Оглавление

1. Разработка функциональной схемы МПС

1.1. Микроконтроллер 80C54

Главным звеном разрабатываемой микропроцессорной системы является микроконтроллер 80C54. Данный микроконтроллер обладает характеристиками, изображенными в таблице 1.

Таблица 1

Ядро	8051
Частота, МГц	12, 24, 33
ROM, kB	16
RAM, B	256
Таймеры	3x16 bit
I/O	4x8 bit
V_{CC}, B	4.5 – 5.5
T, ºC	От -40 до 85

Сначала необходимо настроить микроконтроллер, задав ему частоту работы так, как показано на рисунке 1. Данная частота играет важную роль в правильном расчете времени таймером.

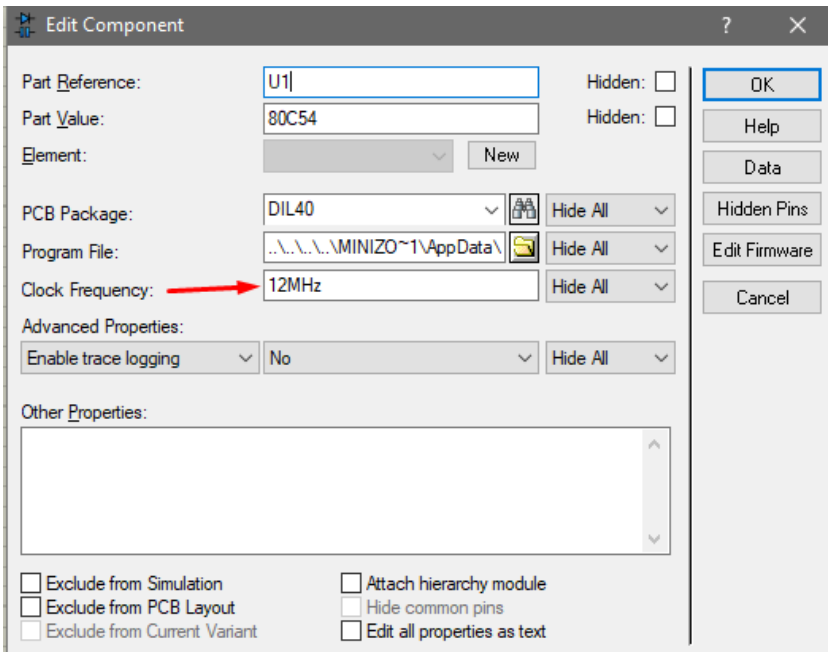

Рисунок 1 – Настройка микроконтроллера

Функциональное назначение микроконтроллера заключается в том, что он обрабатывает и выполняет все функции, заложенные в программу.

1.2. Дисплей LM016L

Далее необходимо подключить к микроконтроллеру дисплей LM016L. Данный дисплей имеет 2 строки по 16 символов в каждой. Он построен на базе HD44780 и, соответственно, выполняет все команды, заложенные в HD44780.

Чтобы подключить дисплей к контроллеру, необходимо соединить через шину порт P1, контроллера и порт D дисплея. По этой шине передаются команды дисплея или коды символов.

Далее необходимо подключить порты RS, RW, E дисплея к портам P3.5, P3.6 и P3.7 соответственно. На основе данных портов реализуется функция чтения/записи команд или символов в дисплее.

Под конец подключения дисплея, необходимо к VSS подсоединить землю, а к VDD и VEE – питание.

Общий вид подключения микроконтроллера к дисплею изображен на рисунке 2.

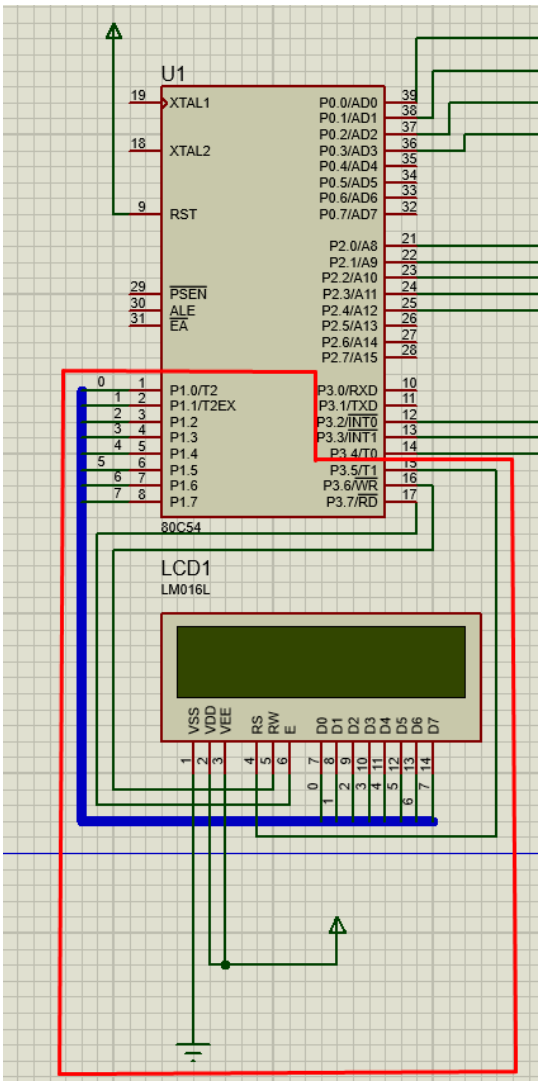

Рисунок 2 – Подключение дисплея к микроконтроллеру

Данный дисплей служит для отображения оператору времени действия красного и зеленного сигналов, активного цвета для настройки и режимов работы светофора.

1.3. Кнопки

Далее необходимо подключить кнопки. Первые 4 кнопки стоит подключить, как матричную клавиатуру, так как показано на рисунке 3.

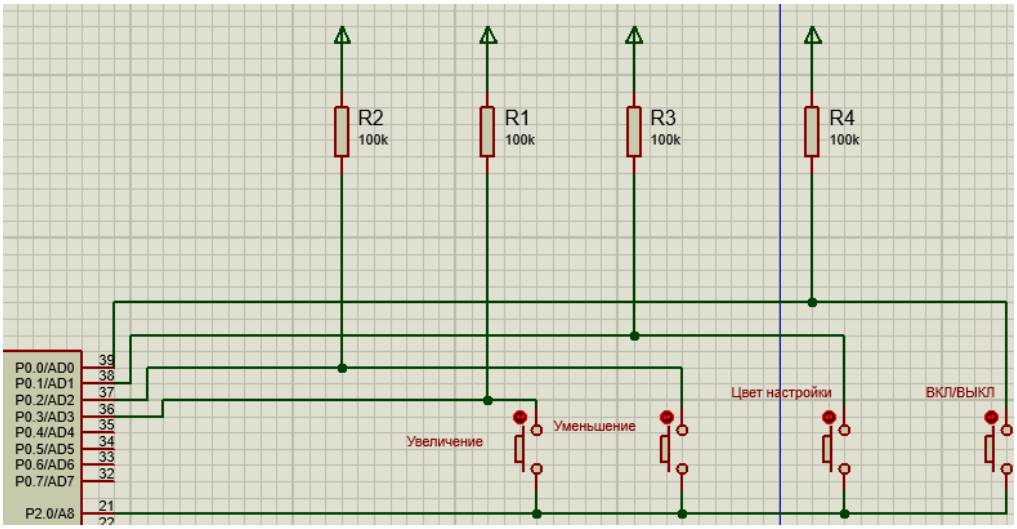

Рисунок 3 – Подключение кнопок на основе матричной клавиатуры

В данной части схемы используются подтягивающие резисторы, т.к. в порт P0 они не встроены по умолчанию, в отличии от портов P1, P2, P3 [1].

Две оставшиеся кнопки необходимо подключить к портам с внешними прерываниями так, как показано на рисунке 4.

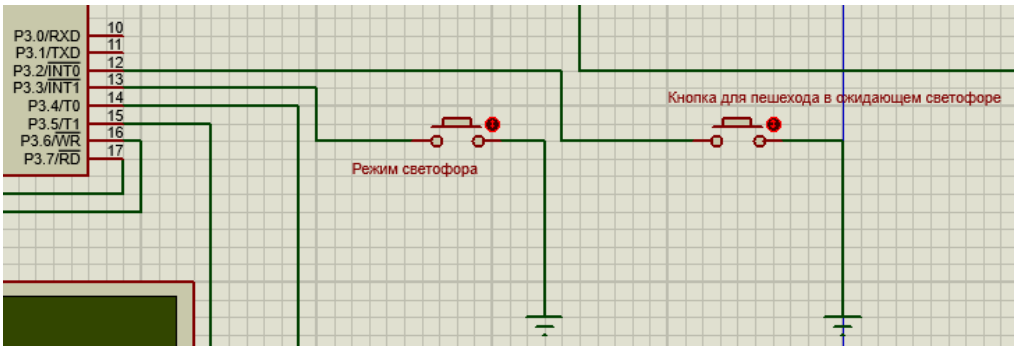

Рисунок 4 – Подключения кнопок к портам для обработки внешних прерывания

Данные кнопки служат для активации определенных функций, заложенных в микроконтроллер. Например, для изменения времени действия красного и зеленого сигналов.

1.4. Speaker

Далее необходимо подключить динамик Speaker. Подключается он следующим образом: к одному входу подключается микроконтроллер, а к другому земля. Подключение изображено на рисунке 5.

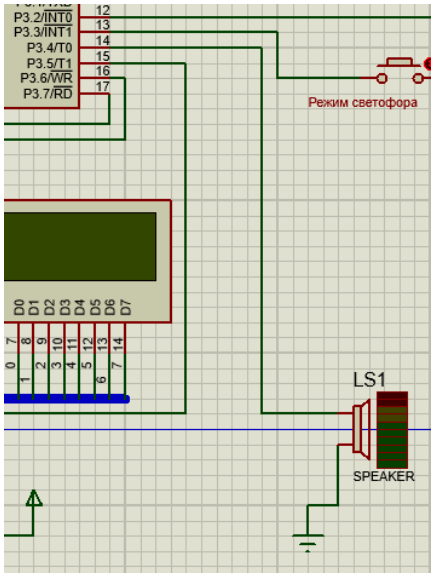

Рисунок 5 – Подключение динамика

Также необходимо задать характеристики динамика так, чтобы звук из него не был слишком громким и оглушающим для человека. Заданные характеристики изображены на рисунке 6.

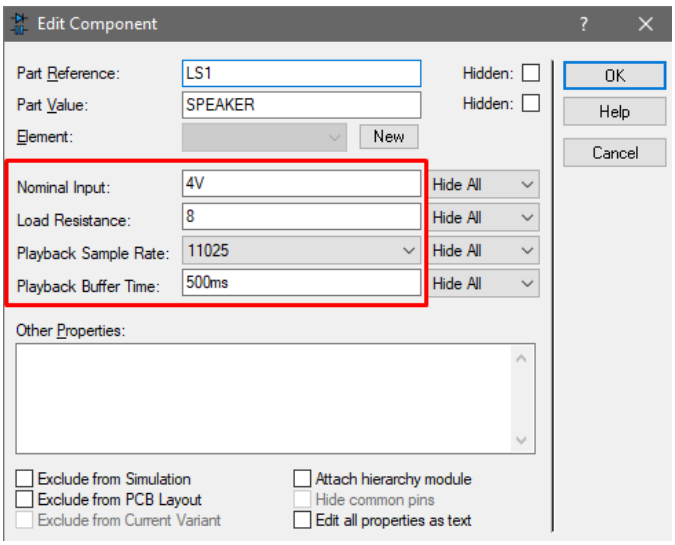

Рисунок 6 – Свойства динамика

Данные динамик служит для создания звукового сигнала при наступлении зеленного цвета у пешеходов.

1.5. Traffic Lights

Далее необходимо подключить светофоры к микроконтроллеру, причем это необходимо сделать так, чтобы при подаче сигнала на красный цвет автомобильного светофора, загорался зеленый цвет у пешеходного светофора, а при подаче сигнала на желтый или зеленый цвет автомобильного светофора, загорался красный у пешеходов. Для этого необходимо подключить два Traffic Lights к микроконтроллеру так, как изображено на рисунке 7.

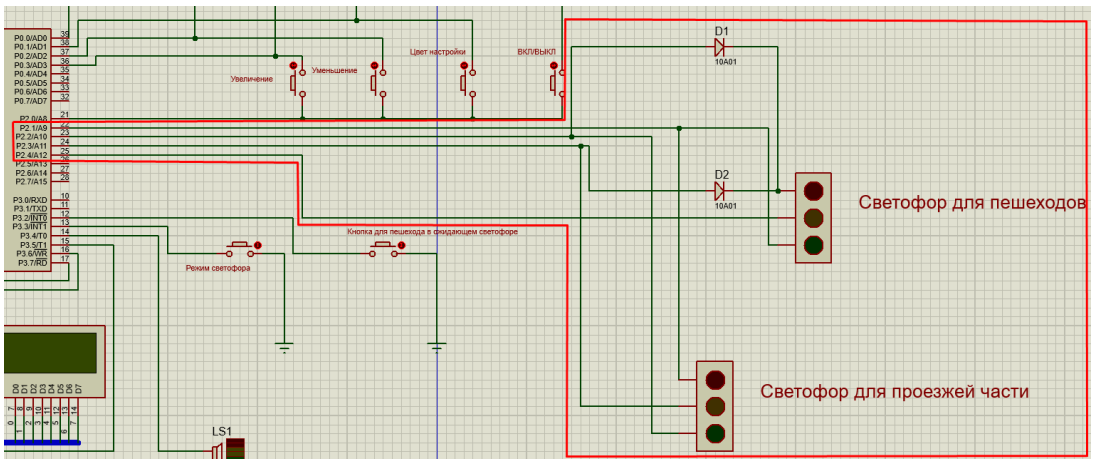

Рисунок 7 – Подключение светофоров.

Здесь используются диоды с максимальным напряжением пробоя в 50 В, чтобы сигнал не уходил в обратную сторону и не нарушал правильную работу светофоров.

Traffic Lights служат для цветового оповещения пешеходов и автомобильного транспорта о разрешении или запрете передвижения по проезжей части.

2. Разработка программы для микроконтроллера

2.1. Переменные

Создание программы начинается с объявления ячеек памяти или, как их еще можно назвать, переменных, которые будут использоваться в программе. Для удобства они разделяются на определенные группы: для кнопок, для светофора, для ЖКИ, для таймера, для карты состояний клавиатуры.

В группе “Для кнопок” объявляются переменные для защиты от зажатия и переменные, которые участвуют в реализации функций светофора и которые требуется изменять после нажатия на определенную кнопку.

В группе “Для светофора” объявляются переменные для реализации работы сигналов светофоров: время действия сигналов, направление смены цветов, активный в данный момент цвет и разрешение на активацию звукового сигнала.

В группе “Для ЖКИ” объявляются переменные для взаимодействия с дисплеем и переменные, отвечающие за вывод символов времени действия сигналов, активного цвета для настройки, режима работы и режима питания.

В группе “Для таймера” объявляются переменные, отвечающие за время таймера.

2.2. Подготовка системы

Далее идет подготовка системы, которая начинается с инициализации переменных, чтобы задать стандартные настройки системы:

- Светофоры должны быть выключены
- Включена настройка красного цвета
- Включен классический режим работы
- Отключены защиты от зажатий

- Установлено направление смены цвета автомобильного светофора с красного до зеленого
- Установлены время действия красного, желтого и зеленого сигналов для автомобильного светофора. Красный и зеленый сигналы устанавливаются в свое минимальное время работы (10 и 30 секунд), а желтый в 3 секунды.
- Установлены символы для времени действия красного, зеленого сигналов.
- Установка таймера 0 в режим работы 2 без разрешения внешнего прерывания 0.
- Значения в регистры TH0 и TL0 для счета сотней мкс.
- Разрешения на прерывания от таймера 0 и внешнее прерывание 1.
- Запрет на внешнее прерывание 0 и активацию таймера 0.

Сразу после инициализации переменный стоит подготовить дисплей, настроив его на 8-ми битный режим, на автоматическое перемещение курсора и на вывод фиксированной части двух строк, занесенных в памяти программы.

После первоначальной подготовки уже должна бесконечно обрабатываться основная часть программы, реализующая опрос клавиатуры и работу светофора в зависимости от режима.

В общем виде эту основную часть с первоначальной подготовкой можно представить обобщенной блок-схемой, изображенной на рисунке 8

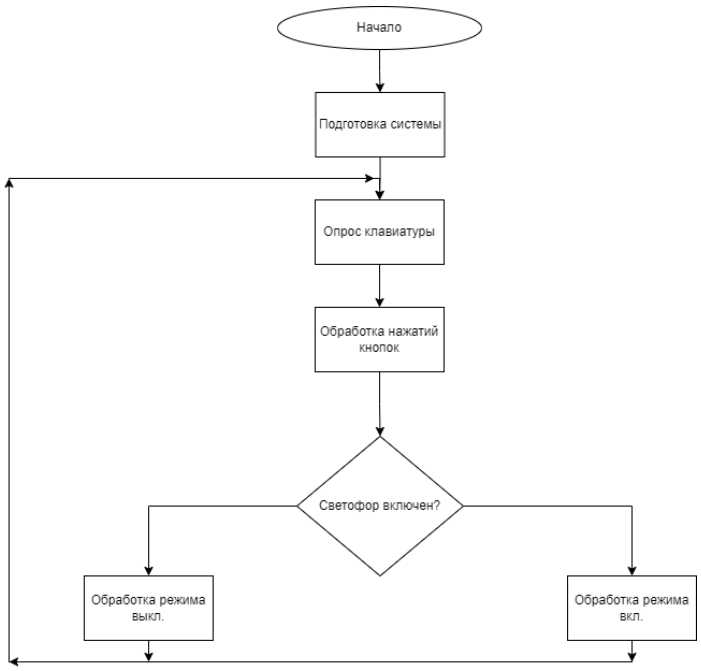

Рисунок 8 – Основная часть программы в виде блок-схемы

2.3. Опрос клавиатуры

Опрос клавиатуры – это операция, предназначенная для определения того, какая кнопка в матричной клавиатуре была нажата.

Так, в нашем случае, строка клавиатуры подключена к выводу P2.0 Данный порт настроен на выход и в начальном состоянии на нем подано напряжение логического нуля, т.к. в данном проекте клавиатура состоит только из одной строки.

Столбцы же подключены к выводам P0.0, P0.1, P0.2, P0.3. Они настроены на чтение, причем эти линии имеют напряжение логической единицы, образованное с помощью подтягивающих резисторов.

Процедура сканирования клавиатуры выглядит следующим образом. Подается 0 на строку, которую требуется опросить. В данном случае в порт P2.0. Далее происходит проверка состояния портов столбцов P0.0, P0.1, P0.2, P0.3, результат которой записывается, как карта состояний клавиатуры.

Данная карта в дальнейшем будет использоваться для обработки нажатий кнопок.

Обобщенная блок схема опроса клавиатуры изображена на рисунке 9.

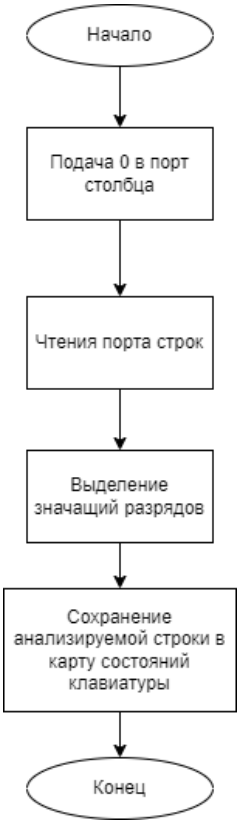

Рисунок 9 – Обобщенная блок схема опроса клавиатуры

Если на каком-то из портов установлен 0, значит, на клавиатуре в данный момент нажата кнопка, номер которой соответствует номеру бита порта P0, в котором был обнаружен 0.

2.4. Обработка нажатий кнопок

Обработка нажатий кнопок начинается с анализа карты состояний, полученной в результате опроса клавиатуры.

2.4.1. Первая кнопка (переключение режима питания)

Так сначала анализируется нулевой бит. Данный бит отвечает за состояние кнопки, которая после нажатия должна переключать режим питания.

Если в нулевом бите 1, т.е. кнопка не нажата, то происходит отключения защиты от зажатия кнопки и переход к обработке следующей кнопки.

Если же в нулевом бите 0, то дальше должна идти проверка на защиту от зажатия.

Если защита не активирована, то выполняются следующие действия: переключение режима питания, переключение таймера, включение защиты от зажатия и обновление режима питания на дисплее, которое состоит из изменения ASCII-кода символа питания и занесения информации в дисплей.

Если же защита активирована, то просто происходит переход к обработке следующей кнопки.

В общем виде обработка первой кнопки изображена на рисунке 10.

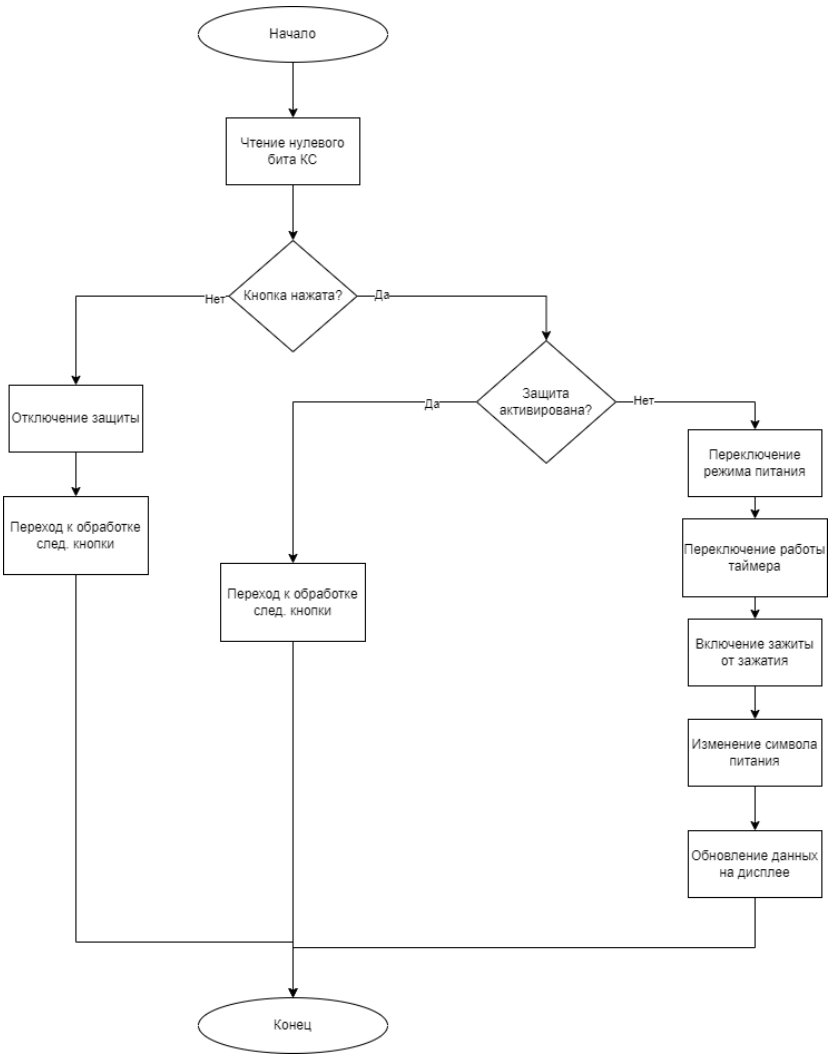

Рисунок 10 – Блок схема обработки первой кнопки

Дальнейший анализ кнопок будет происходить только тогда, когда режим питания находится в состоянии 0, т.к. выкл.

2.4.2. Вторая кнопка (переключение цвета для настройки)

Далее анализируется первый бит карты состояний. Данный бит отвечает за состояние кнопки, которая после нажатия должна переключать активный цвет для настройки.

Если в первом бите 1, т.е. кнопка не нажата, то происходит отключения защиты от зажатия кнопки и переход к обработке следующей кнопки.

Если же в первом бите 0, то дальше должна идти проверка на защиту от зажатия.

Если защита не активирована, то выполняются следующие действия: переключение активного цвета для настройки, включение защиты от зажатия и обновление активного цвета на дисплее, которое состоит из изменения ASCII-кода символа питания и занесения информации в дисплей.

Если же защита активирована, то просто происходит переход к обработке следующей кнопки.

В общем виде обработка второй кнопки изображена на рисунке 11.

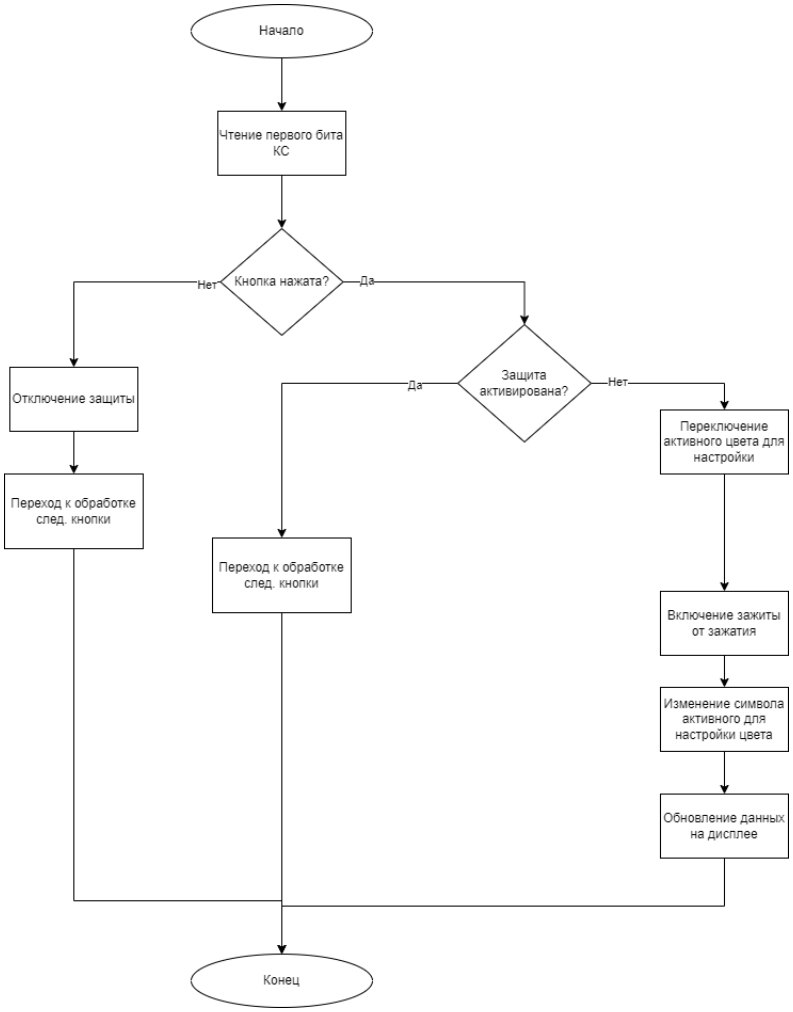

Рисунок 11 – Блок-схема обработки второй кнопки

2.4.3. Третья кнопка (уменьшение времени действия)

Далее анализируется второй бит карты состояний. Данный бит отвечает за состояние кнопки, которая после нажатия должна уменьшать время действия активного для настройки цвета, причем уменьшение не должно уходить за минимум для цвета.

Если в втором бите 1, т.е. кнопка не нажата, то происходит отключения защиты от зажатия кнопки и переход к обработке следующей кнопки.

Если же в втором бите 0, то дальше должна идти проверка на защиту от зажатия.

Если защита не активирована, то выполняются следующие действия: проверка текущего активного цвета, проверка на то, установлено ли уже время действия в минимум, уменьшение времени при текущем времени большем, чем минимум для цвета, и обновление времени действия на дисплее, которое состоит из изменения ASCII-кода символа питания и занесения информации в дисплей.

Если же защита активирована, то просто происходит переход к обработке следующей кнопки.

В общем виде обработка третьей кнопки изображена на рисунке 12.

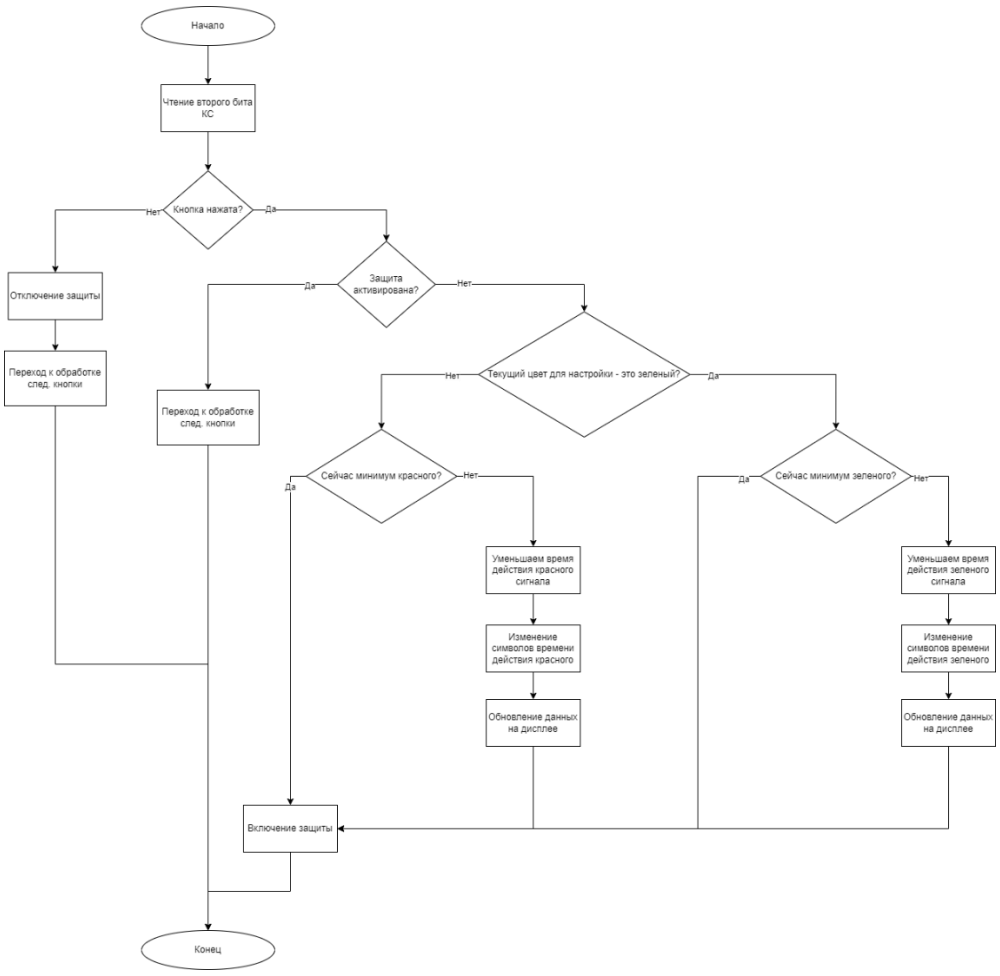

Рисунок 12 – Блок-схема обработки третьей кнопки

2.4.4. Четвертая кнопка (увеличение времени действия)

Далее анализируется третий бит карты состояний. Данный бит отвечает за состояние кнопки, которая после нажатия должна увеличивать время действия активного для настройки цвета, причем увеличение не должно уходить за максимум для цвета.

Если в третьем бите 1, т.е. кнопка не нажата, то происходит отключения защиты от зажатия кнопки и переход к обработке режима вкл или выкл.

Если же в третьем бите 0, то дальше должна идти проверка на защиту от зажатия.

Если защита не активирована, то выполняются следующие действия: проверка текущего активного цвета, проверка на то, установлено ли уже время действия в максимум, увеличение времени при текущем времени меньшем, чем максимум для цвета, и обновление времени действия на дисплее, которое состоит из изменения ASCII-кода символа питания и занесения информации в дисплей.

Если же защита активирована, то просто происходит переход к обработке режима вкл или выкл.

В общем виде обработка первой кнопки изображена на рисунке 13.

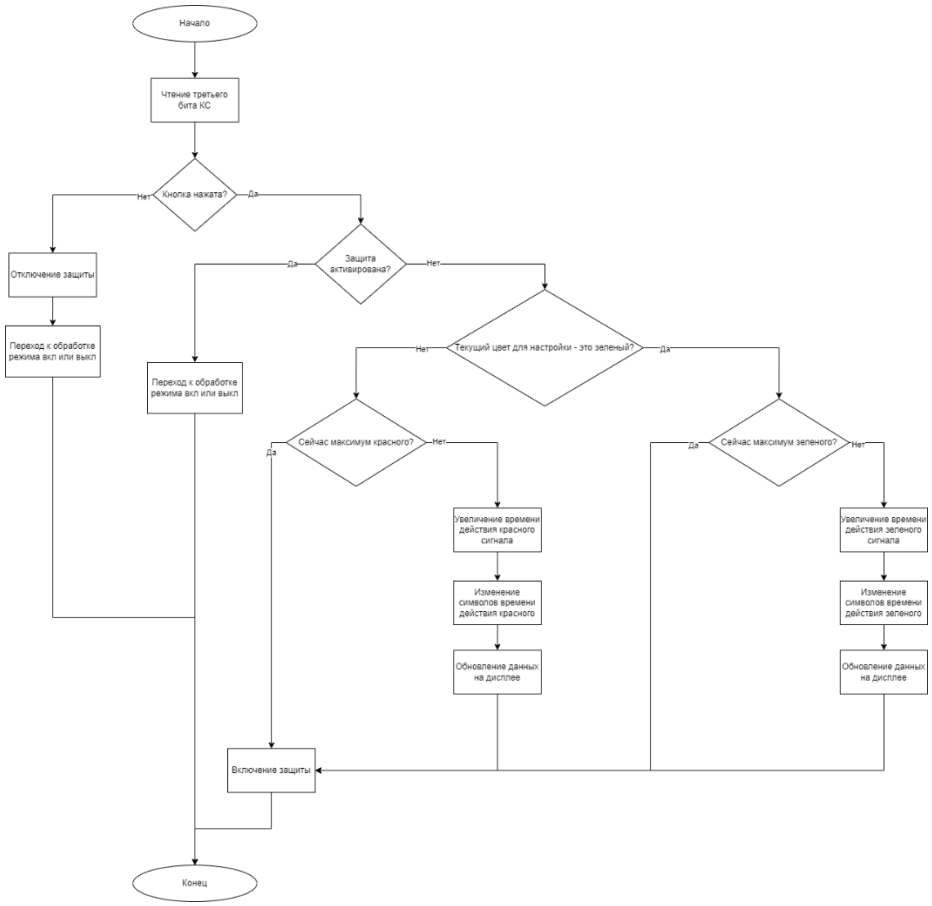

Рисунок 13 – Блок-схема обработки четвертой кнопки

Далее в зависимости от режима питания идет обработка режима вкл или режима выкл.

2.5. Обработка режима вкл

Режим состоит из обычного вызова подпрограммы, реализующей работу индикаторов светофора.

Обработка режима вкл начинается с анализа режима работы система. Если режим стандартный, то идет анализ ячейки памяти, содержащей номер текущего активного цвета. Значение в данной ячейки определяет прерывание от таймера 0.

Если цвет красный, то идет активация только красного цвета на светофоре для автомобилей и срабатывание периодического звукового сигнала, т.к. в этот момент в светофоре для пешеход горит зеленый. Звуковой же сигнал активируется обычным переключение соединенного с динамиком порта с 0 на 1 и обратно на 0.

Если цвет желтый, то идет активация только желтого цвета на светофоре для автомобилей.

Если цвет зеленый, то, соответственно, идет активация только зеленого цвета на светофоре для автомобилей.

Если же активирован ожидающий режим и при этом пешеход не нажал кнопку, то идет активация зеленого цвета для автомобилей, т.к. в ожидающем режиме всегда горит зеленый цвет для автомобилей до нажатия пешеходом на кнопку.

Если же активирован ожидающий режим и при этом пешеход нажал кнопку, то дальше идет такой же анализ, как и при стандартном режиме.

Обобщенная блок-схема изображена на рисунке 14.

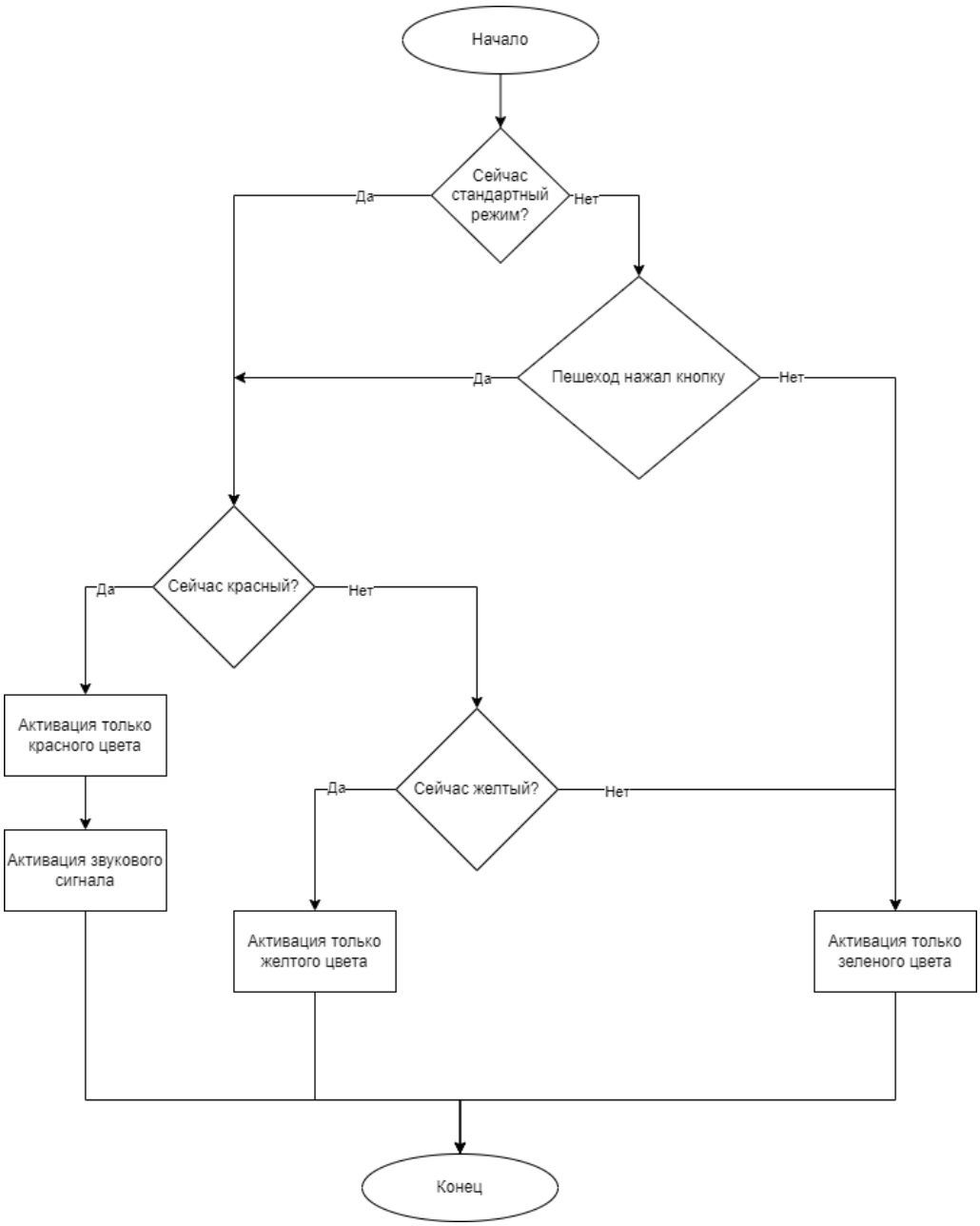

Рисунок 14 – Обобщенная блок-схема обработки режима вкл

2.6. Обработка режима выкл

Обработка данного режима состоит из простых операций: выключения индикаторов светофора, выключения динамика, выключения таймера и обнуления таймера.

Обобщенная блок-схема изображена на рисунке 15.

Рисунок 15 – Обобщенная блок схема режима выкл

2.7. Прерывания

2.7.1. Прерывание от таймера 0

При активации таймера в режиме происходит инкрементирование регистра TL0 каждый машинный цикл. Регистр TL0 определяет сколько времени прошло и, т.к. в процессе инициализации в него было занесено значение 156h, данный регистр должен переполняться при частоте микроконтроллера в 12 МГц каждые десятые доли мс, т.е. каждые 100 мкс. При переполнении данного регистра и срабатывает прерывание от таймера 0.

Данное прерывание в данной микропроцессорной системе должно просто считать время и переключать сигналы, в зависимости от текущего насчитанного времени, от текущего активного цвета для автомобильного светофора и от направления смены цвета.

В общем виде прерывание от таймера изображено в виде блок-схемы на рисунке 16.

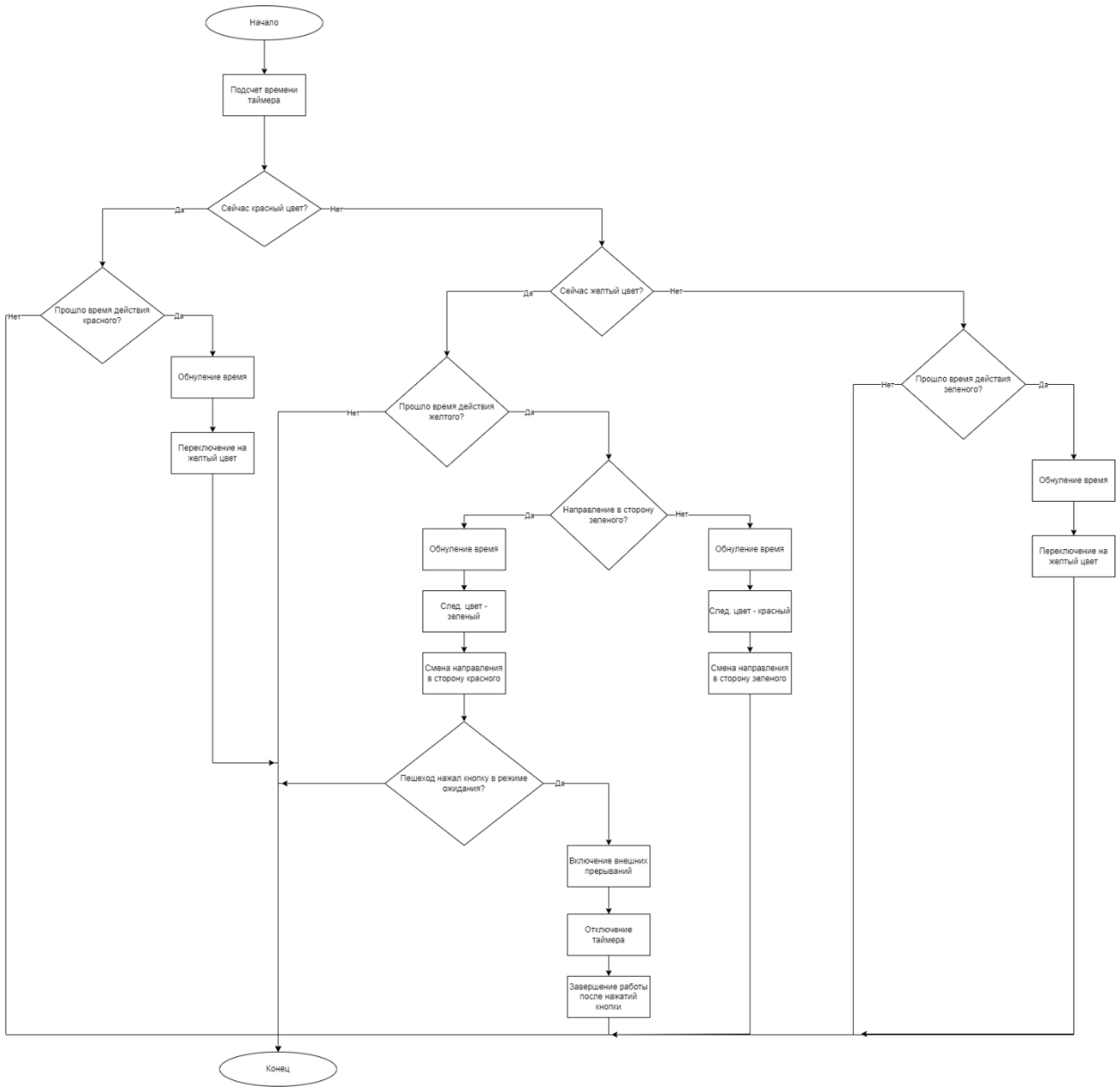

Рисунок 16 – Обобщенная блок-схема прерывания от таймера 0

2.7.2. Внешнее прерывание 0

Внешнее прерывание 0 срабатывает при нажатии в ожидающем режиме пешеходом на кнопку. Данное прерывание запускает смену сигналов в режиме ожидания, чтобы пешеход мог перейти дорогу, с помощью обнуления таймера, запуска таймера и, собственно, активации сигналов в режиме ожидания.

Обобщенная блок-схема прерывания изображена на рисунке 17.

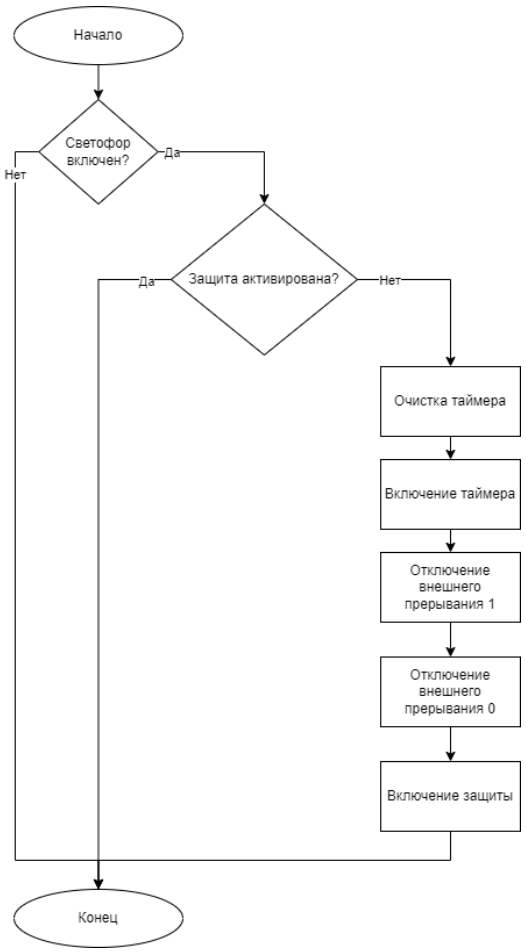

Рисунок 17 – Обобщенная блок схема внешнего прерывания 0

2.7.3. Внешнее прерывание 1

Внешнее прерывание 1 срабатывает при нажатии оператором на определенную кнопку, предназначенную для смены режима работы светофора. Данное прерывание переключает таймер, разрешение на внешнее прерывание 0 и, собственно, сам режим работы.

Также прерывание обнуляет таймер, обновляет символ режима на дисплее и подготавливает систему к нажатию пешеходом на кнопку, задав активный зеленый свет для автомобилей и указав направления смены цвета, чтобы после того, как пешеход нажал на кнопку, цвет менялся с зеленого в сторону красного для автомобилей и обратно.

Обобщенная блок схема изображена на рисунке 18.

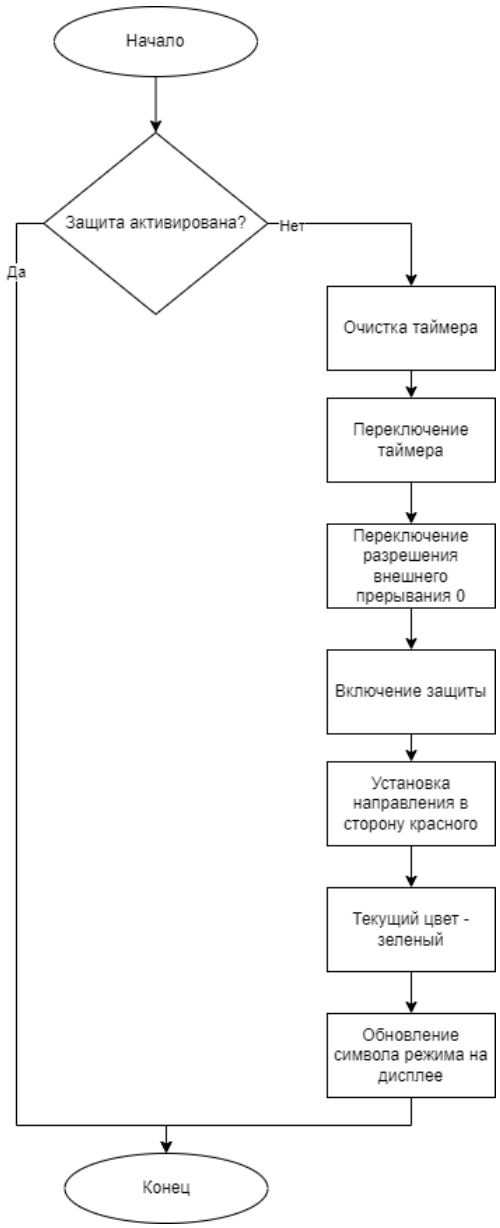

Рисунок 18 – Обобщенная блок-схема внешнего прерывания 1

3. Демонстрация работы МПС

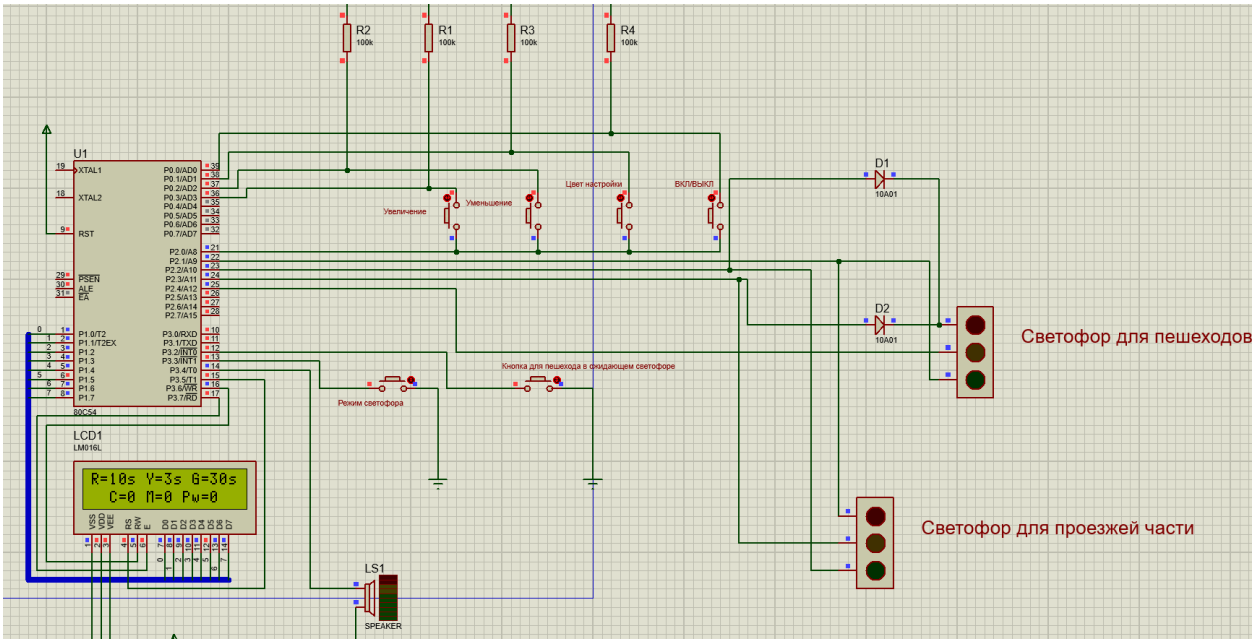

Рисунок 19 – Микропроцессорная система до запуска

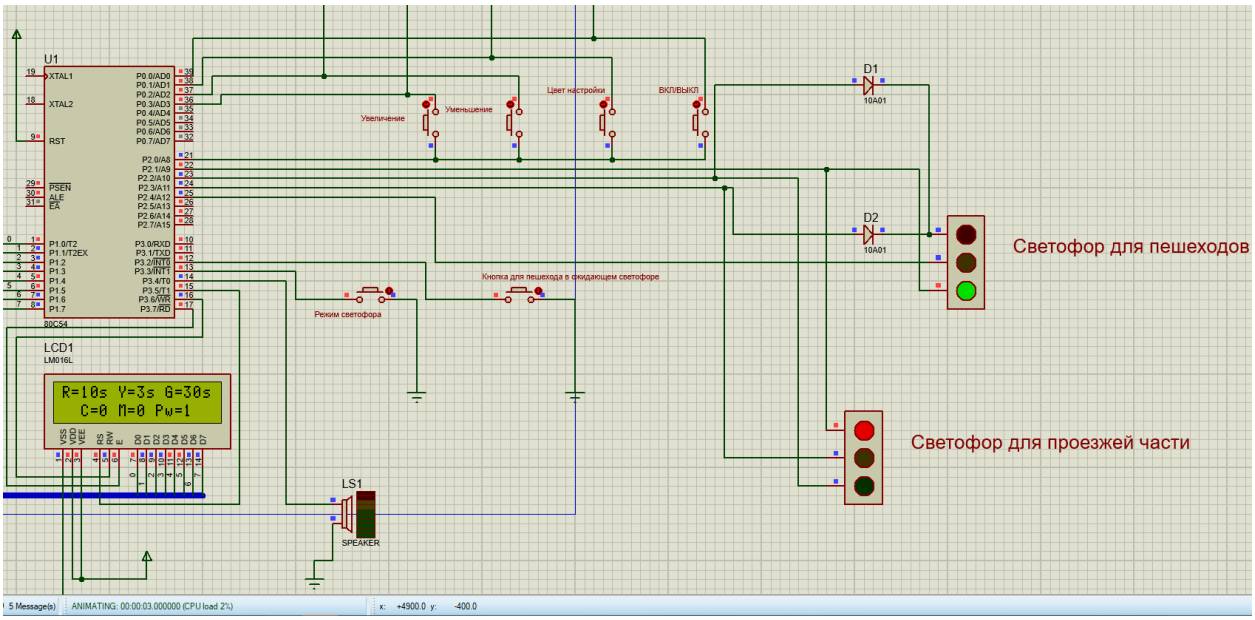

Рисунок 20 – Система после запуска в классическом режиме работы

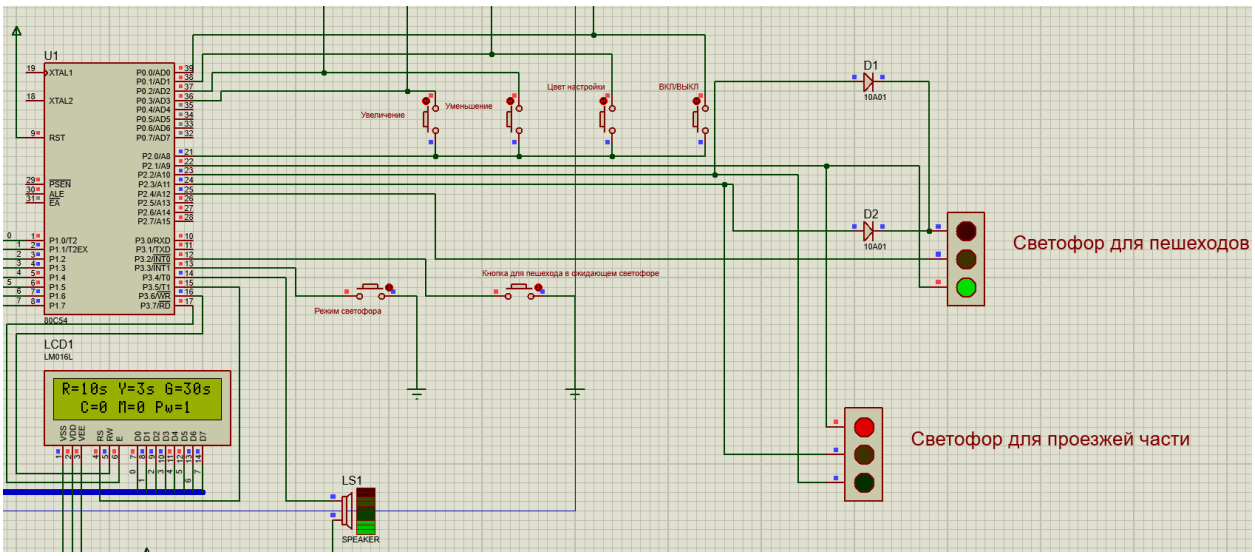

Рисунок 21 – Звуковой сигнал во время зеленного цвета у пешеходов

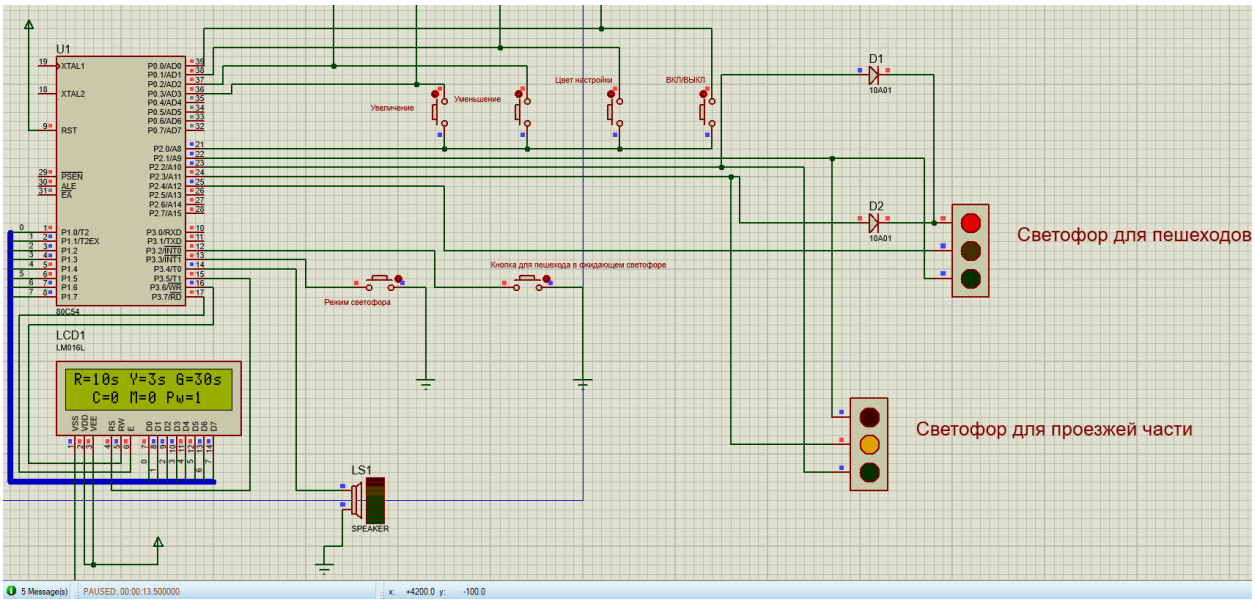

Рисунок 22 – Переключение с красного на желтый сигнал для проезжей части в классическом режиме работы

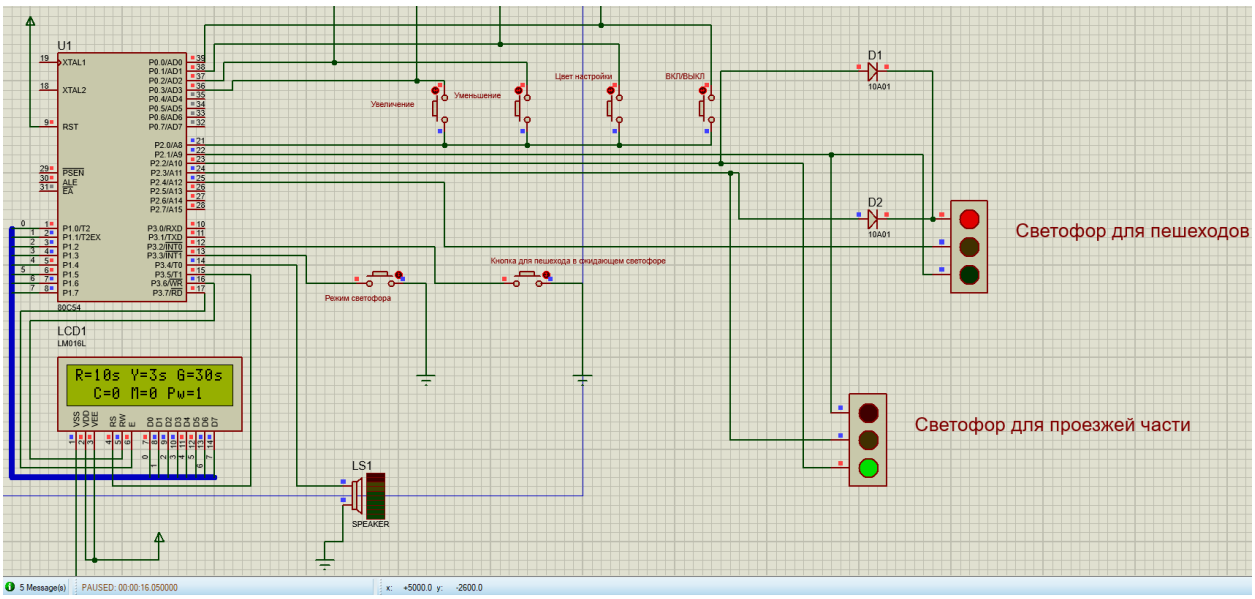

Рисунок 23 – Переключение с желтого на зеленый сигнал для проезжей части в классическом режиме работы

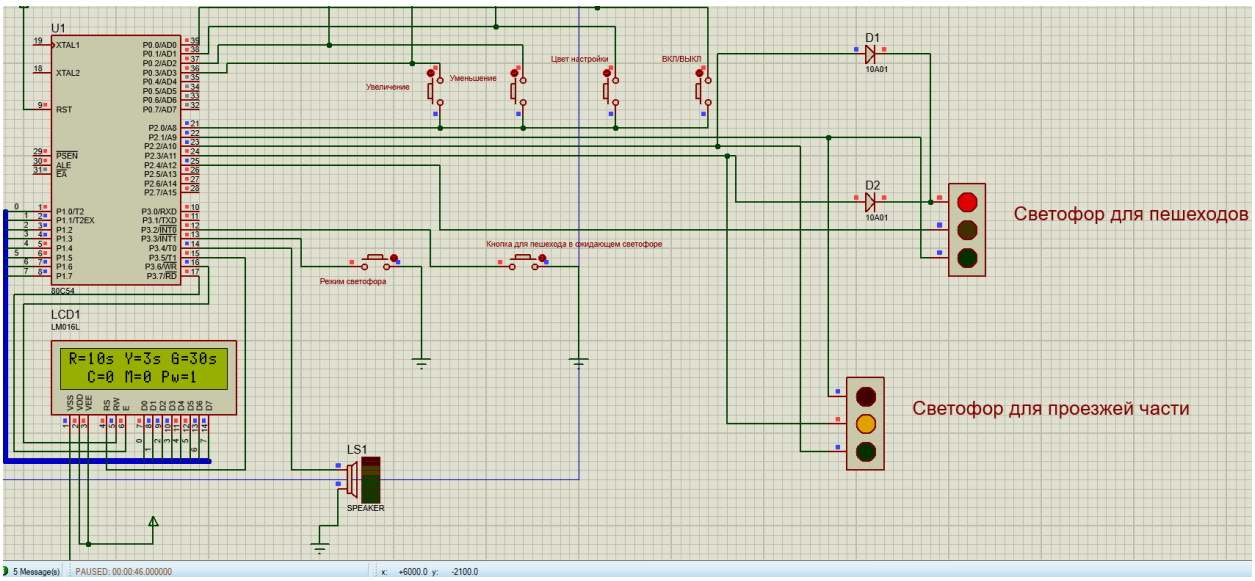

Рисунок 24 – Переключение с зеленого на желтый сигнал для проезжей части в классическом режиме работы

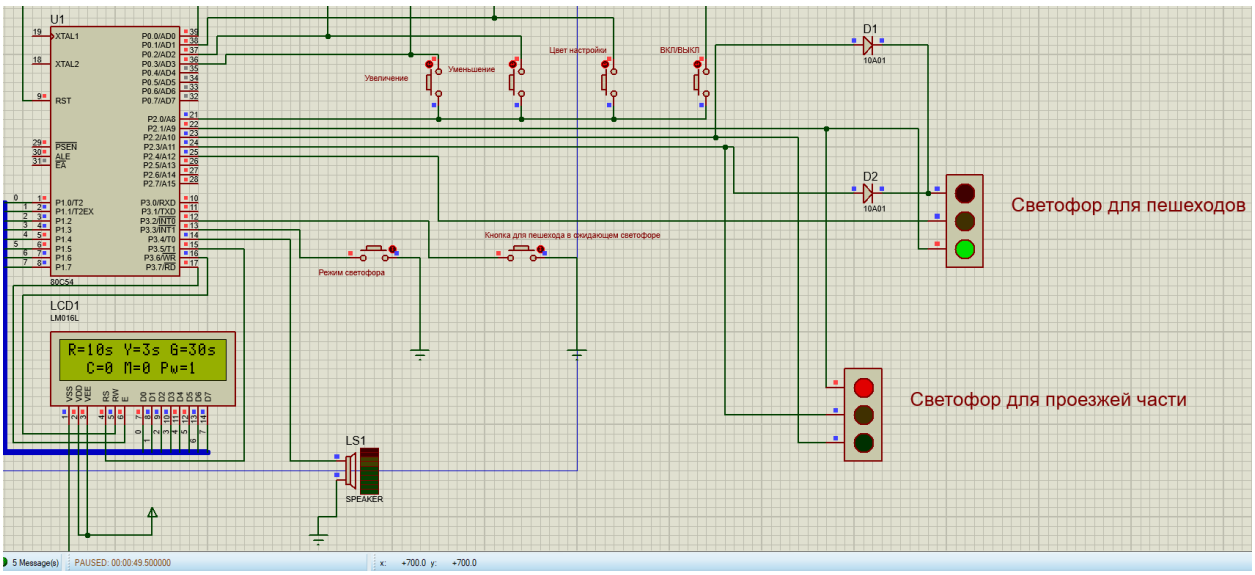

Рисунок 25 – Переключение с желтого на красный сигнал для проезжей части в классическом режиме работы

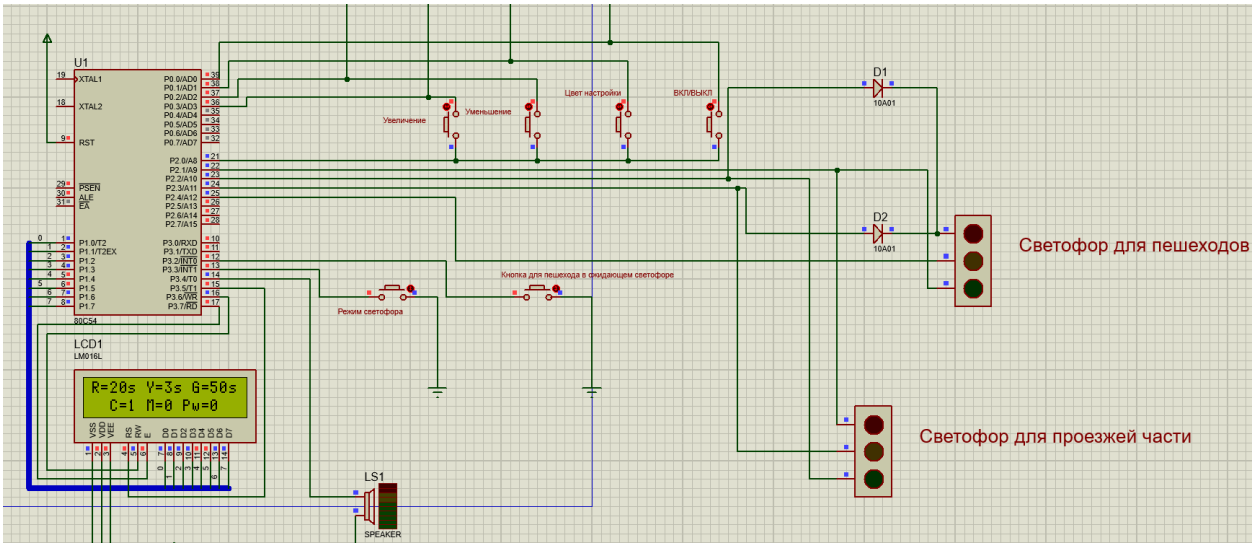

Рисунок 26 – Настройка длительности красного и зеленого сигналов в светофоре для проезжей части в выключенном состоянии

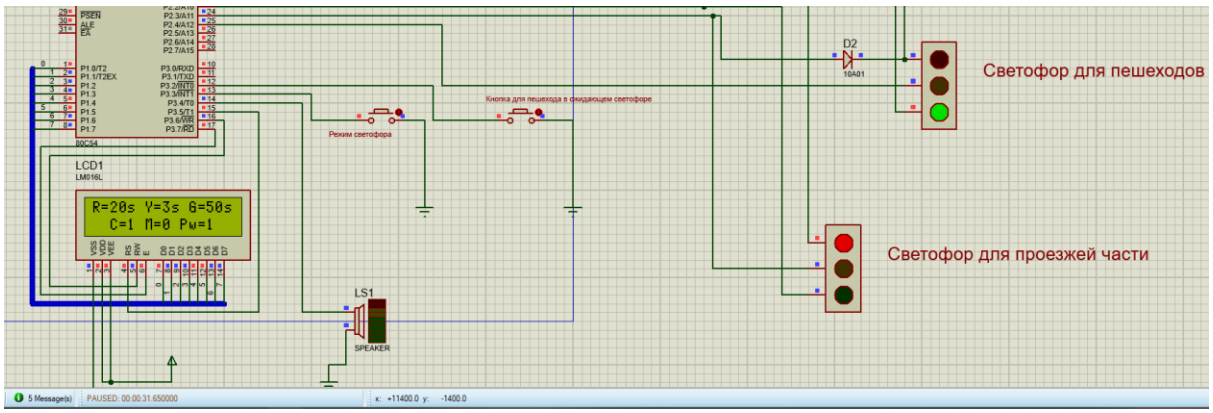

Рисунок 27 – Пуск системы в обычном режиме после настройки времени действия цветов.

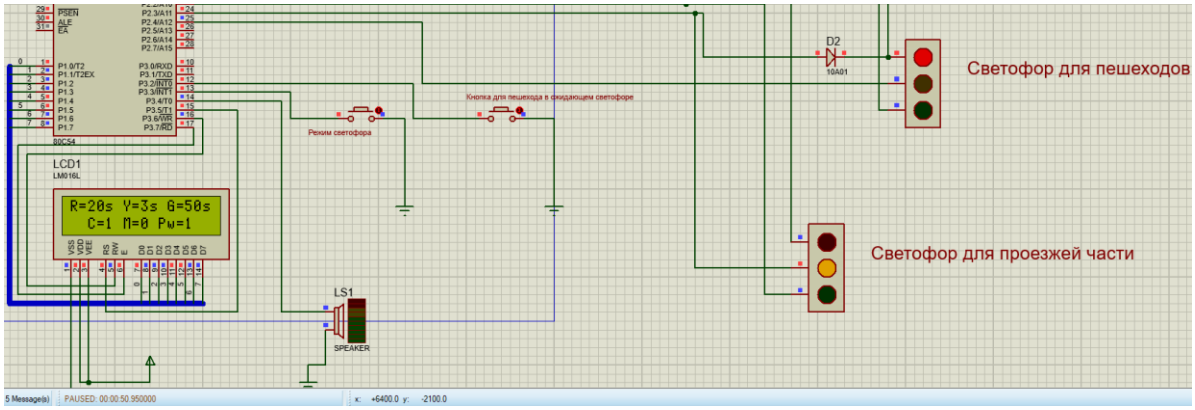

Рисунок 28 – Смена цвета с красного на желтый спустя настроенные 20 секунд

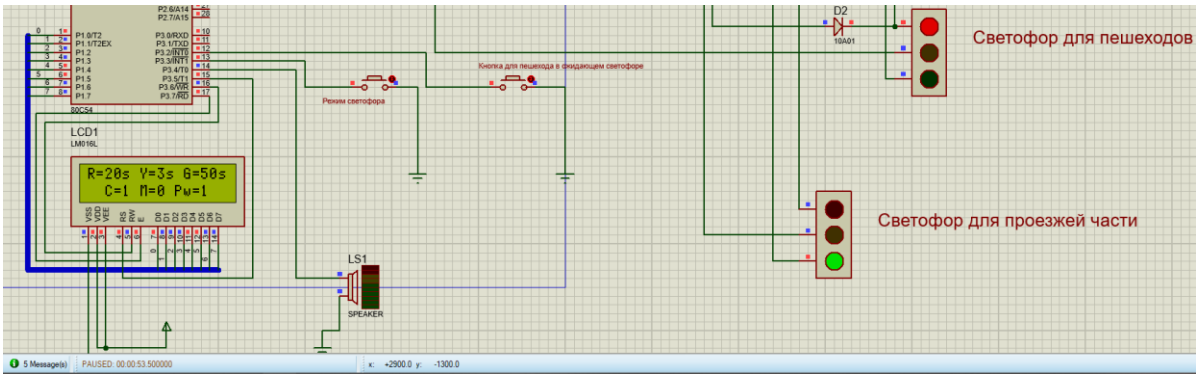

Рисунок 29 – Смена цвета с желтого на зеленый спустя стандартные 3 секунд

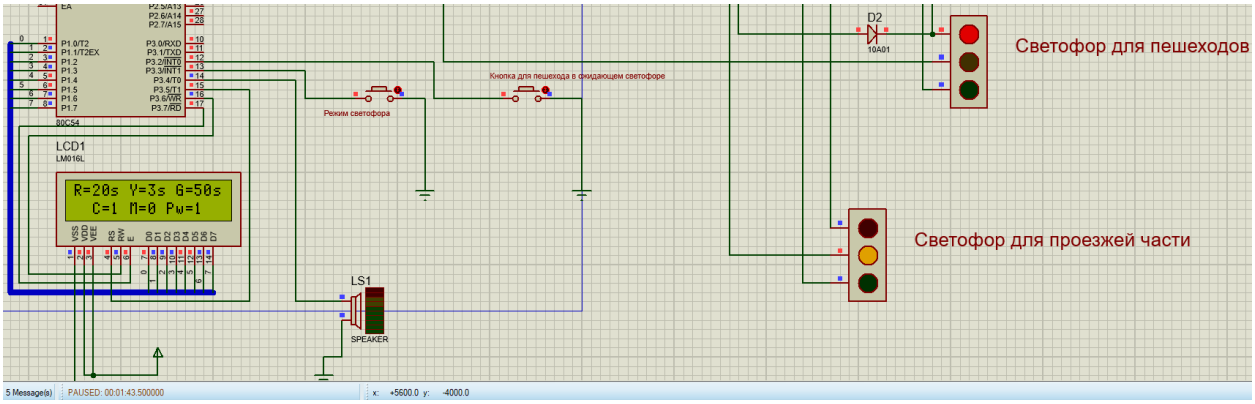

Рисунок 30 – Смена цвета с зеленого на желтый спустя настроенные 50 секунд.

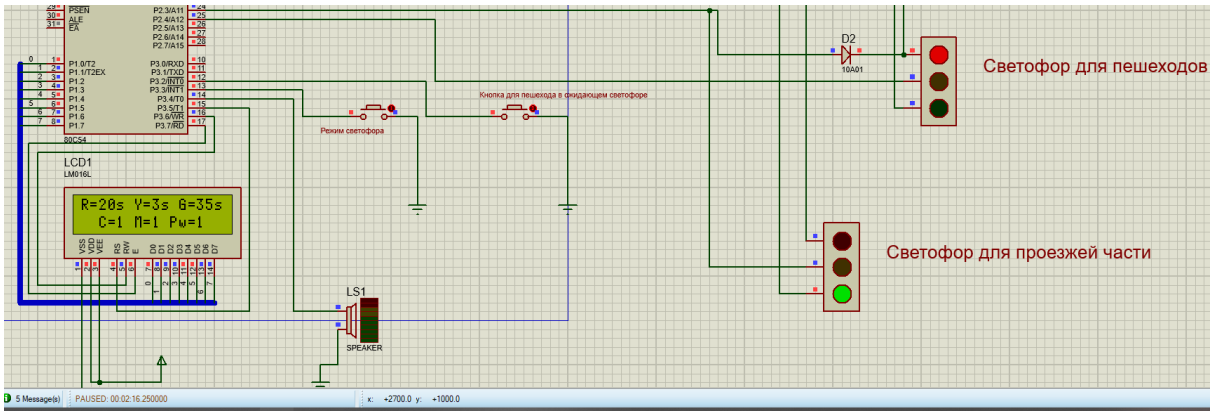

Рисунок 31 – Переход светофорам в ожидающий режим

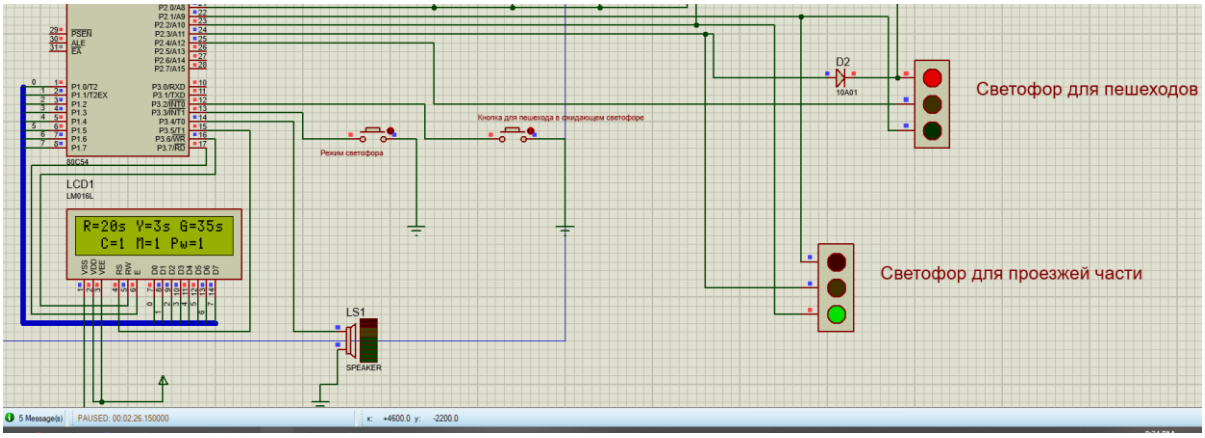

Рисунок 32 – Нажатие пешеходом кнопки для перехода

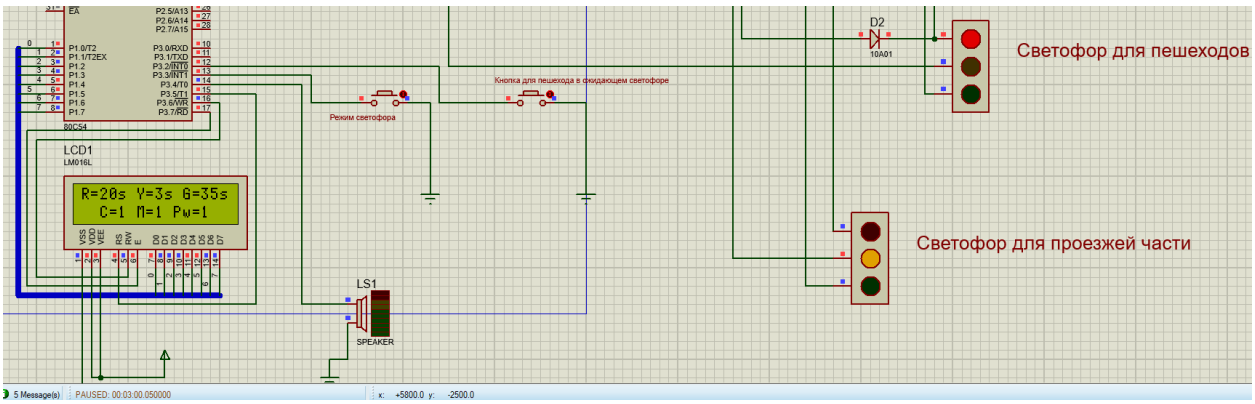

Рисунок 33 – Смена цвета в ожидающем режиме после нажатия кнопки пешеходом спустя настроенные 35 секунд

Список использованных источников:

1. INTEL. 8XC52/54/58 Chmos single-chip 8-bit microcontroller, 23 p.