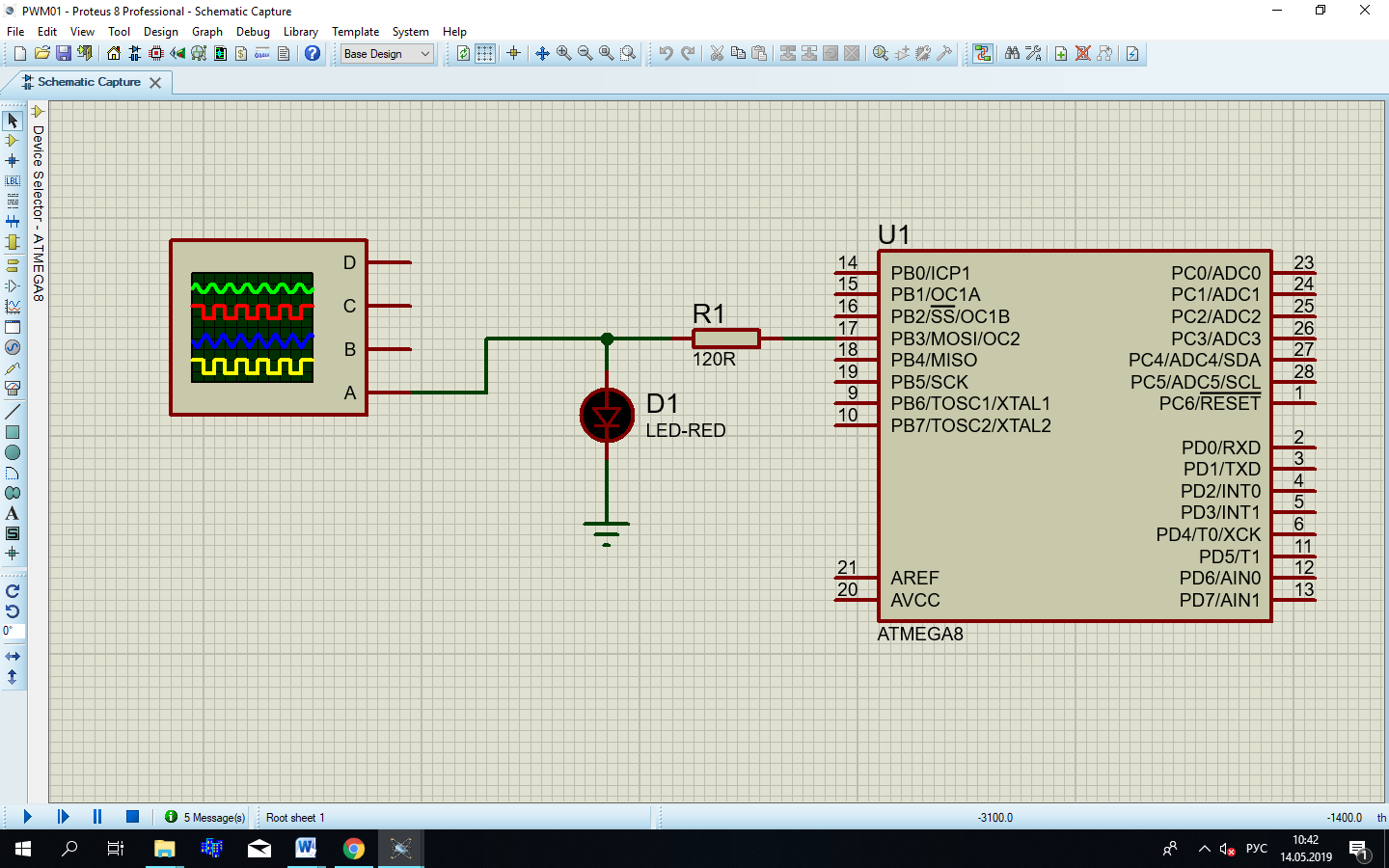
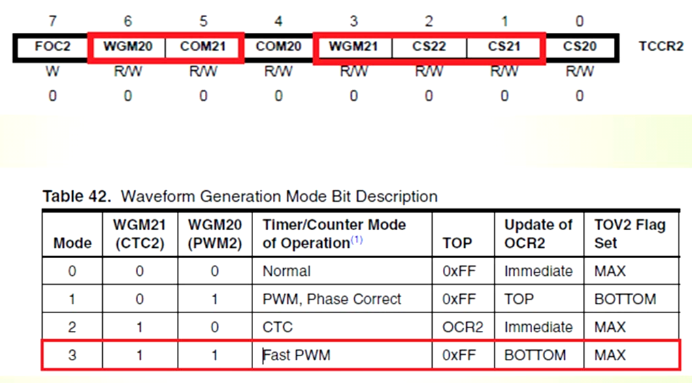
**Широтно-импульсная модуляция.**

В микроконтроллерах **AVR широтно-импульсную модуляцию** можно организовать, как **программно (работает процессор)**, так и **аппаратно (работает таймер-счетчик)**. Рассмотрим аппаратную организацию **ШИМ**в  **AVR**  на примере использования **таймера 2.**



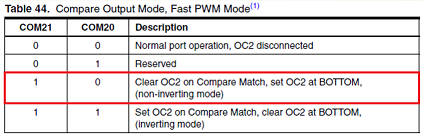
При использовании таймера 2 импульсы ШИМ выводятся на строго определенную ножку порта В (вывод 3, альтернативное обозначение **OC2)**,

Для запуска ШИМ необходимо настроить определенные биты в регистре **TCCR**  таймера.



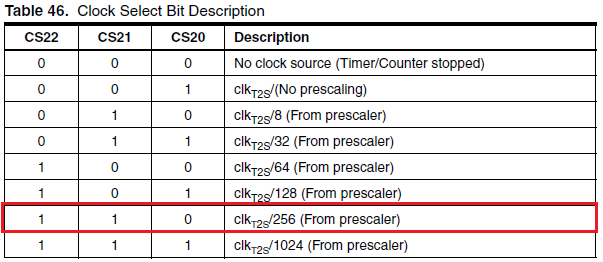
Биты WGM20(21) отвечают за включение режима ШИМ.

Биты COM20 и COM21, отвечают за режим работы самого ШИМ (вид выходного сигнала на выводе ОС2). Мы выберем прямой режим и соответственно ему и включим данные биты.



В данном режиме при начале счёта напряжение на ножке OC2 будет находиться в высоком логическом состоянии, а как только мы досчитаем до определённой цифры, то ножка перейдёт в низкое состояние и будет в нём находиться до окончания счёта.

Далее необходимо настроить делитель тактовой частоты. Выбираем коэффициент деления 256. За деление тактовой частоты отвечают биты CS.



**Регистр TIMSK не трогаем.** Регистр **OCR2** задаёт значение скважности. Пока зададим скважность 100.

OCR2 = 100; // задаем скважность импульсов 100:255

Дальше заносим 0 в регистр счёта **TCNT2**, то есть обнуляем счётчик.

**TCNT2=0x00; // Timer Value = 0.**

На этом программная настройка таймера закончена.

Для проверки работоспособности проекта используем следующий программный код.

int main(void)

{

port\_ini();

init\_PWM\_timer();

while (1)

{

}

}

В случае правильного кода светодиод должен постоянно светиться.

**Плавное программное управление ШИМ.**

Добавим код в бесконечный цикл, который будет управлять плавно скважностью от 0 до 100 процентов. Соответственно содержимое регистра OCR2 мы будем менять от 0 до 255.  Добавим глобальную переменную статуса ШИМ. Будет 2 вида статуса — нисходящий(декрементный) и восходящий ( инкрементный) — 0 и 1.

**Int pwm\_state=0;**

**Изначально переменная равна нулю и работает восходящий статус. Содержимое** OCR2 увеличивается до 255.

В этом случае происходит переключение статуса и включается режим уменьшения OCR2 до нуля. Плавность регулирования изменяется за счет временной задержки \_delay\_ms().

Визуально за работой ШИМ можно наблюдать через осциллограф.