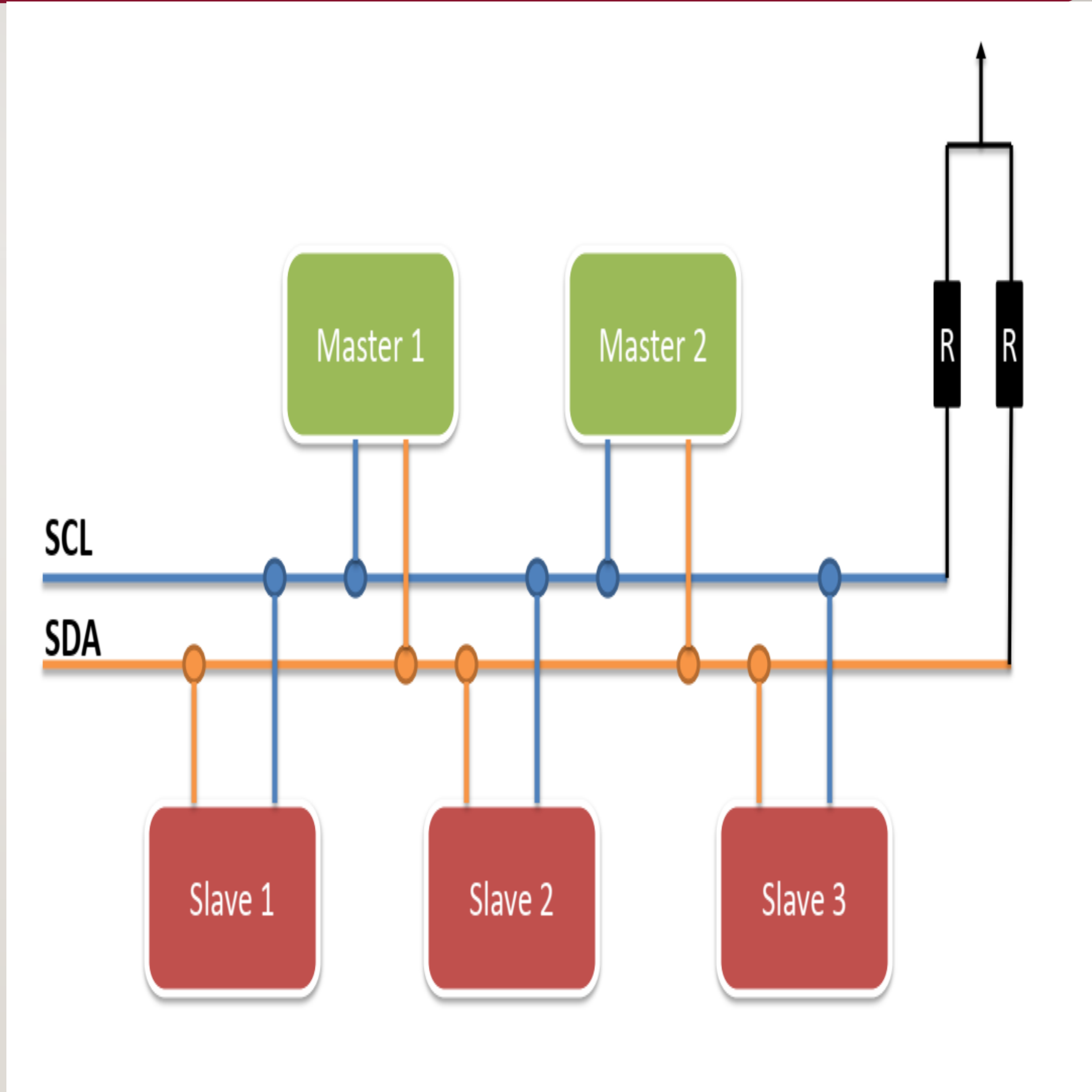
It looks like AND gate, any nodes writes 0, the bus derived to GND, if all nodes write 1, the bus derived to Vcc by the pull up resistor.




<table>
<tr><th>Node 1</th><th>Node 2</th><th>Result on Line</th></tr>
<tr><td>0</td><td>0</td><td>GND</td></tr>
<tr><td>1</td><td>1</td><td>VCC</td></tr>
<tr><td>0</td><td>1</td><td rowspan="2">GND</td></tr>
<tr><td>1</td><td>0</td></tr>
</table>

# RTC Connection With I2C



# Application