UNIT-V
SYSTEM DESIGN USING MP &MC

STANDARD MOTOR CONTROL

A **motor** can be driven and **controlled** by virtually any **microcontroller**; I/O, timers, A/Ds, and D/As are pretty **standard** and it is understood that you will be **using** external circuitry to do the heavy lifting. ... For example, in many applications, **motors** will be driven that are subject to varying loads and stresses

PWM Based DC Motor Speed Control using Microcontroller:

It is important to control the speed of DC motor in many applications, where precision and protection are essential.

Here we will use a technique called PWM (Pulse Width Modulation) to control the speed of DC motor.

We can achieve speed control of DC motor using mechanical or electrical techniques but they require large size hardware to implement but a Microcontroller based system provides an easy way to control the speed of DC motor

How to Generate PWM in 8051 Microcontroller?

Most modern microcontrollers like AVR (Arduino, for example), ARM, PIC, etc. have a dedicated PWM hardware and pins to activate PWM mode instantly. However, 8051 Microcontrollers do not have this provision. So, how to generate PWM in 8051 Microcontroller?

For this we have to use Timers and Interrupts in 8051 Microcontroller. The Timer0 of 8051 is configured in Mode0. By carefully adjusting the High and Low levels, we can maintain a constant period of the signal.
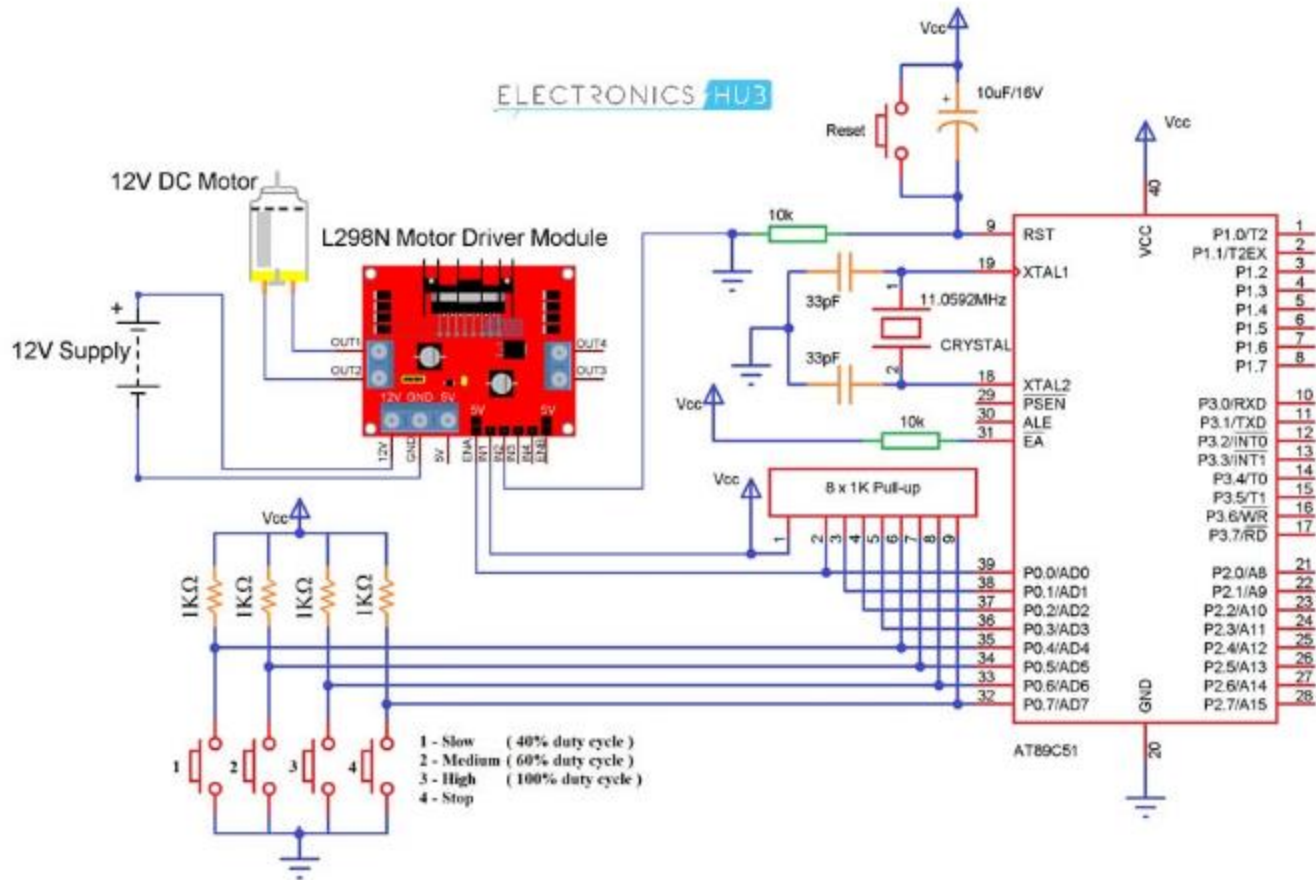
PWM based DC Motor Speed Control using Microcontroller Circuit Design:

The circuit consists of one 8051 Microcontroller (and its supporting circuitry related to oscillator and reset), L298N Motor Driver Module, a DC Motor and a few push buttons.

A 12V DC Motor is connected to the L298N Motor Driver Module at its OUT1 and OUT2 Pins. The IN1 and IN2 pins of the motor driver are connected to +5V (VCC) and GND. The EN1 pin of the Motor driver is connected to Port0 Pin P0.0.

Four Push Buttons are connected to Port0 Pins P0.4, P0.5, P0.6 and P0.7.

Generally, we can interface switches to the micro controller in two configurations; one is Pull-up configuration and the other is pull-down configuration.

Pull-up configuration:

In pull up configuration, the microcontroller pin is pulled HIGH to LOGIC 1 and the button is connected to GND. When button is pressed, microcontroller pin receives LOGIC 0

Pull-down configuration:

In pull down configuration, the microcontroller pin pulled down to LOGIC 0 and the button is connected to VCC. When button is pressed, microcontroller pin receives LOGIC 1.

In our circuit we are using pull up configuration. So, we need to check for logic 0 in order to know whether the button is pressed or not.

THANK YOU