

**МИНЕСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ
УЗБЕКИСТАН
НУКУССКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
ИМЕНИ МУХАММАДА АЛЬ-ХОРЕЗМИ**



Направление: _____

Курсовая Работа

По предмету _____
На тему _____

Выполнил(а):
Принял(а):

Жумамуратов У.

НУКУС 2018

Курсовая работа

Микроконтроллер, громкоговоритель, Термодатчик,
Жидкокристаллический индикатор, ИК- ЛУЧИ

Цель работы: разработка микропроцессорной системы на базе микроконтроллера, автомобильных часов-термометра-вольтметра.

Содержание работы: в работе выполнено построение структурной схемы, сформирован алгоритм работы системы, выбор элементной базы, оптимальной для реализации поставленных задач по диапазону характеристик, разработана программа, разработана принципиальная схема устройства.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1. Описание объекта и функциональная спецификация
2. Описание структуры системы
3. Описание ресурсов МК AT89C2051
4. Ассемблирование
5. Разработка алгоритма работы устройства
6. Описание выбора элементной базы и работы принципиальной схемы

ЗАКЛЮЧЕНИЕ

СПИСОК ЛИТЕРАТУРЫ

ПРИЛОЖЕНИЕ А. ЛИСТИНГ ПРОГРАММЫ И ОБЪЕКТНЫЙ ФАЙЛ

ВВЕДЕНИЕ

Современную микроэлектронику трудно представить без такой важной составляющей, как микроконтроллеры. Микроконтроллеры незаметно завоевали весь мир. Микроконтроллерные технологии очень эффективны. Одно и то же устройство, которое раньше собиралось на традиционных элементах, будучи собрано с применением микроконтроллеров, становится проще, не требует регулировки и меньше по размерам. С применением микроконтроллеров появляются практически безграничные возможности по добавлению новых потребительских функций и возможностей к уже существующим устройствам. Для этого достаточно просто изменить программу.

Однокристалльные (однокорпусные) микроконтроллеры представляют собой приборы, конструктивно выполненные в виде БИС и включающие в себя следующие составные части: микропроцессор, память программ и память данных, а также программируемые интерфейсные схемы для связи с внешней средой.

Мировая промышленность выпускает огромную номенклатуру микроконтроллеров. По области применения их можно разделить на два класса: специализированные, предназначенные для применения в какой-либо одной конкретной области (контроллер для телевизора, контроллер для модема, контроллер для компьютерной мышки) и универсальные, которые не имеют конкретной специализации и могут применяться в самых различных областях микроэлектроники, с помощью которых можно создать как любое из перечисленных выше устройств, так и принципиально новое устройство.

Цель курсового проекта – разработка микропроцессорной системы автомобильные часы-термометр-вольтметр на базе микроконтроллера.

1. Описание объекта и функциональная спецификация

Данное устройство предназначено для использования в автомобиле.

Основой устройства является микроконтроллер AT89C2051 фирмы «Atmel». Для отображения информации используется жидкокристаллический индикатор типа ЖКИ13-8/7-02. Несмотря на то, что в настоящее время доступны ЖКИ с встроенными контроллерами, иногда оказывается целесообразным применение специального ЖКИ. Причин может быть несколько. Распространенные ЖКИ со встроенными контроллерами обладают целым рядом недостатков: отсутствие десятичных точек, плохой угол обзора, недостаточный в некоторых случаях размер символов. В то же время существует доступная и довольно удобная в использовании микросхема драйвера ЖКИ КР1820ВГ1. Она выпускается Минским ПО «Интеграл».

Рассмотренное в этой работе устройство устанавливается в автомобиле для индикации времени, контроля заряда аккумулятора и регистрации температуры. Диапазон контролируемого напряжения можно выбрать любой, однако в программе он установлен в пределах от 12,0 В до 15,0 В, а при отклонении от этих значений напряжения включается зуммер.

Функциональная спецификация

1. Входы
 - a. 4 датчика температуры
 - b. Кнопка запуска (включение питания)
 - c. Панель управления с сенсорным переключателем и ИФ приемником
2. Выходы
 - a. Жидкокристаллический индикатор
 - b. Звуковой динамик
3. Функции
 - a. индикация текущего времени

- b. будильник
- c. таймер
- d. индикация температуры в четырех точках
- e. звуковая сигнализация при повышении температуры
- f. индикация напряжения в бортовой сети автомобиля
- g. звуковая сигнализация при падении напряжения бортовой сети
- h. управление режимами работы устройства с помощью ИК-пульта

2. Описание структуры системы

После определения входов и выходов устройства разработана структурная схема устройства. Структурная схема автомобильных вольтметра-термометра-часов приведена на рис. 1.

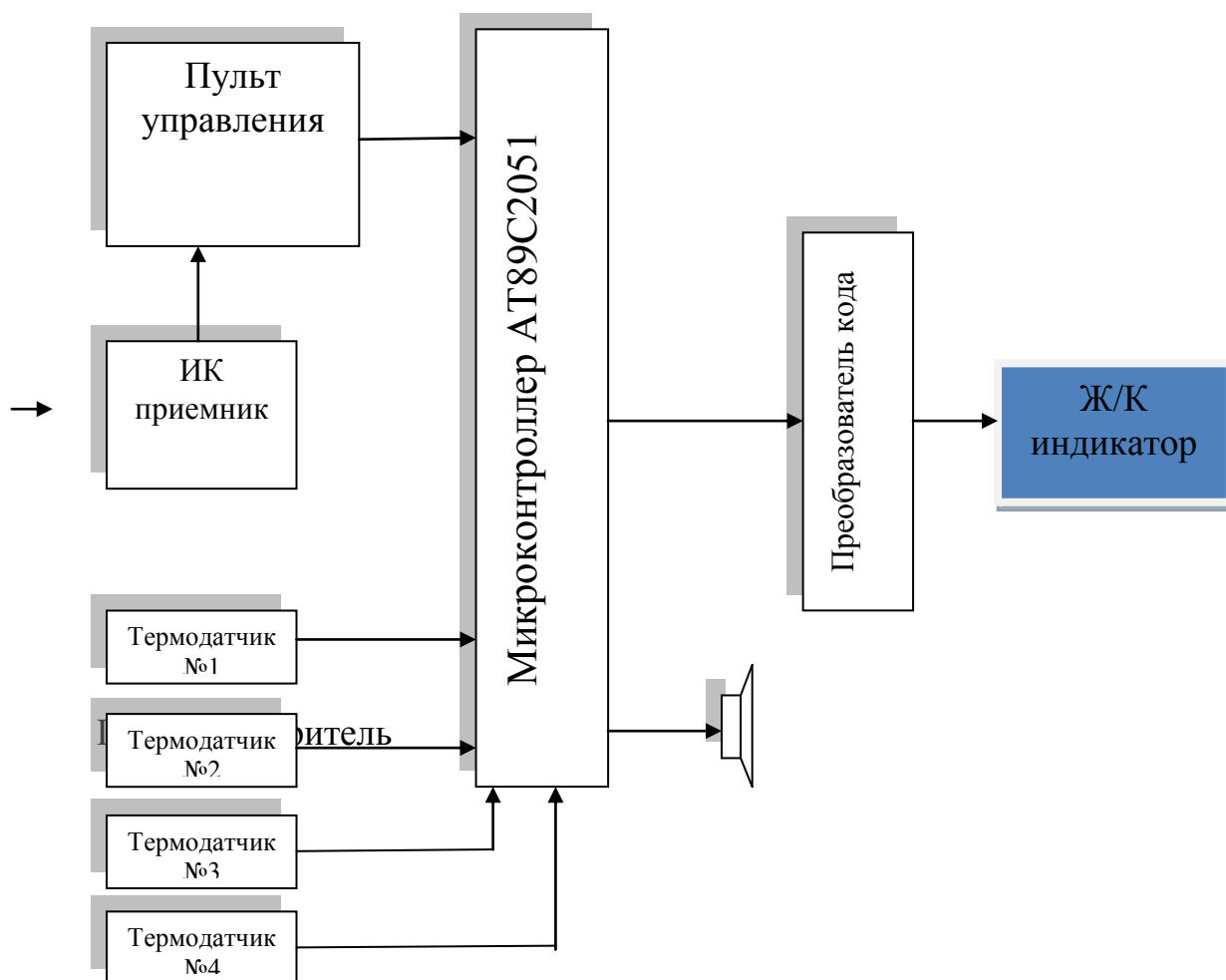


Рис. 1. Структурная схема автомобильных часов-термометра-вольтметра

3. Описание ресурсов МК AT89C2051

AT89C2051 разработан по технологии КМОП. Микроконтроллер оснащенный Flash программируемым и стираемым ПЗУ, а также совместим по системе команд и по выводам со стандартными приборами семейства MCS-51. Объем Flash ПЗУ - 2 Кбайта, ОЗУ - 128 байтов. Имеет 15 линий ввода/вывода, один 16-разрядный таймера/счетчика событий, полнодуплексный порт (UART) пять векторных двухуровневых прерываний, встроенный прецизионный аналоговый компаратор, встроенные генератор и схему формирования тактовой последовательности. Напряжение программирования Flash памяти - 12 В и ее содержимое может быть

защищено от несанкционированных записи/считывания. Имеется возможность очистки Flash памяти за одну операцию и возможность считывания встроенного кода идентификации. Ток потребления в активном режиме на частоте 12 МГц не превышает 15 мА при 6 В и 5,5 мА при напряжении питания 3 В. В пассивном режиме (ЦПУ остановлено, но система прерываний, ОЗУ, таймер/счетчик событий и последовательный порт остаются активными) потребление не превышает 5 мА и 1 мА. В стоповом режиме ток потребления не превышает 100 мкА и 20 мкА при напряжении питания 6 В и 3 В, соответственно. Микроконтроллер AT89C2051 ориентирован на использование в качестве встроенного управляющего контроллера.

Для питания устройства используется интегральный стабилизатор U5 типа 7805. Потребляемый устройством ток очень небольшой, поэтому радиатор для этой микросхемы не нужен.

Поскольку микросхемы контроллера ЖКИ требуют небольшого количества сигналов для связи с микроконтроллером, индикатор можно выполнить конструктивно в отдельном корпусе минимального размера и расположить его в удобном для обозрения месте. Провода датчиков температуры могут иметь длину несколько метров. При этом обязательно должен присутствовать земляной провод. Использовать в качестве земли кузов автомобиля нежелательно. Удобно для термометров использовать аудио кабель, который имеет два провода в общем экране, изолированном снаружи.

AT89C2051 - 8-разрядный микроконтроллер с Flash ПЗУ

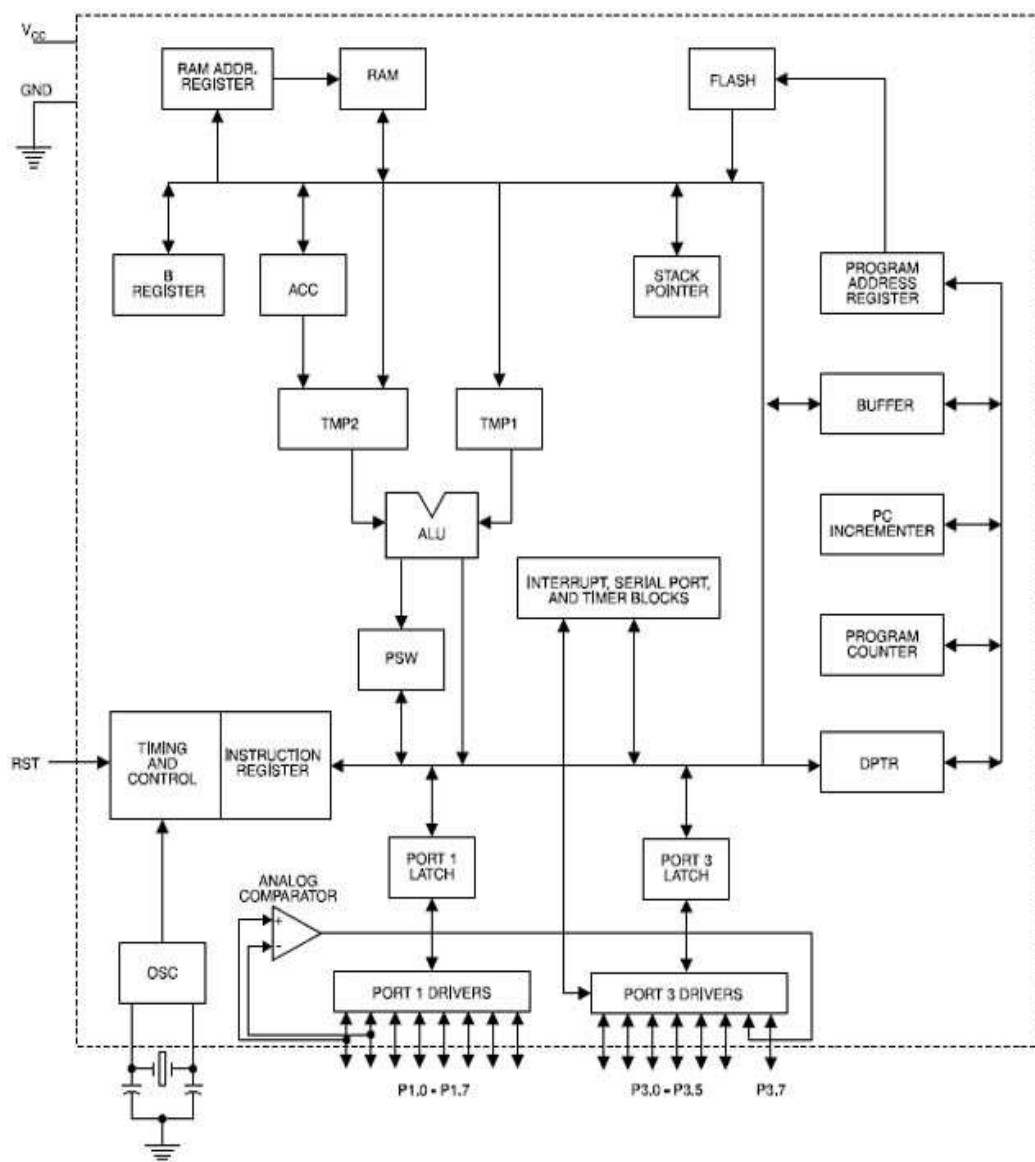


Рис.2. Структурная схема AT89C2051

Микроконтроллер AT89C51 построен по процессорной архитектуре MCS-51, т.е. он умеет выполнять ассемблерные команды описанные этим стандартом. Стандарт был разработан фирмой INTEL и в дальнейшем стал основой для создания современных INTEL процессоров, но проблема создания маленьких устройств (микроконтроллерных систем) осталась актуальной и по сей день. В результате первые миниатюрные процессоры эксплуатируется до сих пор (например, в телефонах АОН).

Цифры 31 или 51 в названии процессора (контроллера) указывают на принадлежность к системе команд MCS-51 (31 в отличии от 51, не имеет возможности использовать порт P0 и P2 как порты - на 31 кристалле это

только адресные линии и линии данных внешних устройств [ПЗУ,ОЗУ,Регистров...] = 51 же кристалл имеет возможность незадействованные выводы адресов использовать как выводы портов ввода - вывода). Цифра 80 в начале указывает на то, что исполняемая программа может быть размещена только во внешней ПЗУ.

Цифра 83,87 или 89 указывает, что программа может быть как во внешней ПЗУ, так и в ПЗУ кристалла (это более поздние модели 1990-е годы, уже научились ПЗУ делать на одной подложке вместе с самим процессором), 83 - масочная ПЗУ (программируется на заводе изготовителе - например контроллер клавиатуры АТ-ХТ), 87 - однократно программируемая ПЗУ на кристалле процессора в корпусе из пластика или многократно (до 100 раз) перепрограммируемая ПЗУ на кристалле в керамическом корпусе и окошком для УФ стирания.

89 - многократно (до 10000 раз) перепрограммируемая ПЗУ на кристалле, электрически стираемая.

АТ- название фирмы изготовителя ATMEL <http://www.atmel.com/> или <http://www.atmel.ru/> (русскоязычный сайт ATMEL).

Кроме того это может быть DS - Dallas, N- Intel, P-Philips...

Так что данная микросхема - это микропроцессор (правильнее сказать микроконтроллер) со встроенной ПЗУ, которую (ПЗУ внутри процессора) и надо запрограммировать, чтобы микросхема начала выполнять требуемые функции.

Данный микроконтроллер программируется стандартным программатором, поддерживающим программирование этого типа микроконтроллеров (например, программатор UNIPRO).



Рис.3. Общий вид выводов AT89C2051

4. Ассемблирование

Для ассемблирования используется макроассемблер MPASM, он содержит все необходимые нам возможности. MPASM входит в пакет программ Microchip MPLAB фирмы Microchip Technology.

В результате работы ассемблера создаются файлы со следующими расширениями:

- * HEX - объектный файл
- * LST - файл листинга
- * ERR - файл ошибок и предупреждений
- * COD

Объектный файл создается в 16-ричном формате и содержит код, который должен быть записан в микросхему. Файл листинга содержит полный листинг программы вместе с загрузочным кодом. В файл ошибок и предупреждений записываются все ошибки и предупреждения, возникающие в процессе ассемблирования. Они также присутствуют и в файле листинга.

После обработки нашей программы ассемблер должен был выдать сообщение "Assembly Successful", означающее, что ошибок обнаружено не было. Файл ошибок не должен был создаваться.

Листинг программы и объектный файл приведен в Приложении А.

5. Разработка алгоритма работы устройства

Алгоритм работы программы показан на рис. 4. После запуска и инициализации микроконтроллера программа переходит к распределителю, в котором каждую секунду последовательно измеряется напряжение, проверяются кнопки, и выполняется вывод на индикацию. Прерывание от этой последовательности происходит каждую секунду для подсчета времени в часах и таймере, если он включен.

После установки флага "Одна секунда" проверяется бортовое напряжение. Если присутствует его отклонение от установленного, то включается звуковой сигнал. Если отклонения нет, то измеренные значения перекодируются для индикации в двоично-десятичный код.

Далее программа переходит к проверке кнопок. Поскольку кнопки — многофункциональные, то и их проверка несколько усложнена. Вначале проверяется флаг индикации часов. Если индикация часов отсутствует, то кнопка установки курсора "Разряд" не проверяется, а сразу проверяется кнопка "Режим". Если индикация часов включена и нажата кнопка "Разряд", то к регистру курсора прибавляется единица.

Если кнопка "Режим" нажата, то на единицу увеличивается регистр режима. По значению регистра режима из таблицы выбирается режим индикации (рис. 5).

При индикации напряжения ранее перекодируемые значения напряжения переписываются в регистры индикации.

При индикации часов проверяется, был ли ранее введен курсор в поле индикатора. Если значение регистра курсора — ненулевое, то выполняется установка часов. Если при этом нажата кнопка "Установка", то к выбранному разряду прибавляется единица, а регистры индикации заполняются новыми значениями. Если установка отсутствует, то регистры индикации заполняются значениями текущего времени.

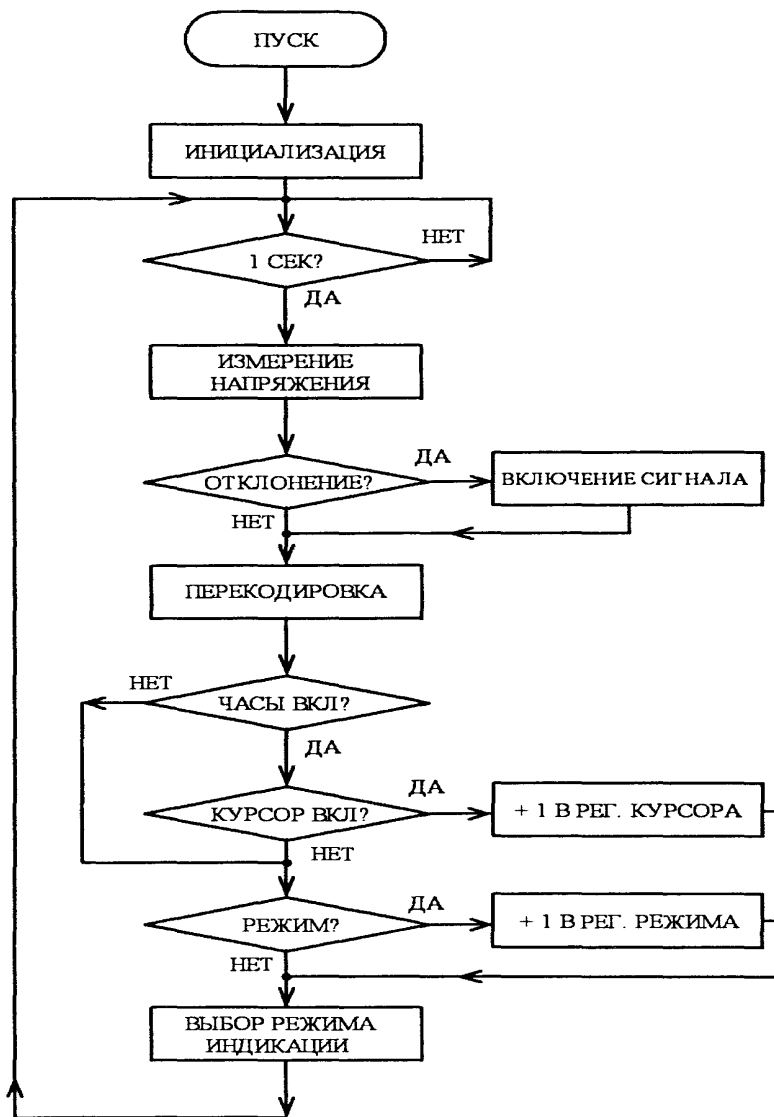


Рис. 4. Алгоритм работы автомобильных часов (начало)

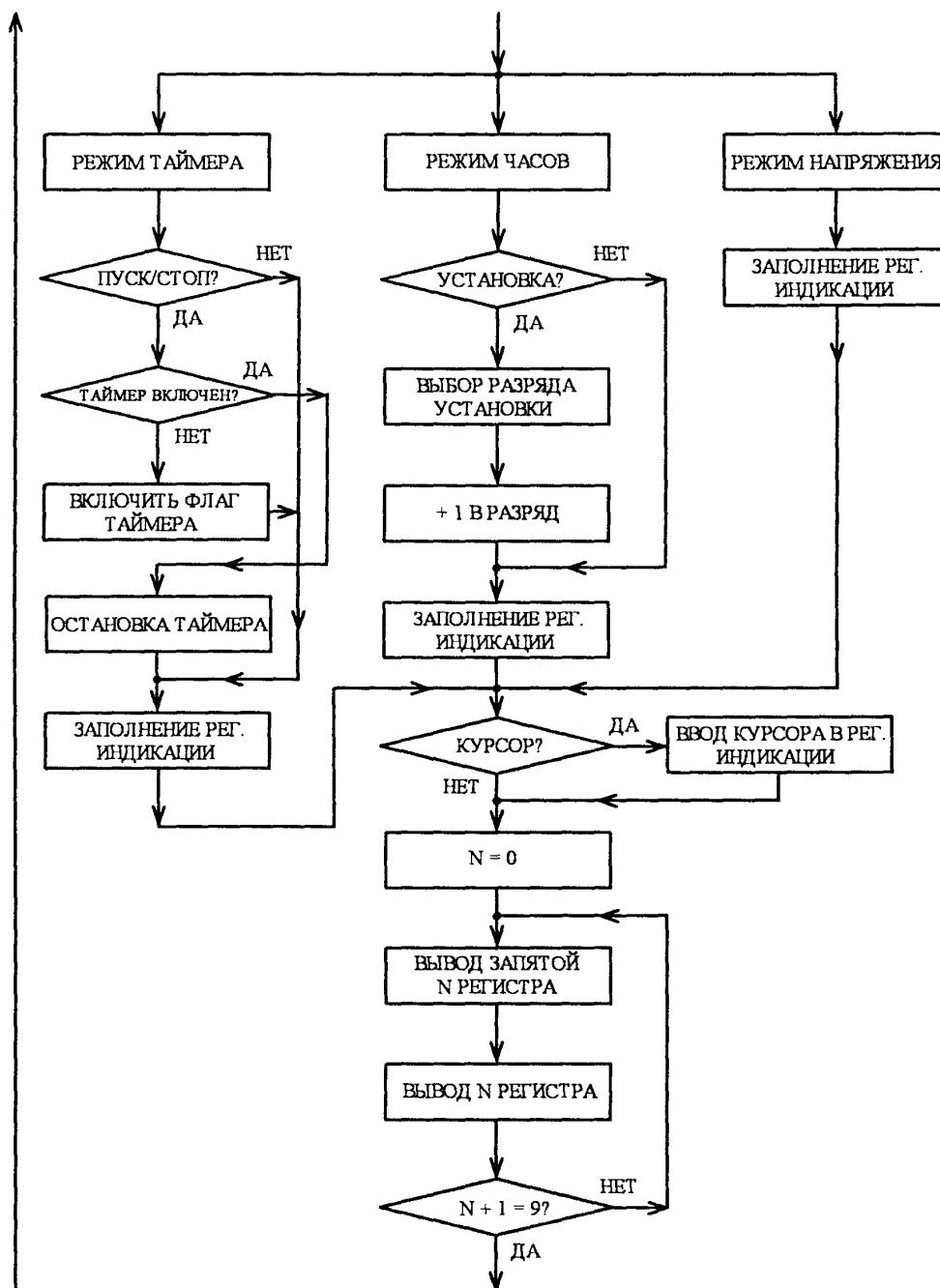


Рис. 5. Алгоритм работы автомобильных часов (продолжение)

Однако заполненные регистры индикации еще не готовы к выводу на индикацию — в них необходимо записать значение курсора. Если значение курсора — ненулевое (т.е. он находится в поле индикатора), то он вводится в младший разряд регистра индикации соответствующего знакоместа.

Если курсор в поле индикатора отсутствует, то обнуляется счетчик цикла записи, и первым импульсом выводится значение запятой для N-го разряда. В принципе, запятая в данном устройстве необходима только одна:

для выделения десятых долей напряжения, — однако подпрограмма вывода на индикацию универсальна, и потому нет смысла ее изменять. Значения запятых заранее записываются в позиционном коде в регистр запятой (т.е. если необходимо высветить запятую в пятом разряде индикатора, то записывают единицу в пятый разряд регистра). При этом необходимо помнить, что первыми в импульсной последовательности идут значения крайнего справа разряда.

После вывода запятой последовательно выводится значение N-ro регистра, начиная с младшего разряда. Затем прибавляется единица к счетчику циклов и, если его значение не равно девяти, цикл вывода данных на индикатор повторяется со следующим регистром. После вывода значения последнего регистра программа возвращается к ожиданию установки флага "Одна секунда" во время прерывания.

Прерывание организовано обычным образом: по переполнению таймера TMR0. При частоте кварцевого резонатора 32 768 Гц коэффициент деления предделителя составляет 32, что вместе с коэффициентом деления таймера, равным 256, и циклом, равным 4, дает одну секунду ($4 \times 32 \times 256 = 32\,768$).

6. Описание выбора элементной базы и работы принципиальной схемы

Микросхема КР1820ВГ1 [1] используется для управления 36-сегментным ЖКИ в режиме 3-уровневого мультиплексирования. Микросхема изготавливается по КМОП-технологии и выпускается в 20-выводном пластмассовом DIP-корпусе. Микросхема содержит встроенный тактовый генератор, резистивный делитель напряжения и делители частоты, с помощью которых формируются сигналы управления строками (общими электродами) и столбцами (сегментными электродами) ЖКИ в режиме 3-уровневого мультиплексирования. Одна микросхема имеет три выхода

управления строками и 12 выходов управления столбцами. Предусмотрена возможность каскадирования схем, что позволяет использовать их для управления мультиплексным ЖКИ с числом сегментов более 36. Микросхема не требует никаких навесных компонентов и работает в диапазоне напряжения питания от 3 до 6 вольт. Назначение выводов микросхемы КР1820ВГ1 показано в таблице 1.

Таблица 1. Назначение выводов микросхемы КР1820ВГ1.

Вывод	Обозначение	Тип	Назначение
1...3	COB1, СОС3, COB3	Выход	Управление столбцами В1, С3, В3
4	CS	Вход	Выбор кристалла
5	Ucc	-	Напряжение источника питания
6	GND	-	Общий
7	D	Вход	Данные
8...13	COA2, COB4, COB2, COA1, СОС2, СОС4	Выход	Управление столбцами А2, В4, В2, А1, С2, С4
14	C	Вход	Тактовый сигнал С
15	COA/G	Выход	Управление сторокой А (вход генератора G)
16	СОС/G	Выход	Управление сторокой С (выход генератора G)
17	COB	Выход	Управление строкой В
18...20	СОС1, COA3, COA4	Выход	Управление столбцами В1, А3, А4

Микросхема КР1820ВГ1 имеет четыре режима работы: одиночный, старший, младший и тестовый. В одиночном режиме одна микросхема управляет 36-сегментным ЖКИ, обеспечивая полную синхронизацию его работы. Старший и младний режимы предназначены для организации управления ЖКИ с числом сегментов более 36, тестовый режим – для контроля качества микросхем в процессе изготовления. Данные вводятся в микросхему в последовательном коде по входу D с синхронизацией записи фронтом тактовых импульсов по входу С (рис. 6).

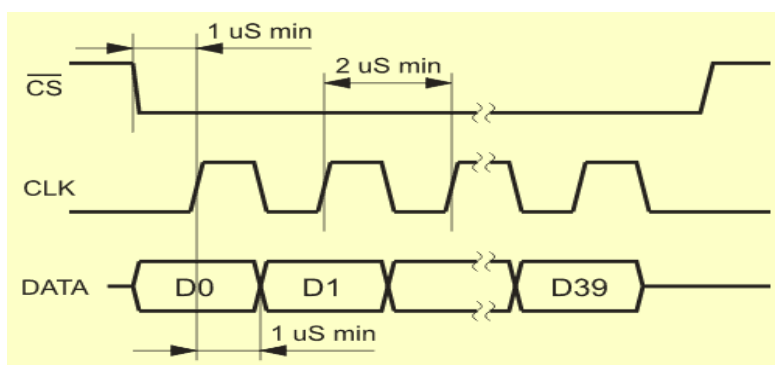


Рис. 6. Загрузка микросхемы KP1820BG1 по последовательной шине

Код записываемых данных определяется конкретной схемой подключения шин управления строками и столбцами к сегментам ЖКИ, а также конфигурацией ЖКИ.

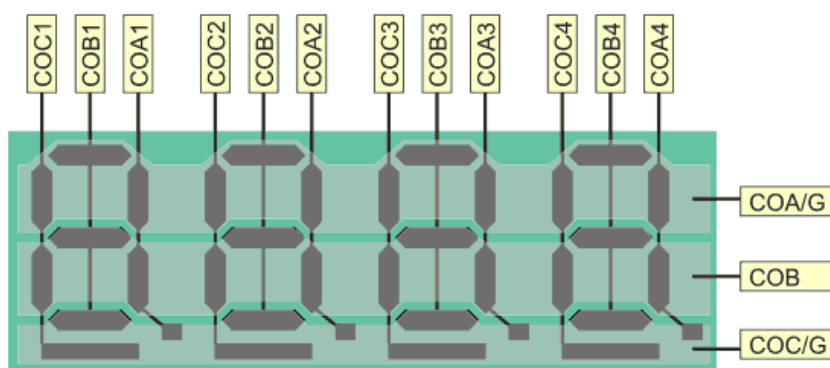


Рис.7. Конфигурация сегментов ЖКИ.

На рис. 7 показан пример конфигурации ЖКИ, а в таблице 2 показан порядок следования битов в кодовой послыке для этого варианта подключения такого ЖКИ.

Таблица 2. Порядок следования битов в кодовой послыке

Бит	Вывод	Сегмент ЖКИ		Бит	Вывод	Сегмент ЖКИ
D0	COA1, COC/G	H1		D20	COB3, COC/G	D3
D1	COB1, COB	G1		D21	COA3, COB	C3
D2	COC1, COA/G	F1		D22	COA3, COA/G	B3
D3	COC1, COB	E1		D23	COB3, COA/G	A3
D4	COB1, COC/G	D1		D24	COA4, COC/G	H4

D5	COA1,COB	C1		D25	COB4, COB	G4
D6	COA1, COA/G	B1		D26	COC4,COA/G	F4
D7	COB1, COA/G	A1		D27	COC4, COB	E4
D8	COA2, COC/G	H2		D28	COB4, COC/G	D4
D9	COB2, COB	G2		D29	COA4,COB	C4
D10	COC2,COA/G	F2		D30	COA4, COA/G	B4
D11	COC2, COB	E2		D31	COB4, COA/G	A4
D12	COB2, COC/G	D2		D32	COC1, COC/G	P1
D13	COA2,COB	C2		D33	COC2, COC/G	P2
D14	COA2, COA/G	B2		D34	COC3, COC/G	P3
D15	COB2, COA/G	A2		D35	COC4, COC/G	P4
D16	COA3, COC/G	H3		D36	Не используется	-
D17	COB3, COB	G3		D37	Q6	-
D18	COC3,COA/G	F3		D38	Q7	-
D19	COC3, COB	E3		D39	Q8	-

Биты D0..D7 соответствуют сегментам первого разряда, биты D8..D15 – второго и т. д. Биты D32..D35 соответствуют специальным сегментам P1...P4. Бит D36 может принимать любое значение. Биты D37 и D38 (Q6 и Q7) управляют режимом работы схемы согласно таблице 3. Бит D39 (Q8) предназначен для синхронизации работы двух и более микросхем при каскадировании.

Для загрузки микросхемы в одиночном режиме необходимо выполнить следующую последовательность действий:

- установить на сходе CS уровень логического 0
- записать восемь битов данных для каждой цифры первого-четвертого разрядов
- записать четыре бита для специальных сегментов и четыре бита управления: 0|0|1|1|P4|P3|P2|P1
- установить на входе CS уровень логической 1

Таблица 3. Назначение битов управления микросхемой

Бит		Режим работы	Выход	
D36 (Q7)	D37 (Q6)		СOC/G	COA/G
1	1	Младший	Выход управления строкой С	Вход генератора
0	1	Одиночный	То же	Выход управления строкой А
1	0	Тестовый	-	-
0	0	Старший	Выход внутреннего генератора	Выход управления строкой А

После установки микросхемы в нужный режим для последующей смены данных необязательно записывать все 40 бит информации. Для загрузки микросхемы в старшем и младшем режимах необходимо выполнить следующую последовательность действий:

- установить на входе CS обеих схем уровень логического 0
- записать 32 бита данных для «младшей» схемы
- записать четыре бита для специальных сегментов младшей схемы и четыре бита управления: 1|1|1|1|P4|P3|P2|P1 (при подаче последней единицы обе микросхемы устанавливаются в младший режим, выходы COA/G обеих схем работают как входы генератора. Происходит синхронизация работы микросхем)

- установить на входах CS обеих схем уровень логической 1
- установить на входе CS «старшей» схемы уровень логического 0
- записать 32 бита данных для старшей схемы
- записать четыре бита для специальных сегментов старшей схемы и четыре бита управления: 0|0|0|0||P4|P3|P2|P1 (после этого вывод COA/G старшей схемы начинает работать как выход управления строкой А, а вывод СОС/G – как выход встроенного генератора. Импульсы с выхода генератора старшей схемы поступают на вход генератора COA/G младшей схемы, и оба кристалла начинают работать синхронно от генератора старшей схемы)

- установить на входе CS установить уровень логической 1

Чтобы записать во внутренние регистры-защелки новые данные, нет необходимости сбрасывать обе схемы: достаточно записать данные по очереди во внутренние регистры-защелки каждой схемы. При этом в последний бит D39 должен записываться ноль как для старшей, так и для младшей схем.

Нужно сказать, что некоторые типы ЖК индикаторов неудовлетворительно работают при питании микросхем драйверов напряжением 5В. Однако картина резко улучшается при снижении напряжения питания до 3.3 – 4.0В. Это сделать совсем несложно, так как потребляемый драйверами ток очень мал. В цепь питания можно включить параметрический стабилизатор напряжения на основе TL431 или даже простой резистивный делитель. На всех цифровых входах драйверов также понадобятся делители напряжения.

В качестве часов реального времени применена микросхема DS1302 фирмы Dallas. Эта микросхема имеет отдельные входы для подключения основного и резервного источников питания, что избавляет от проектирования довольно хитрых схем перехода на резервный источник. Кроме того, имеется встроенная схема «капельной» зарядки резервного источника питания, которая может быть включена программно. Дополнительно микросхема имеет ОЗУ объемом 31 байт, которое может быть использовано для энергонезависимого хранения параметров. Из навесных элементов требуется только кварцевый резонатор. Здесь хочется предостеречь от применения дешевых некачественных резонаторов. Согласно рекомендациям фирмы Dallas, требуется резонатор, рассчитанный на емкость нагрузки 6 пФ. В противном случае точность хода часов будет неудовлетворительной или даже появятся проблемы с запуском кварцевого генератора.

Для обмена с микросхемой DS1302 используются общие с драйверами ЖКИ линии данных и тактирования. Разделены только сигналы CS и RST. К сожалению, микросхема DS1302 имеет довольно специфический 3-х

проводный интерфейс, который в фирменной документации описан весьма неоднозначно. Это довольно редкий пример плохого фирменного описания. Поэтому в новых разработках лучше применять более современные микросхемы, например DS1307 с интерфейсом I2C.

В качестве датчиков температуры применены микросхемы цифровых термометров DS1821 фирмы Dallas. В цепях данных термометров включены защитные цепочки R11-R14, VD1-VD8, а в цепи питания включен ограничивающий резистор R10 для защиты от короткого замыкания. Несмотря на то, что аппаратно имеется возможность подключить четыре термометра, данная версия программы работает только с тремя. Это вызвано недостаточным объемом памяти программ. Термометры устанавливаются в разных местах автомобиля. В данном случае они были установлены в салоне, на открытом воздухе и в моторном отсеке. Благодаря наличию заданных программно порогов, кроме индикации температуры осуществляется ещё и контроль ее выхода за безопасные пределы. Ввиду недостаточного объема памяти программ, редактирование порогов температур не поддерживается. Пороги в виде констант внесены в текст программы. Для первого термометра +55 градусов, а для второго и третьего термометра +99 градусов.

Для измерения напряжения бортовой сети построен простейший 8-разрядный АЦП на основе встроенного в микроконтроллер компаратора. Для уменьшения влияния помех используется 16-кратное усреднение результатов. Принцип работы АЦП пояснен на рис. 8.

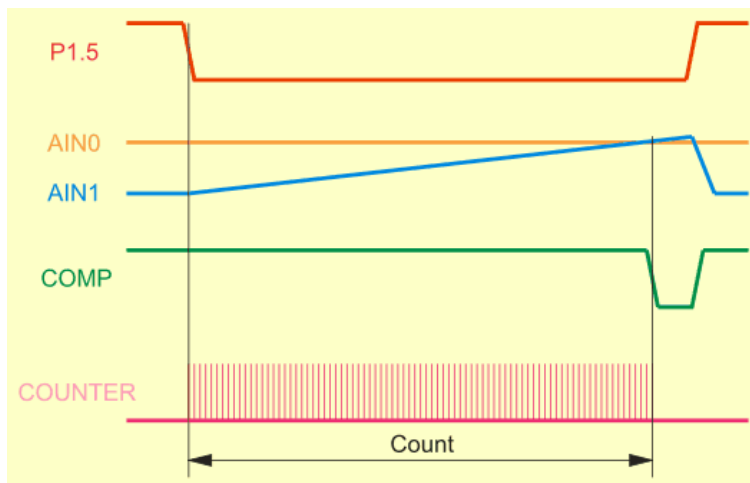


Рисунок 8. Принцип работы АЦП

На входе AIN1 формируется пилообразное напряжение, которое сравнивается с входным напряжением, которое через делитель R2, R3 поступает на вход компаратора AIN0. Емкость C8 снижает влияние помех на показания вольтметра. Пилообразное напряжение формируется на емкости C9 в результате заряда ее стабильным током от генератора тока, собранного на элементах VT2, VD9, R6. Перед началом измерения конденсатор C9 разряжен с помощью открытого ключа VT3. Когда начинается цикл измерения, на порту P1.5 устанавливается низкий логический уровень, транзистор VT3 закрывается, и напряжение на конденсаторе C9 начинает линейно нарастать. В это время разрешается счет программному счетчику. Счет идет до тех пор, пока напряжение на C9 не станет равным входному (на средней точке делителя R2, R3). При этом переключается встроенный компаратор, и счет запрещается. Значение, накопленное в счетчике, будет пропорционально входному напряжению. Применение генератора тока (а не резистора) позволило получить линейный закон заряда C9, что исключило необходимость программной линеаризации АЦП, которая потребовала бы дополнительных затрат и так дефицитной памяти программ. Необходимо отметить, что конденсатор C9 должен быть термостабильным, например, с пленочным диэлектриком типа К73-17. С помощью резистора R6 подбирают ток генератора таким образом, чтобы показания АЦП совпадали с реальным

значением напряжения на входе +В. Кроме индикации напряжения осуществляется контроль его падения ниже порога 10В. В случае такого падения включается звуковая сигнализация.

Для управления устройством применяется ИК-пульт дистанционного управления. Конструктивно он выполнен на базе дешевого малогабаритного калькулятора. Используются только его корпус и клавиатура. В пульте применена микросхема INA3010D в корпусе SOIC. Для питания используются два элемента СЦ-30. Используемый номер системы кода RC-5 - 1ЕН. Схема пульта не приводится, так как практически повторяет типовую схему включения микросхемы INA3010 (SAA3010) и зависит от конфигурации конкретной клавиатуры. Коды, соответствующие кнопкам, также могут отличаться от заданных. Для восстановления соответствия необходимо правильно заполнить перекодировочную таблицу в программе. Сделать это можно даже не перетранслируя программы с помощью шестнадцатиричного редактора прямо в .bin – файле. Таблица расположена по адресам 7В8Н - 7Е3Н . Соответствие функций управления, их внутренних кодов (после перекодировки) и кодов ИК ДУ (до перекодировки) приведено в таблице 4.

Таблица 4. Коды кнопок управления

Номер команды	Название команды	Внутренний код команды (после перекодировки)	Код ИК ДУ (до перекодировки)
1	TIMER	0СН	00Н
2	CLOCK	0DН	01Н
3	ALARM	0ЕН	02Н
4	LOCK	0FН	03Н
5	7	08Н	08Н
6	8	09Н	09Н
7	9	0АН	0АН
8	LIST	10Н	0ВН
9	4	05Н	10Н
10	5	06Н	11Н

11	6	07H	12H
12	ESCAPE	11H	13H
13	ALARM DISABLE	14H	18H
14	TIMER CLEAR	13H	1AH
15	0	01H	20H
16	BACKSPACE	12H	22H
17	1	02H	28H
18	2	03H	29H
19	3	04H	2AH
20	ENTER	0BH	2BH

Вот краткое описание команд управления:

- CLOCK – вход в режим установки текущего времени
- ALARM – вход в режим установки времени будильника
- ALARM DISABLE – выключение будильника
- TIMER – включение индикации значения таймера
- TIMER CLEAR – очистка таймера
- LIST – включение циклической смены параметров
- LOCK – запрещение смены параметров
- 0..9 – кнопки для ввода числовых значений параметров
- ENTER – ввод отредактированного параметра
- ESCAPE – отказ от редактирования параметра
- BACKSPACE – возврат на один символ при редактировании

В качестве ИК приемника использована интегральная микросхема SFH-506 фирмы Siemens. Эта микросхема весьма чувствительна к помехам по цепи питания, поэтому применен RC фильтр R15 C7.

В случае срабатывания будильника, превышения температурой установленного порога или понижения напряжения в бортовой сети формируется звуковой сигнал. Для его формирования использована малогабаритная динамическая головка HA1, которая подключена через

транзисторный ключ VT1. Звуковые сигналы также формируются при нажатиях на кнопки управления.

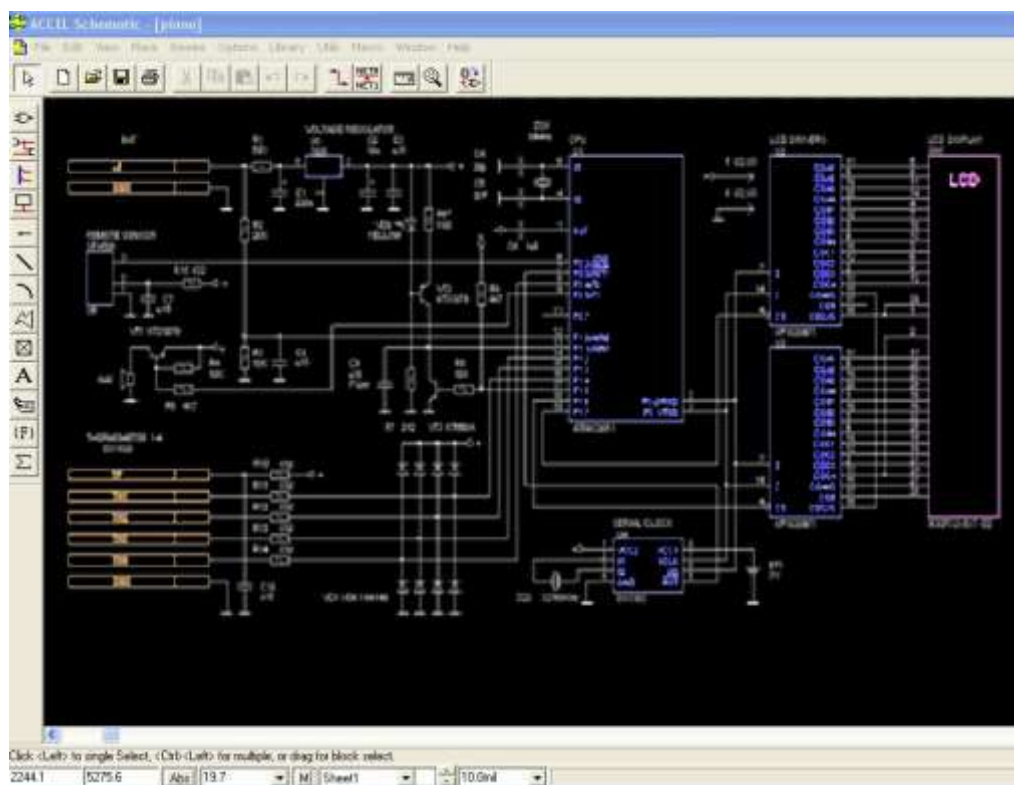


Рис. 9. Принципиальная схема в Accel EDA.

ЗАКЛЮЧЕНИЕ

В данном курсовом проекте разработано устройство - электронные часы-вольтметр-термометр. Разработана электрическая принципиальная этого устройства и программа для микроконтроллера. В результате сборки получена прошивка программы для памяти микроконтроллера. Применение микроконтроллера позволило упростить принципиальную схему и расширить функциональные возможности микроконтроллера, так как для изменения функций устройства достаточно внести изменения в программу микроконтроллера.

СПИСОК ЛИТЕРАТУРЫ

1. Белов А.В. Микроконтроллеры AVR в радиолюбительской практике – СПб, Наука и техника, 2007 – 352с.
2. Проектирование цифровых устройств на однокристальных микроконтроллерах / В.В. Сташин [и др.]. – М.: Энергоатомиздат, 1990. – 224 с.
3. Евстифеев А.В. Микроконтроллеры Microchip: практическое руководство/А.В.Евстифеев. – М.: Горячая линия – Телеком, 2002. – 296 с.
4. Кравченко А.В. 10 практических устройств на AVR-микроконтроллерах. Книга 1 – М., Додэка –XXI, МК-Пресс, 2008 – 224с.
5. Трамперт В. Измерение, управление и регулирование с помощью AVR-микроконтроллеров: Пер. с нем – К., МК-Пресс, 2006 – 208с.
6. Мортон Дж. Микроконтроллеры AVR. Вводный курс /Пер. с англ. – М., Додэка –XXI, 2006 – 272с.
7. Техническая документация на микроконтроллеры AT89C2051 фирмы «Atmel». ООО «Микро -Чип», Москва, 2002.-184 с.

ПРИЛОЖЕНИЕ А

Листинг программы и объектный файл

; ЧАСЫ-ТЕРМОМЕТР-ВОЛЬТМЕТР ДЛЯ АВТОМОБИЛЯ.

РАЗРАБОТАЛ ДЕРКАЧ

; ПРОГРАММА = АВТО.ASM

; ВЕРСИЯ: 20-01-07.

; АССЕМБЛЕР И ОТЛАДЧИК: MPLAB IDE, ВЕРСИЯ: 5.70.40.

LIST P=16F676

#INCLUDE P16F676.INC

__CONFIG 31D0H

;=====

; ИСПОЛЬЗУЕТСЯ КВАРЦ ЧАСТОТОЙ 32768 ГЦ.

; КОЭФФИЦИЕНТ ДЕЛЕНИЯ ПРЕДЕЛИТЕЛЯ РАВЕН 32, ЧТО
ВМЕСТЕ

; С TMR0 (256) И ЦИКЛОМ, РАВНЫМ 4 ТАКТАМ

; ДАЕТ НА ВЫХОДЕ 1 СЕКУНДУ ($4 \times 32 \times 256 = 32768$).

;=====

; RA1 - РЕЖИМ - УСТАНОВКА, RA2 - РАЗРЯД - ПУСК,

; RA3 - ВЫХОД ИЗЛУЧАТЕЛЯ,

; RC5 - LOAD, RC3 - DIN,

; RC4 - DCLK

;=====

CBLOCK H'20'

;=====

; ОПРЕДЕЛЕНИЕ РЕГИСТРОВ ВРЕМЕНИ.

;=====

HOU ;ЧАСЫ ДВОИЧНЫЕ.

CL ;ЕДИНИЦЫ СЕКУНД ЧАСОВ.

CH	;ДЕСЯТКИ СЕКУНД.
ML	;ЕДИНИЦЫ МИНУТ.
MH	;ДЕСЯТКИ МИНУТ.
HL	;ЕДИНИЦЫ ЧАСОВ.
HH	;ДЕСЯТКИ ЧАСОВ.
TCL	;ДЛЯ ТАЙМЕРА.
TCH	;
TML	;
TMH	;
THL	;
TNH	;

=====

; РЕГИСТРЫ ДЛЯ ОРГАНИЗАЦИИ ИНДИКАЦИИ.

=====

ZPT	;РЕГИСТР ЗАПЯТОЙ.
TZPT	;ЗАПЯТАЯ ДЛЯ ВЫВОДА НА ИНДИКАЦИЮ.
COUZ	;СЧЕТЧИК ВЫВОДА ЗАПЯТЫХ.
COU	;СЧЕТЧИК ВЫВОДА БИТ.
KYPC	;ВЫБОР РАЗРЯДА УСТАНОВКИ.
KYPCI	;ИНДИКАЦИИ.
REID	;РЕЖИМА ИНДИКАЦИИ.
TEMP	;ВРЕМЕННЫЙ.
EDI	;ДЕСЯТЫЕ НАПРЯЖЕНИЯ.
DEI	;ЕДИНИЦЫ ВОЛЬТ.
COI	;ДЕСЯТКИ ВОЛЬТ.

=====

; ВРЕМЕННЫЕ.

=====

WTEMP ;БАЙТ СОХРАНЕНИЯ РЕГИСТРА W ПРИ
ПРЕРЫВАНИИ.

STEMP ;БАЙТ СОХРАНЕНИЯ РЕГИСТРА STATUS ПРИ
ПРЕРЫВАНИИ.

FTEMP ;ВРЕМЕННЫЙ ДЛЯ FSR.

TEKH ;

EDA ;

DEA ;

YCTL ;

YCTLI ;

=====

; ОПРЕДЕЛЕНИЕ БИТОВ СОСТОЯНИЯ ФЛАГОВ.

=====

FLAG

;

; 0-> ВКЛЮЧЕНИЕ СИГНАЛА.

; 1-> ТЕКУЩЕЕ ВРЕМЯ.

; 2-> ВКЛЮЧЕН РЕЖИМ ТАЙМЕРА.

; 3-> ИНДИКАЦИЯ НАПРЯЖЕНИЯ.

; 4-> НЕТ КУРСОРА.

; 5-> ПУСК-СТОП.

; 6-> ПРОЧЕРК В ЧАСАХ.

; 7-> УСТАНОВКИ.

=====

FLAG1

;

; 1-> 1 СЕК ЦИКЛА.

; 2-> ПЕРЕДАНА ЗАПЯТАЯ.

; 4-> ЗУММЕР ВКЛЮЧЕН.

; 5-> СТОРОЖОК НАПОМИНИНИЯ АВАРИИ ЧЕРЕЗ 1 ЧАС.

=====

ENDC

```

;=====
; ОПРЕДЕЛЕНИЕ РЕГИСТРОВ ИНДИКАЦИИ.
;=====

R1 EQU 50H ;МЛАДШИЙ РАЗРЯД.
R2 EQU 51H ;
R3 EQU 52H ;
R4 EQU 53H ;
R5 EQU 54H ;
R6 EQU 55H ;
R7 EQU 56H ;
R8 EQU 57H ;СТАРШИЙ РАЗРЯД.
TEKL EQU 58H ;

;=====
; ОПРЕДЕЛЕНИЕ БИТ ПОРТОВ ВВОДА/ВЫВОДА.
;=====

YC EQU 1 ;РЕЖИМ/УСТАНОВКА.
KY EQU 2 ;ПУСК/КУРСОР.
LOAD EQU 5 ;ЗАГРУЗКА.
DIN EQU 3 ;ДАННЫЕ.
DCLK EQU 4 ;СИНХРОИМПУЛЬСЫ.

;=====
; 1. ПУСК.
;=====

    ORG 0
    GOTO INIT
    ORG 4
    GOTO PRER

;=====
; 2. ИНИЦИАЛИЗАЦИЯ.
;=====

```

```

INIT
BSF STATUS,5 ;ПЕРЕХОДИМ В БАНК 1.
MOVLW 0FFH ;
MOVWF ADCON1^80H ;ТАКТ АЦП ОТ ВНУТРЕННЕГО
ГЕНЕРАТОРА 500 кГц.
MOVLW B'00000100';K=32.
MOVWF OPTION_REG^80H ;РЕЗИСТОРЫ ВКЛЮЧЕНЫ.
MOVLW B'10100000' ;РАЗРЕШЕНИЕ ПЕРЕРЫВАНИЙ ОТ TMR0.
MOVWF INTCON ;
CLRFPIE1^80H ;ЗАПРЕЩЕНЫ ВСЕ ПЕРИФЕРИЙНЫЕ
ПЕРЕРЫВАНИЯ.
MOVLW B'00001111' ;ВСЕ ВЫХОДЫ. 0 - ВХОД АЦП.
MOVWF TRISA^80H ;
CLRFTRISC^80H ;ВСЕ ВЫХОДЫ.
CLRFVRCON^80H ;ИОН ОТКЛЮЧЕН.
CLRFPCON^80H ;ПЕРЕРЫВАНИЯ ПО ПИТАНИЮ ЗАПРЕЩЕНЫ.
MOVLW B'00000110' ;
MOVWF WPUA^80H ;ПОДТЯГИВАЮЩИЕ РЕЗИСТОРЫ
ВКЛЮЧЕНЫ.
CLRFIOCA^80H ;ПЕРЕРЫВАНИЯ ЗАПРЕЩЕНЫ.
MOVLW .1
MOVWF ANSEL^80H ;ВЫБРАН АНАЛОГОВЫЙ ВХОД RA0/AN0.
BCF STATUS,5 ;ПЕРЕХОДИМ В БАНК 0.
CLRFPORTC ;ВЫХОД И СВЕТОДИОД ВЫКЛЮЧЕНЫ.
CLRFT1CON ;ТАЙМЕР 1 ОТКЛЮЧЕН.
MOVLW .7
MOVWF CMCON ;КОМПАРАТОР ВЫКЛЮЧЕН.
CLRFFLAG ;ВСЕ ОБНУЛЯЕМ И УСТАНАВЛИВАЕМ.
CLRFFLAG1
CLRFEDI

```



```

CLRFDEI
CLRFCOI
CLRFCL
CLRFCH
CLRFML
CLRFMH
CLRFHL
CLRFHH
CLRFTCL
CLRFTCH
CLRFTML
CLRFTMH
CLRFTHL
CLRFTHH
CLRFPEID
CLRFZPT
CLRFCOUZ
CLRFCOU
CLRFHOU
MOVLW  96H
MOVWF  YCTL      ;УСТАНОВКА МАКСИМУМА = 15,0 В.
MOVLW  78H
MOVWF  YCTLI     ;УСТАНОВКА МИНИМУМА = 12,0 В.
CLRFKYPC
BSF  FLAG,4
GOTO  PAC
;=====
; 3. ТАБЛИЦА СЕГМЕНТОВ.
;=====
SEG

```

;D E G F A B C K

```
ADDWF    PCL,1          ;
RETLW    B'11011110';0
RETLW    B'00000110';1
RETLW    B'11101100';2
RETLW    B'10101110';3
RETLW    B'00110110';4
RETLW    B'10111010';5
RETLW    B'11111010';6
RETLW    B'00001110';7
RETLW    B'11111110';8
RETLW    B'10111110';9
RETLW    B'00000000';10-> ПУСТО.
RETLW    B'00100000';11-> ПРОЧЕРК.
RETLW    B'10000000';12-> ПРОЧЕРК.
```

;=====

; 4. ТАБЛИЦА ПЕРЕКОДИРОВКИ КУРСОРА.

;=====

КУРСУ

```
MOVFW    КУРС          ;
ADDWF    PCL,1          ;
RETURN    ;НЕТ КУРСОРА.
RETLW    B'00000100';3
RETLW    B'00001000';4
RETLW    B'00010000';5
RETLW    B'00100000';6
RETLW    B'01000000';7
```

;=====

; 5. ВЫБОР РЕЖИМА ИНДИКАЦИИ.

;=====

VUBOR

```
MOVFW    PEID ;ИЗМЕНЯЕМ РЕЖИМ ИНДИКАЦИИ.  
ADDWF    PCL,1 ;  
GOTO     INDH;ИНДИКАЦИЯ ЧАСОВ.  
GOTO     INDТ;ТАЙМЕРА.  
GOTO     INDU;НАПРЯЖЕНИЯ.
```

```
;=====
```

; 6. ВЫБОР РАЗРЯДА УСТАНОВКИ.

```
;=====
```

УСТ

```
BTFSC    PORTA,УС;ЕСЛИ КНОПКА "РЕЖИМ" НАЖАТА,  
RETURN  
MOVFW    КУРС ;ТО ПО КУРСОРУ  
ADDWF    PCL,1 ;ВЫБИРАЕМ РАЗРЯД УСТАНОВКИ.  
RETURN    ;НЕТ КУРСОРА.  
GOTO     УС0 ;МИНУТЫ.  
GOTO     УС1 ;ДЕСЯТКИ МИНУТ.  
GOTO     УСЕ ;ОБНУЛЕНИЕ.  
GOTO     УС2 ;ЧАСЫ.  
GOTO     УС3 ;ДЕСЯТКИ ЧАСОВ.
```

```
;=====
```

; 7. ТАБЛИЦЫ ПЕРЕВОДА ДЕСЯТКОВ В ДВОИЧНОЕ ЧИСЛО.

```
;=====
```

DEBIN

```
ADDWF    PCL,1 ;  
RETLW    .0  
RETLW    .10  
RETLW    .20  
RETLW    .30  
RETLW    .40
```

RETLW .50

RETLW .60

RETLW .70

RETLW .80

RETLW .90

;=====

; 8. ПРОВЕРКА НАЖАТЫХ КНОПОК УСТАНОВКИ.

;=====

КНОП

BTFSS FLAG,1 ;ЕСЛИ НЕТ РЕЖИМА ЧАСОВ,

GOTO \$+5 ;ТО КУРСОР НЕ МЕНЯЕТСЯ.

BTFSS PORTA,KY;ПРИ НАЖАТОЙ КНОПКЕ

CALL КУРСОР ;ИДЕМ НА УСТАНОВКУ КУРСОРА.

BTFSC FLAG,7 ;ЕСЛИ ИДЕТ УСТАНОВКА,

GOTO VUBOR ;ТО РЕЖИМ НЕ МЕНЯЕТСЯ.

BTFSC PORTA,YS;ЕСЛИ КНОПКА "РЕЖИМ" НАЖАТА,

GOTO VUBOR ;ИЛИ ИДЕМ НА ВЫБОР РЕЖИМА

ИНДИКАЦИИ.

INCF PEID,1 ;ИЗМЕНЕНИЕ РЕЖИМА ИНДИКАЦИИ.

MOVLW .3 ;3 РЕЖИМОВ ИНДИКАЦИИ.

SUBWF PEID,0 ;ЕСЛИ БОЛЬШЕ,

BTFSS STATUS,2;ТО ПОЙДЕМ НА СБРОС.

GOTO VUBOR ;НА ЗАПИСЬ В РЕГИСТРЫ

ИНДИКАЦИИ.

CLRFPEID ;СБРОС РЕЖИМА.

GOTO VUBOR ;НА ВЫБОР РЕЖИМА ИНДИКАЦИИ.

RETURN

;=====

; 9. УСТАНОВКА КУРСОРА (ВЫБОР РАЗРЯДА УСТАНОВКИ).

;=====

KYPCOP

```
BSF FLAG,7 ;УСТАНОВКА.  
BCF FLAG,4 ;СБРОС ФЛАГА НЕТ КУРСОРА.  
INCF KYPC,1 ;ПРИБАВИМ ЕДИНИЦУ В КУРСОР.  
MOVLW .6 ;НЕ БОЛЕЕ 5.  
SUBWF KYPC,0 ;  
SKPC ;ЕСЛИ БОЛЬШЕ ИЛИ РАВНО 6,  
RETURN ;  
CLRFKYPC ;ОБНУЛИМ.  
BSF FLAG,4 ;НЕТ КУРСОРА В ПОЛЕ.  
BCF FLAG,7 ;НЕТ УСТАНОВКИ.  
RETURN ;
```

```
;=====
```

; 10. ВВОД КУРСОРА В МЛ. РАЗРЯД РЕГИСТРОВ.

```
;=====
```

KYPVO

```
BTFSK FLAG,4 ;ЕСЛИ НЕТ КУРСОРА,  
RETURN ;ТО ВЕРНЕМСЯ.  
CALL KYPCY ;УСТАНОВИМ РЕЖИМЫ.  
MOVWF KYPCI ;В КУРСОР ИНДИКАЦИИ.  
RRF KYPCI,1 ;ЗАПОЛНИМ БИТ "С".  
ADDCF R1,1 ;ПЕРЕНЕСЕМ В РЕГИСТР ИНДИКАЦИИ.  
RRF KYPCI,1 ;ЗАПОЛНИМ БИТ "С".  
ADDCF R2,1 ;ПЕРЕНЕСЕМ В РЕГИСТР ИНДИКАЦИИ.  
RRF KYPCI,1 ;ОСТАЛЬНЫЕ РЕГИСТРЫ  
ADDCF R3,1 ;ЗАПОЛНЯЕМ АНАЛОГИЧНО.  
RRF KYPCI,1 ;  
ADDCF R4,1 ;  
RRF KYPCI,1 ;  
ADDCF R5,1 ;
```

```

RRF KYPCI,1 ;
ADDCF R6,1 ;
RRF KYPCI,1 ;
ADDCF R7,1 ;
RRF KYPCI,1 ;
ADDCF R8,1 ;
RETURN ;

```

```

;=====

```

```

; 11. ВЫВОД НА ИНДИКАЦИЮ.

```

```

;=====

```

```

IND

```

```

CALL KYPVO ;ВВЕДЕМ КУРСОРЫ В РЕГИСТРЫ
ИНДИКАЦИИ.

```

```

MOVFW ZPT ;ЗНАЧЕНИЯ ЗАПЯТЫХ ПЕРЕПИШЕМ

```

```

MOVWF TZPT;ВО ВРЕМЕННЫЙ РЕГИСТР.

```

```

BSF FLAG1,2 ;ЗАПЯТАЯ ПЕРЕДАНА.

```

```

BCF PORTC,DIN ;ДАННЫЕ РАВНЫ НУЛЮ.

```

```

BCF PORTC,LOAD ;НАЧАЛО ПЕРЕДАЧИ (LOAD=0).

```

```

RRF TZPT,1 ;ВЫТОЛКНЕМ ОЧЕРЕДНУЮ ЗАПЯТУЮ.

```

```

CALL VUV0 ;

```

```

BCF FLAG1,2 ;ЗАПЯТАЯ ПЕРЕДАНА.

```

```

MOVLW R1 ;ЗАПИШЕМ АДРЕС ПЕРВОГО РЕГИСТРА
ИНДИКАЦИИ.

```

```

MOVWF FSR ;

```

```

MOVFW INDF;ЗНАЧЕНИЕ ПЕРВОГО РЕГИСТРА

```

```

MOVWF TEMP ;ПЕРЕПИШЕМ ВО ВРЕМЕННЫЙ.

```

```

BCF PORTC,LOAD ;НАЧАЛО ПЕРЕДАЧИ (LOAD=0).

```

```

CALL VUVOD ;НА ВЫВОД.

```

```

POVT

```

```

BSF FLAG1,2 ;ЗАПЯТАЯ ПЕРЕДАНА.

```

```

RRF TZPT,1 ;ВЫТОЛКНЕМ ОЧЕРЕДНУЮ ЗАПЯТУЮ.
CALL VUV0 ;
BCF FLAG1,2 ;ЗАПЯТАЯ ПЕРЕДАНА.
INCF FSR,1 ;УВЕЛИЧИМ АДРЕС РЕГИСТРА
ИНДИКАЦИИ.

```

```

MOVFW INDF ;ПЕРЕПИШЕМ ЕГО ЗНАЧЕНИЕ
MOVWF TEMP ;ВО ВРЕМЕННЫЙ.
CALL VUVOD ;НА ВЫВОД.
INCF COUZ,1 ;ПОДСЧИТАЕМ ЧИСЛО
MOVLW .7 ;ПЕРЕДАВАЕМЫХ
SUBWF COUZ,0 ;ЗАПЯТЫХ.
BTFSC STATUS,2;
CLRF COUZ ;ОБНУЛИМ СЧЕТЧИК.
BTFSS STATUS,2;
GOTO POVT ;ПОВТОРИМ ВЫВОД.
BSF PORTC,LOAD ;КОНЕЦ ПЕРЕДАЧИ.
RETURN ;

```

CUNX

```

BSF PORTC,DCLK ;СИНХРОТМПУЛЬС = 1.
CALL PAUS ;ПАУЗА.
BCF PORTC,DCLK ;СИНХРОИМПУЛЬС = 0.
RETURN ;ВОЗВРАТ.

```

PAUS

```

MOVLW .5 ;МОЖНО ПОДБИРАТЬ ЗНАЧЕНИЕ ПАУЗЫ.
ADDLW -1 ;ПАУЗА = ЧИСЛО X 4 МКС.
BTFSS STATUS,2;
GOTO $-2 ;ПОВТОРИМ.
RETURN ;ВЕРНЕМСЯ.

```

VUVOD

```

RRF TEMP,1 ;СДВИНЕМ ВПРАВО.

```

VUV0

```
BTFSS    STATUS,0;ПО НУЛЕВОМУ РАЗРЯДУ
BCF  PORTC,DIN    ;УСТАНОВЛИВАЕМ ДАННЫЕ
BTFSC    STATUS,0;В 0 ИЛИ 1.
BSF  PORTC,DIN    ;
CALL     CUNX     ;СИНХРОНИЗИРУЕМ ДАННЫЕ.
BTFSC    FLAG1,2  ;ЕСЛИ ЗАПЯТАЯ ПЕРЕДАНА,
RETURN   ;ТО ВЕРНЕМСЯ.
INCF COU,1    ;ПОДСЧИТАЕМ ЧИСЛО БИТ.
MOVLW    .8     ;
SUBWF    COU,0   ;
BTFSS    STATUS,2;ЕСЛИ НЕ ВСЕ БИТЫ ПЕРЕДАНЫ,
GOTO     VUVOD   ;ПОВТОРИМ ВЫВОД.
CLRF COU ;ОБНУЛИМ СЧЕТЧИК.
RETURN   ;
```

=====

; 12. СОХРАНЕНИЕ И ВОССТАНОВЛЕНИЕ ЗНАЧЕНИЙ РЕГИСТРОВ ПРИ ПЕРЕРЫВАНИИ.

=====

PRER

```
MOVWF    WTEMP    ;СОХРАНЕНИЕ ЗНАЧЕНИЙ РЕГИСТРОВ W,
MOVFW    STATUS    ;STATUS,
MOVWF    STEMP    ;
MOVFW    FSR    ;FSR.
MOVWF    FTEMP    ;
BSF  FLAG1,1    ;1 СЕК ЦИКЛА.
CALL    S1    ;ПОДСЧИТАЕМ ВРЕМЯ.
BTFSC    FLAG,2    ;
CALL    TAIM    ;
REPER ;ВОССТАНОВЛЕНИЕ СОХРАЕННЫХ ЗНАЧЕНИЙ.
```



```

MOVFW  STEMP    ;ВОССТАНОВЛЕНИЕ РЕГИСТРОВ:
MOVWF  STATUS   ;STATUS,
MOVFW  FTEMP    ;
MOVWF  FSR      ;FSR,
MOVFW  WTEMP    ;W.
BCF INTCON,2;СБРАСЫВАЕМ ФЛАГ ПЕРЕРЫВАНИЯ ОТ
TMR0.

```

```

RETFIE      ;ВОЗВРАТ ИЗ ПЕРЕРЫВАНИЯ.

```

```

;=====

```

```

; 13. ПОДСЧЕТ ВРЕМЕНИ.

```

```

;=====

```

```

S1

```

```

BTFSC  FLAG1,5  ;
GOTO   $+8      ;
BTFSS  FLAG1,4  ;
GOTO   $+6      ;
BTFSC  PORTC,0  ;СМЕНА ВКЛЮЧЕНИЯ ЗУММЕРА.
GOTO   $+3      ;
BSF PORTC,0     ;ВКЛЮЧИМ СИГНАЛ.
GOTO   $+2      ;
BCF PORTC,0     ;
MOVLW  .9       ;ЕСЛИ УЖЕ 9 СЕКУНД,
SUBWF  CL,0     ;
BC SH   ;ИДЕМ НА СРАВНЕНИЕ ДЕСЯТКОВ СЕКУНД.
INCF CL,1      ;ИНАЧЕ ПРИБАВИМ ЕДИНИЦУ.
RETURN

```

```

SH

```

```

CLRFCL  ;ОБНУЛИМ СЕКУНДЫ.
MOVFW  CH   ;ЕСЛИ ДЕСЯТКИ СЕКУНД
ADDLW  -5H  ;РАВНЫ 5,

```

BZ MIL ;ИДЕМ СРАВНИВАТЬ МИНУТЫ.
INCF CH,1 ;ИНАЧЕ УВЕЛИЧИМ ДЕСЯТКИ СЕКУНД.
RETURN

MIL

CLRFCH ;ОБНУЛИМ ДЕСЯТКИ СЕКУНД.
MOVFW ML ;ЕСЛИ ЕДИНИЦЫ МИНУТ
ADDLW -9H ;РАВНЫ 9,
BZ MIN ;ИДЕМ СРАВНИВАТЬ ДЕСЯТКИ.
INCF ML,1 ;ИНАЧЕ УВЕЛИЧИМ МИНУТЫ.
RETURN

MIN

BSF FLAG1,4 ;ПРОШЛО 10 МИНУТ КУРСОР
ВЫКЛЮЧАЕТСЯ.

BCF FLAG,7 ;НЕТ УСТАНОВКИ.
CLRFBKURC ;НЕТ КУРСОРА.
CLRFBML ;ОБНУЛИМ ЕДИНИЦЫ МИНУТ.
MOVFW MN ;ЕСЛИ ДЕСЯТКИ МИНУТ
ADDLW -5H ;РАВНЫ 5,
BZ HOL ;ИДЕМ СРАВНИВАТЬ ЧАСЫ.
INCF MN,1 ;ИЛИ УВЕЛИЧИМ ДЕСЯТКИ МИНУТ.
RETURN

HOL

BCF FLAG1,5 ;НАПОМИНАНИЕ ОБ АВАРИИ
НАПРЯЖЕНИЯ.

CLRFBMN ;ОБНУЛИМ ДЕСЯТКИ МИНУТ.
MOVFW HN ;ЕСЛИ ДЕСЯТКИ ЧАСОВ
ADDLW -2H ;РАВНЫ 2,
BZ HL4 ;ПРОВЕРИМ ЕДИНИЦЫ ЧАСОВ.
MOVFW HL ;ЕСЛИ ЕДИНИЦЫ ЧАСОВ РАВНЫ 9,
ADDLW -9H ;

```
BZ    $+3    ;УВЕЛИЧИМ ДЕСЯТКИ ЧАСОВ.  
INCF HL,1    ;ИЛИ УВЕЛИЧИМ ЕДИНИЦЫ ЧАСОВ.  
RETURN  
CLRFHL    ;  
INCF HH,1    ;  
RETURN
```

HL4

```
MOVFW    HL    ;ЕСЛИ ЕДИНИЦЫ ЧАСОВ  
ADDLW    -3H    ;РАВНЫ 3,  
BZ    HOH    ;ИДЕМ ОБНУЛЯТЬ.  
INCF HL,1    ;ИЛИ ПРИБАВИМ ЕДИНИЦУ.  
RETURN
```

HOH

```
CLRFHL    ;  
CLRF HH    ;ОБНУЛИМ ДЕСЯТКИ ЧАСОВ.  
RETURN    ;
```

;=====

; 14. ТАЙМЕР.

;=====

ТАИМ

```
BSF    FLAG,5    ;СЛЕДУЮЩАЯ ОСТАНОВКА ТАЙМЕРА.  
MOVLW    .9    ;ЕСЛИ УЖЕ 9 СЕКУНД,  
SUBWF    TCL,0    ;  
BC    $+3    ;ИДЕМ НА СРАВНЕНИЕ ДЕСЯТКОВ СЕКУНД.  
INCF TCL,1    ;ИНАЧЕ ПРИБАВИМ ЕДИНИЦУ.  
RETURN  
CLRFTCL    ;ОБНУЛИМ СЕКУНДЫ.  
MOVFW    TCH    ;ЕСЛИ ДЕСЯТКИ СЕКУНД  
ADDLW    -5H    ;РАВНЫ 5,
```

```

BZ    $+3    ;ИДЕМ СРАВНИВАТЬ МИНУТЫ.
INCF TCH,1    ;ИНАЧЕ УВЕЛИЧИМ ДЕСЯТКИ СЕКУНД.
RETURN
CLRFTCH ;ОБНУЛИМ ДЕСЯТКИ СЕКУНД.
MOVFW  TML ;ЕСЛИ ЕДИНИЦЫ МИНУТ
ADDLW  -9H ;РАВНЫ 9,
BZ    $+3    ;ИДЕМ СРАВНИВАТЬ ДЕСЯТКИ.
INCF TML,1    ;ИНАЧЕ УВЕЛИЧИМ МИНУТЫ.
RETURN
CLRFTML ;ОБНУЛИМ ЕДИНИЦЫ МИНУТ.
MOVFW  TMH ;ЕСЛИ ДЕСЯТКИ МИНУТ
ADDLW  -5H ;РАВНЫ 5,
BZ    $+3    ;УВЕЛИЧИМ ЧАСЫ.
INCF TMH,1    ;ИЛИ УВЕЛИЧИМ ДЕСЯТКИ МИНУТ.
RETURN
CLRFTMH ;ОБНУЛИМ ДЕСЯТКИ МИНУТ.
MOVFW  THL ;ЕСЛИ ЕДИНИЦЫ ЧАСОВ РАВНЫ 9,
ADDLW  -9H ;
BZ    $+3    ;УВЕЛИЧИМ ДЕСЯТКИ ЧАСОВ.
INCF THL,1    ;ИЛИ УВЕЛИЧИМ ЕДИНИЦЫ ЧАСОВ.
RETURN
CLRFTHL ;ОБНУЛИМ ЧАСЫ.
MOVFW  TNH ;ЕСЛИ ДЕСЯТКИ ЧАСОВ РАВНЫ 9,
ADDLW  -9H ;
BZ    $+3    ;ОБНУЛИМ.
INCF TNH,1    ;ИЛИ УВЕЛИЧИМ ДЕСЯТКИ ЧАСОВ.
RETURN
CLRFTNH ;
RETURN

```

```

    BTFSS    FLAG,5    ;
    GOTO     $+4    ;
    BCF FLAG,2    ;ТАЙМЕР ВЫКЛЮЧЕН.
    BCF FLAG,5    ;СЛЕДУЮЩЕЕ ВКЛЮЧЕНИЕ ТАЙМЕРА.
    RETURN     ;
    CLRFTCL ;ВСЕ ОБНУЛЯЕМ.
    CLRFTCH ;
    CLRFTML ;
    CLRFTMH ;
    CLRFTHL ;
    CLRFTHH ;
    BSF FLAG,2    ;ВКЛЮЧИМ ТАЙМЕР.
    RETURN

```

```

;=====
; 15. АЦП - ПРЕОБРАЗОВАНИЯ (ИЗМЕРЕНИЕ ВХОДНЫХ
ВЕЛИЧИН).

```

```

;=====
ADP
MOVLW    B'10000001';СИНХРОНИЗАЦИЯ ОТ RC
MOVWF    ADCON0    ;ГЕНЕРАТОРА, ВХОД 0, ВКЛЮЧЕНИЕ
АЦП (УВХ).
    CALL    ZAD    ;
    BSF    ADCON0,1 ;ВКЛЮЧИМ ПРЕОБРАЗОВАНИЕ.
    BTFSC    ADCON0,1 ;ОЖИДАЕМ ЗАВЕРШЕНИЯ
    GOTO     $-1    ;ПРЕОБРАЗОВАНИЯ.
    MOVFW    ADRESH ;ПЕРЕПИШЕМ                РЕЗУЛЬТАТ
ПРЕОБРАЗОВАНИЯ
    MOVWF    ТЕКН    ;В СТАРШИЙ ТЕКУЩИЙ РЕГИСТР.
    BSF    STATUS,5 ;ПЕРЕХОДИМ В БАНК 1.
    MOVLW    58    ;

```

```

MOVWF  FSR      ;ПО КОСВЕННОЙ АДРЕСАЦИИ
MOVFW  ADRESL    ;ЗАПИСЬ МЛ. РЕГИСТРА АЦП
MOVWF  INDF      ;В РЕГИСТР ТЕКЛ.
BCF  STATUS,5 ;ПЕРЕХОДИМ В БАНК 0.
CALL  COMPA      ;
GOTO  BINDEC     ;ПЕРЕКОДИРУЕМ В 2_10 КОД.

```

ZAD

```

MOVLW  .5        ;ЗАДЕРЖКА 20 МКС.
ADDLW  -1        ;
BTFSS  STATUS,2  ;
GOTO  $-2        ;
RETURN

```

;=====

; 16. СПРАВНЕНИЕ С УСТАНОВКОЙ.

;=====

COMPA

```

TSTF TEKL      ;
BTFSC  STATUS,2;
GOTO  VUKL     ;
MOVFW  YCTL     ;УСТАНОВКА МАКСИМУМА.
SUBWF  TEKL,0   ;ИЗМЕРЕНИЕ,
BTFSC  STATUS,0;ЕСЛИ БОЛЬШЕ ИЛИ РАВНО,
GOTO  VUKL     ;ТО ВКЛЮЧАЕТСЯ ЗУММЕР.
MOVFW  YCTLI    ;УСТАНОВКА МИНИМУМА.
SUBWF  TEKL,0   ;ИЗМЕРЕНИЕ,
BTFSS  STATUS,0;ЕСЛИ МЕНЬШЕ,
GOTO  VUKL     ;ТО ВКЛЮЧАЕТСЯ ЗУММЕР.
BTFSC  STATUS,2;ЕСЛИ РАВНО,
GOTO  VUKL     ;ТО ВКЛЮЧАЕТСЯ ЗУММЕР.
BCF  FLAG1,4   ;ЗУММЕР ВЫКЛЮЧЕН.

```

```
BCF FLAG1,5 ;ЗУММЕР ВЫКЛЮЧЕН.  
BCF PORTC,0 ;ВЫКЛЮЧИМ СИГНАЛ.  
RETURN
```

VUKL

```
    BTFSC    PORTA,KY;  
    GOTO     $+5 ;  
    BSF FLAG1,5 ;ПОСТАВИМ СТОРОЖОК НА 1 ЧАС.  
    BCF PORTC,0 ;ВЫКЛЮЧИМ СИГНАЛ.  
    BTFSC    FLAG1,5 ;ЕСЛИ 1 ЧАС ПРОШЕЛ, ТО ВКЛЮЧИМ  
СИГНАЛ.  
    RETURN  
    BTFSC    FLAG1,4 ;ЕСЛИ ВЫХОД УЖЕ ВЫКЛЮЧЕН,  
    RETURN   ;СРАВНЕНИЙ НЕТ.  
    BSF PORTC,0 ;ВКЛЮЧИМ СИГНАЛ.  
    BSF FLAG1,4 ;ЗУММЕР ВКЛЮЧЕН.  
    RETURN
```

```
;=====
```

; 17. ПЕРЕКОДИРОВКА ИЗ 16-И РАЗРЯДНОГО 2-ГО В 5-
РАЗРЯДНОЕ 2-10-Е.

; АЛГОРИТМ ПЕРЕКОДИРОВКИ ОСНОВЫВАЕТСЯ НА
ПРИБАВЛЕНИИ 3 В МЛАДШИЙ

; И СТАРШИЙ ПОЛУБАЙТЫ. ЕСЛИ РЕЗУЛЬТАТ СПЕРЕНОСОМ 1
В 3 РАЗРЯД ($10=7+3$), ТО ЗАПИСЫВАЕМ

; НОВОЕ ЗНАЧЕНИЕ В РЕГИСТР. ВЫПОЛНЯЕМ 16 РАЗ СДВИГАЯ
БИТЫ РЕГИСТРОВ.

```
;=====
```

```
BINDEC  
MOVLW .16 ;ЗАПИШЕМ ЧИСЛО СДВИГОВ  
MOVWF COU ;В СЧЕТЧИК.  
BIDE
```

BCF STATUS,0 ;ОБНУЛИМ БИТ "С".

RLF TEKL,1 ;СДВИНЕМ ПЕРЕКОДИРУЕМОЕ

RLF TEKH,1 ;ЧИСЛО ПЕРЕМЕЩАЯ ЕГО СТАРШИЙ БИТ

RLF EDA,1 ;В МЛАДШИЙ БИТ РЕГИСТРОВ

RLF DEA,1 ;РЕЗУЛЬТАТА.

DECFSZ COU,1 ;ЗАФИКСИРУЕМ СДВИГ В СЧЕТЧИКЕ.

GOTO RASDEC ;ПРОВЕРИМ ПОЛУБАЙТЫ НА СЕМЕРКУ.

GOTO MESTO ;ЕСЛИ СЧЕТЧИК ПУСТ, ЗАПОЛНИМ РЕГИСТРЫ

ИНДИКАЦИИ.

RASDEC

MOVLW EDA ;ЗАПИШЕМ АДРЕС РЕГИСТРА

MOVWF FSR ;В РЕГИСТР КОСВЕННОЙ АДРЕСАЦИИ.

CALL BCD ;ПРОВЕРИМ ЗНАЧЕНИЕ РЕГИСТРА НА 7.

MOVLW DEA ;АНАЛОГИЧНЫЕ ОПЕРАЦИИ ПРОДЕЛАЕМ

MOVWF FSR ;С ДРУГИМИ РЕГИСТРАМИ.

CALL BCD ;

GOTO VIDE ;ПОЙДЕМ ПОВТОРЯТЬ СДВИГ.

BCD

MOVLW 3 ;0000 0011

ADDWF 0,0 ;ПРИБАВИМ 3 К РЕГИСТРУ И РЕЗУЛЬТАТ

MOVWF TEMP ;ЗАПИШЕМ ВО ВРЕМЕННЫЙ РЕГИСТР.

BTFSK TEMP,3 ;ПРОВЕРИМ 3 БИТ И ЕСЛИ ОН РАВЕН НУЛЮ,

MOVWF 0 ;ПРОПУСКАЕМ ЗАПИСЬ РЕЗУЛЬТАТА В РЕГИСТР.

MOVLW 30 ;48=0011 0000

ADDWF 0,0 ;ПРИБАВИМ 3 К СТАРШЕМУ ПОЛУБАЙТУ РЕГИСТРА

И РЕЗУЛЬТАТ

MOVWF TEMP ;ЗАПИШЕМ ВО ВРЕМЕННЫЙ РЕГИСТР.

BTFSK TEMP,7 ;ЕСЛИ БИТ ЕДИНИЧНЫЙ,

MOVWF 0 ;ТО ЗАПИШЕМ НОВОЕ ЗНАЧЕНИЕ В РЕГИСТР.

RETURN ;ВЕРНЕМСЯ ДЛЯ ЗАГРУЗКИ НОВОГО ЗНАЧЕНИЯ
РЕГИСТРА.

=====

; 18. ИЗВЛЕКАЕМ ПОЛУБАЙТЫ ИЗ РЕГИСТРОВ СЧЕТА В
РЕГИСТРЫ ИНДИКАЦИИ.

=====

MESTO

MOVLW B'00001111';ИЗВЛЕКАЕМ ПОЛУБАЙТЫ
ANDWF DEA,0 ;В РЕГИСТРЫ ИНДИКАЦИИ.
MOVWF COI ;
MOVLW B'11110000';
ANDWF EDA,0 ;
MOVWF DEI ;
SWAPF DEI,1 ;
MOVLW B'00001111';
ANDWF EDA,0 ;
MOVWF EDI ;
CLRFEDA ;
CLRFDEA ;
RETURN ;

=====

; 19. СМЕНА ИНДИКАЦИИ ПРИ СМЕНЕ РЕЖИМА.

=====

INDH

BTFSC FLAG,7 ;ЕСЛИ КУРСОР ЕСТЬ,
CALL YCT ;ТО ИДЕМ НА УСТАНОВКУ.
MOVLW .10 ;ПУСТО.
CALL SEG ;ЗАПОЛНИМ РЕГИСТРЫ ИНДИКАЦИИ.
MOVWF R1 ;
MOVWF R2 ;

```

MOVWF  R8  ;
BTFSS  FLAG,6  ;
GOTO   $+6  ;
MOVLW  .11  ;ПРОЧЕРК СРЕДНИЙ.
CALL   SEG  ;
MOVWF  R5  ;
BCF  FLAG,6  ;
GOTO   $+5  ;
MOVLW  .12  ;ПРОЧЕРК НИЖНИЙ.
CALL   SEG  ;
MOVWF  R5  ;
BSF  FLAG,6  ;
MOVFW  ML  ;
CALL   SEG  ;
MOVWF  R3  ;
MOVFW  MH  ;
CALL   SEG  ;
MOVWF  R4  ;
MOVFW  HL  ;
CALL   SEG  ;
MOVWF  R6  ;
MOVFW  HH  ;
CALL   SEG  ;
MOVWF  R7  ;
BSF  FLAG,1  ;ВКЛЮЧИМ РЕЖИМ ЧАСОВ.
CLRFZPT  ;
RETURN  ;

```

INDT

```

BTFSS  PORTA,KY;ЕСЛИ КНОПКА НАЖАТА,
CALL   T00  ;ТО ИДЕМ ОБНУЛЯТЬ ТАЙМЕР.

```

```

MOVFW   TCL ;ЗАПОЛНИМ РЕГИСТРЫ ИНДИКАЦИИ
CALL    SEG ;МИНУТ И СЕКУНД ПРАВЫХ ЧАСОВ.
MOVWF   R1  ;
MOVFW   TCH ;
CALL    SEG ;
MOVWF   R2  ;
MOVFW   TML ;
CALL    SEG ;
MOVWF   R4  ;
MOVFW   TMH ;
CALL    SEG ;
MOVWF   R5  ;
MOVLW   .11 ;ПРОЧЕРК.
CALL    SEG ;
MOVWF   R3  ;
MOVWF   R6  ;
MOVFW   THL ;
CALL    SEG ;
MOVWF   R7  ;
MOVFW   THH ;
CALL    SEG ;
MOVWF   R8  ;
BCF     FLAG,1 ;РЕЖИМ ЧАСЫ ВЫКЛЮЧЕН.
BCF     FLAG,7 ;НЕТ УСТАНОВКИ.
RETURN  ;

```

INDU

```

MOVLW   .10 ;ПУСТО.
CALL    SEG ;ЗАПОЛНИМ РЕГИСТРЫ ИНДИКАЦИИ.
MOVWF   R1  ;
MOVWF   R2  ;

```

```

MOVWF R3 ;
MOVWF R7 ;
MOVWF R8 ;
MOVFW COI ;
CALL SEG ;
MOVWF R6 ;
MOVFW DEI ;
CALL SEG ;
MOVWF R5 ;
MOVFW EDI ;
CALL SEG ;
MOVWF R4 ;
BSF ZPT,4;
RETURN ;

```

```

;=====

```

```

; 20. РАСПРЕДЕЛИТЕЛЬ.

```

```

;=====

```

```

PAC

```

```

BTFSS FLAG1,1 ;ЖДЕМ 1 СЕКУНДУ.
GOTO $-1 ;
CALL ADP ;
CALL КНОР ;НА ПРОВЕРКУ КНОПОК.
CALL IND ;НА ИНДИКАЦИЮ.
BCF FLAG1,1 ;
GOTO PAC ;ПОВТОРИМ.

```

```

;=====

```

```

; 21. УСТАНОВКА ВРЕМЕНИ.

```

```

;=====

```

```

YCO

```

```

INCF ML,1 ;УВЕЛИЧИМ ЕДИНИЦЫ МИНУТ.

```

```

    MOVLW    .10        ;НЕ БОЛЕЕ 9.
    SUBWF    ML,0        ;
    SKPNC                    ;ЕСЛИ БОЛЬШЕ ИЛИ РАВНО 10,
    CLRFML                    ;ТО ОБНУЛИМ.
    RETURN

```

YC1

```

    INCF    MH,1        ;УВЕЛИЧИМ ДЕСЯТКИ МИНУТ.
    MOVLW    .6          ;НЕ БОЛЕЕ 5.
    SUBWF    MH,0        ;
    SKPNC                    ;ЕСЛИ БОЛЬШЕ ИЛИ РАВНО 6,
    CLRFMH                    ;ТО ОБНУЛИМ.
    RETURN

```

YC2

```

    INCF    HL,1        ;УВЕЛИЧИМ ЕДИНИЦЫ ЧАСОВ.
    MOVLW    .10        ;НЕ БОЛЕЕ 9.
    SUBWF    HL,0        ;
    SKPNC                    ;
    CLRFHL                    ;ЕСЛИ БОЛЬШЕ, ТО ОБНУЛИМ.
    MOVFW    HH          ;ПЕРЕКОДИРУЕМ В ДВОИЧНЫЙ
    CALL     DEBIN        ;КОД ДЕСЯТКИ.
    ADDWF    HL,0        ;ПРИБАВИМ ЕДИНИЦЫ
    MOVWF    HOU          ;ДВОИЧНОЕ ЗНАЧЕНИЕ НЕ ДОЛЖНО
    ADDLW    -18H        ;ПРЕВЫШАТЬ - 24.
    SKPC                    ;ЕСЛИ БОЛЬШЕ ИЛИ РАВНО 24,
    RETURN
    CLRFHOU                    ;ТО ОБНУЛИМ ЧАСЫ ДВОИЧНЫЕ
    CLRFHL                    ;И РАЗРЯДЫ СТАРШИЙ
    CLRFHH                    ;И МЛАДШИЙ.
    RETURN

```

YC3

```

INCF HH,1      ;УВЕЛИЧИМ ДЕСЯТКИ ЧАСОВ.
MOVLW  .3      ;НЕ БОЛЕЕ 2.
SUBWF  HH,0    ;
SKPNC                      ;
CLRFHH          ;ЕСЛИ БОЛЬШЕ, ТО ОБНУЛИМ.
    MOVFW  HH      ;ПЕРЕКОДИРУЕМ В ДВОИЧНЫЙ
    CALL  DEBIN      ;КОД ДЕСЯТКИ.
    ADDWF  HL,0      ;ПРИБАВИМ ЕДИНИЦЫ
    MOVWF  HOU      ;И ПОЛУЧИМ ДВОИЧНОЕ ЧИСЛО.
ADDLW  -18H      ;
    SKPC                      ;ЕСЛИ БОЛЬШЕ ИЛИ РАВНО 24,
    RETURN
    CLRFHOU          ;ТО ОБНУЛИМ ЧАСЫ ДВОИЧНЫЕ
    CLRFHL           ;И РАЗРЯДЫ СТАРШИЙ
    CLRFHH           ;И МЛАДШИЙ.
RETURN          ;ВЕРНЕМСЯ.
YCE
    CLRFCL           ;ОБНУЛЕНИЕ.
    CLRFCH           ;
    CLRFML           ;
    RETURN           ;

```

```

;=====

```

```

END

```

```

;=====

```