❖ **Explain how UART is used for communication highlighting the advantages of UART.**

June-07,6M

UART takes parallel data and transmits serially & UART receives serial data and converts to parallel.

A simple UART may possess

i) **Some configuration registers &**

ii) **Two independently operating processors, one for receiving and the other for transmitting.**

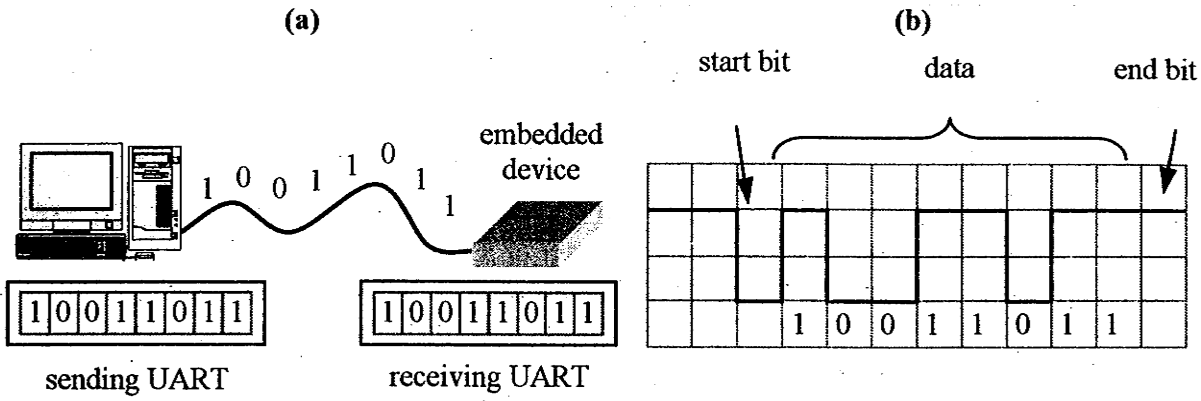


Figure 1 : Serial transmission using UARTs, (a) A PC communicating serially with an embedded device, (b) transmission protocol used by the two UARTs.

* The transmitter may posses a register called a transmit buffer, that holds data to be sent. This register is a shift register, so the data can be transmitted one bit at a time by shifting at the appropriate rate.

* The receiver receives data into a shift register, and then the data can be read in parallel
The receiver is constantly monitoring the receive pin (rx) for a start bit. The start bit is typically signaled by a hight to low transition on the rx pin

* After the start bit has been detected, the receiver starts

sampling the rx pin at predetermined intervals shifting each sampled bit into the receive shift register

* To determine whether the transmitted data is correct, the transmitter transmits an additional parity bit.

The UART can be configured to check for even parity or no parity at all. once data is received, the UART signals its host processor. The host processor in turn reads the byte out of the receive shift register. The receiver is now ready to receive more data.

Transmitter Operation:-

The host processor (Txing side processor) writes a byte to the transmit buffer of the UART, the transmitter sends a start bit over its transmit pin (tx), signaling the beginning of a transmission to the remote UART. Then, the transmitter shifts out the data in its transmit buffer over its tx pin at a predetermined rate.

(Txer can also Tx its an additional parity bit)

At this point, the UART processor signals its host processor, indicating that it is ready to send more data if available.

* The transmission protocol used by UART's determines the rate at which bits are sent and received & is called baud rate. The protocol also specifies the number of bits of data and the type of parity sent during each transmission.

* The baudrate determines the speed at which data is exchanged between two serially connected UART's. The

commonly used baud rates are 2400, 4800, 9600 & 19200.

* To use a UART, we must configure its baud rate by writing to the configuration register, and then we must write data to the transmit register and/or read data from the received register.

* To use a UART, we must configure its baud rate by writing to the configuration register, and then we must write data to the transmit register and/or read data from the received register.

For ex, to configure the UART of an 8051 microcontroller we must use the following equation:

$$\boxed{\text{Baud rate} = \left(\frac{2^{smod}}{32}\right) * \frac{\text{OSC freq}}{[12 * (256 - TH1)]}}$$

Where,
smod corresponds to 2-bits in a special-function register, OSC freq is the frequency of the oscillator, & TH1 is an 8-bit rate register of a build-in timer.

❖ **Determine the values for smod and TH1 to generate a baud t=rate of 9600 for the 8051 baud rate equation, assuming an 11.981MHz oscillator. Remember that smod is a 2 bit and TH1 is 8-bits. There is more than one correct answer.**

Given:

Baud rate = 9,600

TH1 = 8-bit

$f = 11.981 MHz$

smod = 2 bit ie 00, 01, 10, 11.

Sol :-

$$\text{Baudrate} = \left(\frac{2^{\text{Smod}}}{32} \right) * \frac{\text{osc freq}}{[12 * (256 - TH1)]}$$

for smod = 2 = 10 :-

$$9600 \cancel{=} \left(\frac{2^2}{32} \right) * \frac{11.981 \times 10^6}{[12 * (256 - TH1)]}$$

$$76,800 = \frac{11.981 \times 10^6}{12 * (256 - TH1)}$$

$$12 \times 256 - 12\,TH1 = \frac{11.981 \times 10^6}{76800}$$

$$3072 - 12TH1 = \frac{11.981 \times 10^6}{76800}$$

$$12TH1 = 3072 - 156.0026$$

$$12TH1 = 2,915.9974$$

$$TH1 = 242.999$$

$$\boxed{TH1 = 243 = 11110011}$$

for smod = 3 = 11 :-

$$\text{Baud rate} = \left(\frac{2^{\text{smod}}}{32} \right) * \frac{\text{osc freq}}{[12 * (256 - TH1)]}$$

$$9600 = \frac{2^3}{32} \times \frac{11.981 \times 10^6}{[12 * (256 - TH1)]}$$

$$\frac{9600 \times 32}{8} = \frac{11.981 \times 10^6}{[12 \times (256 - TH1)]}$$

$$38,400 = \frac{11.981 \times 10^6}{(12 \times 256 - 12\,TH1)}$$

$$3072 - 12TH1 = \frac{11.981 \times 10^6}{38400}$$

$$3072 - 12TH1 = 312.0052$$

$$12TH1 = 3072 - 312.0052$$

$$12TH1 = 2759.994$$

$$\boxed{TH1 = 229.99}$$

$$\boxed{TH1 = 230 = 11100110}$$

PULSE WIDTH MODULATION (PWM) :-

❖ **Describe the working of PWM unit with timing diagrams. How can it be used for speed control of DC motor.**

June-11,10M

❖ **With a neat diagram, explain how the pulse width modulator works. What are the considerations in selecting the clock, the prescalar and the counter? Assuming an 8-bit up counter, calculate the count to be loaded in the 'cycle-high' register to get pulses of duty cycle 75%.**

Jan-11,10M June-06,10M

❖ **Describe the working of a PWM unit with a circuit and waveform.**

June-09,6M

❖ **Describe the working of a PWM unit with timing diagrams. How it can be used for speed control of DC motor.**

Jan-08,8M

❖ **Schematically explain how a PWM helps in controlling the speed of DC motor.**

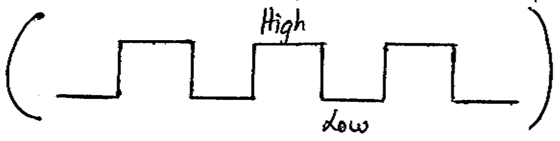
June-07,6M

❖ **How PWM can be used for speed control of DC motor? Explain.**

Jan-07,8M

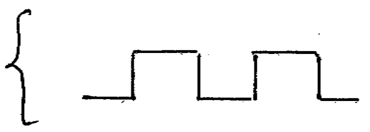
Pulse width Modulator (PWM):-

* A pulse width modulator (PWM) generates an o/p signal that repeatedly switches between high and low values



* We control the duration of the high value and of the low value by indicating the desired period (T) & the desired duty cycle (D).

* Duty cycle is defined as the ratio of ON time to the total period (ON + off) & is expressed in percentage.



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

* There are 3 common use of PWM:

1) To generate a clock-like signal to another device
   ex:- PWM can be used to blink a light at a specific rate.

2) To control the average current or voltage input to a device.
   Ex:- A DC electric motor rotates when its input voltage is set high, with the rotation speed proportional to the input voltage level.

   Suppose the revolutions per minute (rpm) equals 10 Times the input voltage. To achieve a desired rpm of 125, we we would need to set the input voltage to 1.25V, where as achieving 250 rpm would require an input voltage of 2.50V

3) To encode control commands in a single signal for use by another device.

ex:- We may control a radio - controlled car by sending pulses of different widths.

A width of 1-mes corresponds to a turn left command, a 4-msec width to turn right, and an 8-msec width to forward.

* The PWM approach makes use of the fact that a DC motor does not come to an immediate stop when its input voltage is lowered to '0', but rather its it coasts.

Thus the average input voltage is set to obtain the desired speed.

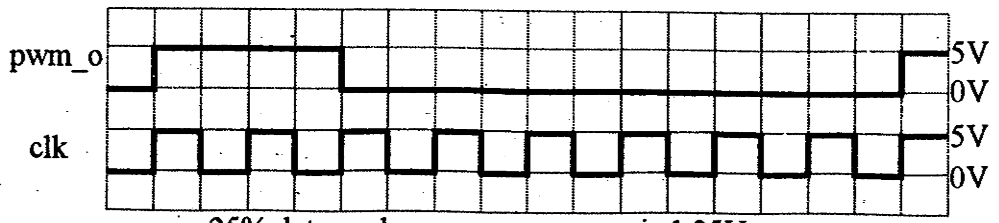
using a PWM, duty cycle is set to achieve the appropriate average voltage.

* Assuming the PWM's out is 5V when high and 0V when low, then.

* We can obtain an average o/p of 1.25V by setting the duty cycle to 25% ie 5V * 25% = 1.25V. This duty cycle is shown in fig 1ⓐ.

5V × 0.25 = 1.25V.

(a)



25% duty cycle – average pwm_o is 1.25V.

Fig① a.

* We can obtain an average o/p of 2.50V by setting the duty cycle to 50% as shown in fig ① b.

$0.5 \times 5V = 2.5V$

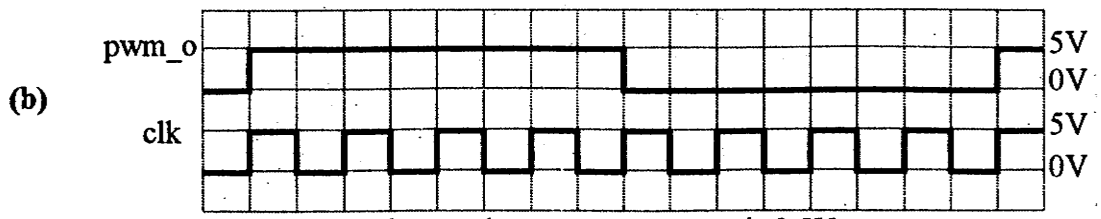


Fig ① b 50% duty cycle – average pwm_o is 2.5V.

* A duty cycle of 75% would result in average o/p of 3.75V as shown in fig 1 ⓒ.

$0.75 \times V = 3.75V$

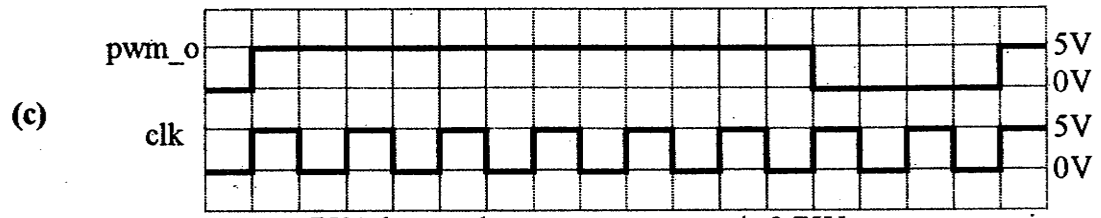


Fig ①. c 75% duty cycle – average pwm_o is 3.75V.

Controlling a DC Motor Using a PWM

* The speed of the DC Motor is proportional to the voltage applied to the motor. We must set the duty cycle of a PWM such that the average o/p voltage equals the desired voltage.

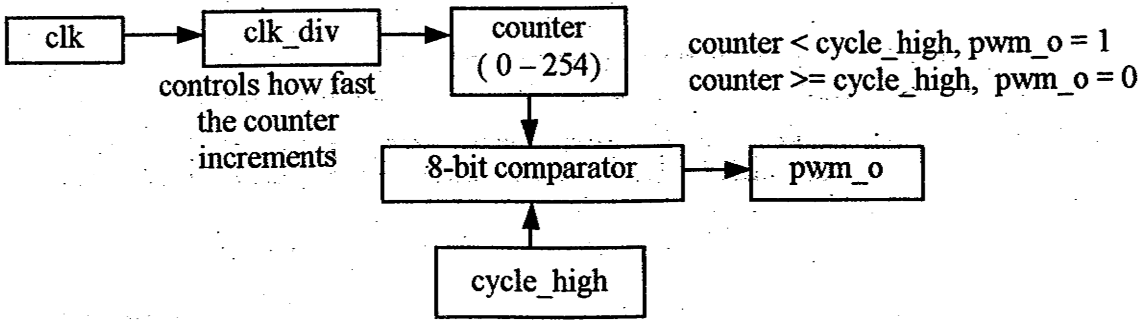

clk
clk_div
controls how fast the counter increments
counter (0 – 254)
counter < cycle_high, pwm_o = 1
counter >= cycle_high, pwm_o = 0
8-bit comparator
pwm_o
cycle_high

Fig 1 ⓐ : Internal Structure of PWM

fig 1 ⓐ : Internal Structure of PWM

* There are two 8 bit registers called <u>clk-div</u> and <u>cycle-high</u>, an <u>8 bit counter</u> and an 8-bit comparator as shown in fig 1-ⓐ.

* Initially, the value of clk_div is loaded into the register. The clk_div register works as a clock divider. After a specified amount of time has elapsed, a pulse is sent to the counter register. This causes the counter register to increment itself. The comparator then looks at the values in the counter register & the cycle-high register.

* When the counter value < cycle_high, a 1 (+5V) is outputted. When the counter value ≥ cycle_high, a 0 (0V) is outputted.

* When the counter value reaches 254, counter is reset to <u>0</u> and the process repeats. Thus clk-div determines the PWM's period, specifying the number of cycles in the period.

* The register <u>cycle_high</u> determines the <u>duty cycle</u>, indicating how many of periods cycle output a 1.
 If the cycle-high is set to 255 (FFh), the o/p signal is always high resulting in a duty cycle of <u>100%</u>.

 Illy if the cycle-high is <u>set to 0(00h)</u>, the o/p signal is always low resulting in a duty cycle of <u>0%</u>.

* If the value loaded to the clk_div is too low, the value outputted by the comparator oscillates too quickly. The comparator never outputs zeros long enough for the DC motor to slow down, causing the DC motor to continously run at full speed.
Setting the value of clk_div to FFh, in this case it works best.

* The relationship between applied voltage & DC motor speed:

input voltage	% of maximum voltage applied	RPM of DC motor
0	0	0
2.5	50	4,600
3.75	75	6,900
5.0	100	9,200

* For the motor to run at 4,600 RPM, 50% duty cycle is needed. The required duty cycle is computed as
 254 X 0.5 = 127 = 7Fh.
 Thus loading 7Fh into the cycle_high register.
* IIIly to run motor at 6,900RPM, 75% duty cycle is needed. The required duty cycle is computed as
 254 X 0.75 = 191 = BFh
 Thus loading BFh into the cycle_high register
* The PWM does not provide enough current to run the DC motor. Thus an NPN transistor is used to drive the DC motor as shown below.

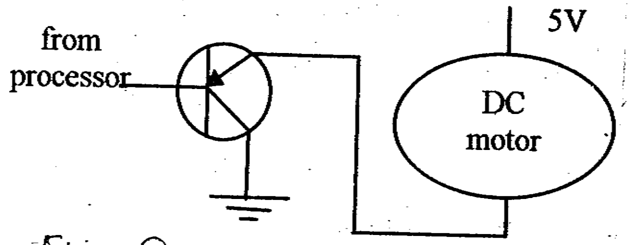


Fig 2(a): Connection to DC motor

The pseudo code is :

```
void main(void) {

   /* controls period */
   PWMP = 0xff;
   /* controls duty cycle */
   PWM1 = 0x7f;
   while(1){};
}
```