**СОДЕРЖАНИЕ**

Введение

1.1 Принцип действия устройства

1.2 Блок-схема устройства

2 Принцип действия блоков схемы

2.1 Первичный преобразователь

2.2 Расчет первичного преобразователя

2.3 Нормирующий преобразотель

2.4 Аналого - цифровой преобразователь

2.5 Выбор мультиплексора

2.6 Выбор микроконтроллера

2.7 Передача информации через последовательный порт

2.8 Настройка контроллера на работу с последов. портом

3.Программирование микроконтроллера

3.1Блок-схема предустановок

3.2Блок-схема измерения веса

3.3Блок-схема вывода инфрмации о весе в компьютер

3.4Блок-схема автокалебровки.

3.5Управление микроконтроллером с компьютера

3.6Программа на ассемблере для микроконтроллера MCS-51

Список литературы

Приложения

В наше время, когда наука и техника достигли очень больших высот, становится возможным моделирование приборов, основанных на микропроцессорах и микроконтроллерах. На этих элементах цифровой электроники можно наряду с новыми аппаратами создавать давно известные, такие как вольтметры, частотомеры, электрические весы, но на базе контроллеров. У таких приборов множество преимуществ. Например, полученные с них данные могут выводиться на экран дисплея или на жидкокристаллический дисплей, они более точны и более универсальны, т.к. перепрограммировав их, можно измерять параметры в других пределах, а если еще произвести небольшое усложнение схем, то можно добиться увеличения функциональных возможностей аппарата. Также значительным преимуществом аппаратов на микропроцессорах является возможность подключения их к компьютерам. В современных системах на различных производствах и т.п. повсеместно используют в виде посредника между датчиком и компьютером быстродействующие аппараты на основе микроконтроллеров. Такие аппараты значительно дешевле, чем на основе компьютеров, но по своим характеристикам не на много им уступают.

Также очень часто приборы на микропроцессорах используются в системах автоматического управления. Процесс протекает строго согласно программам, записанным в память процессоров.

Часто, приборы на микроконтроллере производят прием информации от датчика, преобразуют в определенный формат полученные данные, если необходимо - хранят данные, фильтруют, уплотняют и только после этого передают в компьютер. Таким образом, компьютер не "отвлекается" от главного процесса, -вся "черновая" работа производится контроллером. Информация анализируется компьютером только в случае необходимости, или когда компьютер свободен. Микроконтроллер может даже генерировать прерывание, - такое необходимо, если поступила экстраординарная информация и требуется немедленная реакция на данное событие.

Целью данной работы является проектирование цифровых весов для взвешивания вагонов. Происходит анализ веса, пересчет (суммирование). Затем информация передается в компьютер. Причем можно организовать запись таким образом, чтобы запись производилась только при появлении состава.

Поскольку измерение веса происходит в статике, здесь особенно не важна скорость взвешивания. По этому, применение DSP не целесообразно.

**1.1 Принцип действия устройства**

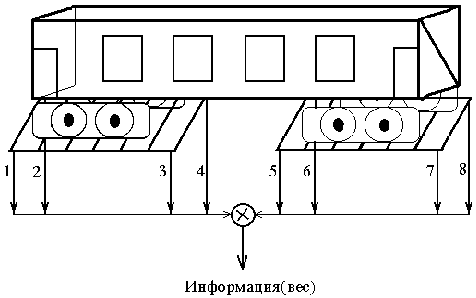


Рис. 1.1.

На рисунке 1.1. изображена схема измерения веса вагона.

Пронумерованными стрелками показано направление потока информации от датчиков. Общее число датчиков выбрано - 8.

Теоретически для описания (задания) плоскости достаточно всего 3 точки. В данном случае необходимо 4 точки, т.к. возможно, что нагрузка на различные точки будет неравномерной, причем может очень сильно различаться. Весы двух платформенные - по этому, общее число точек контроля (датчиков) в два раза больше - 8 штук.

Так, как нагрузка распределяется между опорными точками, в которых расположены датчики, то каждый датчик преобразует частичный вес вагона. Для получения общего веса вагона необходимо просуммировать все частичные в*е*сы. Это можно сделать как до оцифровки, так и после. Оцифровка необходима для того, чтобы передать информацию (полученный вес) в компьютер. Причем данные в компьютер могут поступать в любой форме.

Предполагается располагать само устройство возле недалеко от платформ и рельсов. Дело в том, что полезный сигнал, снимаемый с датчиков очень мал (48...48000 мкВ). По этому, для уменьшения уровня помех и уменьшения потерь на соединительных кабелях расстояние должно быть малым. Само устройство подключается к компьютеру через последовательный порт.

Скорость передачи информации в компьютер можно задавать произвольно, т.к. измерение производится в статике.

**1.2 Блок-схема устройства**

+

Начало

Первичное преобразование веса для 1-го канала

Нормирующий преобразователь электрического сигнала в Unorm для 1-го канала

Аналого - цифровое преобразование

Запись полученного кода в контроллер

Первичное преобразование веса для 8-го канала

Нормирующий преобразователь электрического сигнала в Unorm для 8-го канала

Мультиплексирование сигналов

**...**

**...**

Опрошены все каналы

\_

Изменение адреса активного канала

Суммирование полученного веса и передача веса в компьютер

Конец

Рис 1.2.1.

На рисунке 1.2.1. изображена блок схема, поясняющая принцип работы схемы. Группа первичных преобразователей производит преобразование веса в соответствующие уровни напряжения. Далее нормирующие преобразователи формируют уровни напряжения Unorm для каждого канала в заданном диапазоне для того, чтобы напряжение было "удобным" для дальнейших с ним действий. Этот процесс происходит непрерывно и независимо от всех остальных.

В то же время происходит цифро-аналоговое преобразование выбранного канала. Причем используется один АЦП, поэтому сигналы мультиплексируются. Мультиплексор управляется микроконтроллером. Полученный код веса записывается в микроконтроллер. Затем, МК меняет адрес канала и процесс преобразования повторяется для нового канала. После того, как будут опрошены все каналы и будут записаны полученные коды, характеризующие частичный вес, происходит суммирование частичного веса. Эта сумма характеризует полный вес вагона.

Затем, данные с устройства передаются в последовательном коде в компьютер. Для этого используется драйвер последовательного порта RS-232. Программа-терминал на компьютере, при необходимости, считывает данные из буфера последовательного порта и высвечивает массу вагона на мониторе.

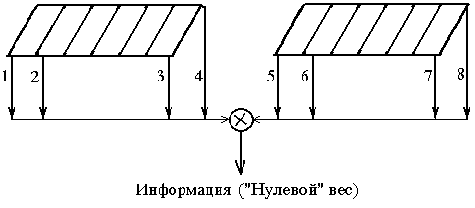


Рис 1.2.2.

На рис. 1.2.2. показана схема калибровки системы.

Калибровка в данном случае представляет собой коррекцию нуля - устройством учитывается вес платформы, рельсов и т.п.

Калибровка осуществляется автоматически и управляется микроконтроллером. Запуск коррекции осуществляется по нажатию кнопки оператора. При этом на мониторе компьютера должен показываться ноль. Следует отметить, что коррекцию производит непосредственно микроконтроллер по каждому каналу, причем компьютер в данной операции не участвует.

Наличие корректировки повышает точность измерения и повышает автономность (автоматичность) данной системы.

При необходимости можно сделать так, чтобы при калибровке, устройство посылало на компьютер "нулевой" вес - т.е. вес при отсутствии вагона - на компьютер. Это позволит контролировать правильность работы устройства.

**2. Принцип действия блоков схемы**

**2.1 Первичный преобразователь**

В роли первичного преобразователя выступает тензорезисторный мост. Исходя из количества тензодатчиков и максимальной массы взвешиваемого вагона выбираем тензодатчики типа ДСТВ -1 , рассчитанные на 16 тонн.



Система с такими тензодатчиками может взвешивать вагоны, массой до . При этом существует запас на 38 тонн. Этот запас необходим, т.к. возможно, что вес не будет равномерно распределен между контрольными точками (датчиками). К тому же в взвешиваемый вес будет включаться и вес платформ.



Здесь учитывается, также то, что при торможении поезда на платформу действует дополнительное усилие. Т.к. данные тензодатчики способны выдерживать дополнительную нагрузку на 50% от номинальной, то

,



что позволяет выдерживать общую нагрузку

, т.е. более, чем в 2 раза.



Основные параметры тензодатчиков ДСТВ - 1. Табл. 2.1.

|  |  |
| --- | --- |
| Параметр | Значение |
| Номинальная нагрузка |  |
| Нижний предел измерения |  |
| Значение рабочего коэффициента передачи (РКП) при номинальной нагрузке |  |
| Значение начального коэффициента передачи (НКП) не более |  |
| Категория точности | 0.25 |
| Входное сопротивление |  |
| Выходное сопротивление |  |
| Наибольшее допустимое значение напряжения питания постоянного или переменного тока с частотой до 1.5 кГц |  |
| Допустимое значение перегрузки, процентов выше номинальной нагрузки  кратковременной  длительной | 100%  50% |
| Предельный диапазон рабочих температур | -30°С .... +50°С |
| Габариты, (мм) | 155 х 155 х 230 |

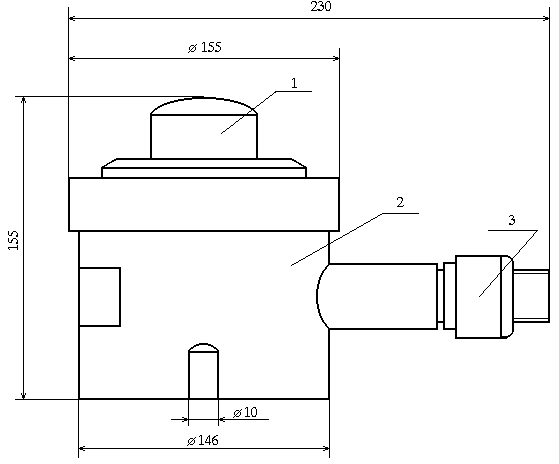


Рис. 2.1.1. Внешний вид и устройство

На рисунке 2.1.1. изображен чертеж тензодатчиков ДСТВ - 1.

Здесь 1 - это концентрическая упругая поверхность, чья деформация преобразуется с помощью тензорезисторов в электрический сигнал; 2 - корпус первичного преобразователя, служащий для защиты устройства от попадания внутрь твердых веществ и воды; 3 - трубка, защищающая соединительные провода.

При оказывании давления на концентрическую поверхность (1), она деформируется. На эту поверхность с внутренней стороны нанесены тензорезисторы. Причем тензорезисторы включены по схеме моста (рис. 2.1.2). К тензорезисторам подводится напряжение , при этом, при условии максимальной нагрузки в на выходе моста будет напряжение, пропорциональное максимальной нагрузке .



Рис. 2.1.2. Тензорезисторный мост.

**2.2 Расчет первичного преобразователя**

Выбираем напряжение питания .



При этом, следуя формуле чувствительности тензодатчика .



Это означает, что при номинальной нагрузке в на выходе первичного преобразователя будет .



Исходя из ТЗ необходимо, чтобы ошибка составляла 0.1%. Для получения такой точности вычислим минимальный шаг квантования :



**2.3 Нормирующий преобразователь**

Учитывая выходные параметры первичного преобразователя, выбираем усилительный элемент.

Т.к. полезный сигнал будет поступать с большим синфазным сигналом, приблизительно равным напряжению питания тензорезисторного моста , то необходимо, чтобы усилитель имел очень высокий уровень подавления синфазного сигнала.



Это соответствует .



Значит, необходимо выбрать такой усилительный элемент, который обеспечивает ослабление синфазного сигнала минимум в .



Коэффициент усиления должен быть таким, чтобы максимальный уровень полезного сигнала усиливался до опорного напряжения АЦП, что соответствовало бы максимальному коду. Опорное напряжение выбираем стандартное . Тогда



Тогда выбираем т.н. *инструментальный операционный усилитель* фирмы BURR BROWN INA128.

В инструментальных усилителях коэффициент усиления задается с помощью подключения внешнего сопротивления RG, значение которого высчитывается для INA128 по следующей формуле

, где



Тогда , выбираем стандартное сопротивление .



Полученный коэффициент усиления будет равен

.



Такое значение допускается, т.к. первичный преобразователь выбирался с запасом, т.е. уровень полезного сигнала на выходе тензорезисторного моста не будет достигать своего максимального значения

.

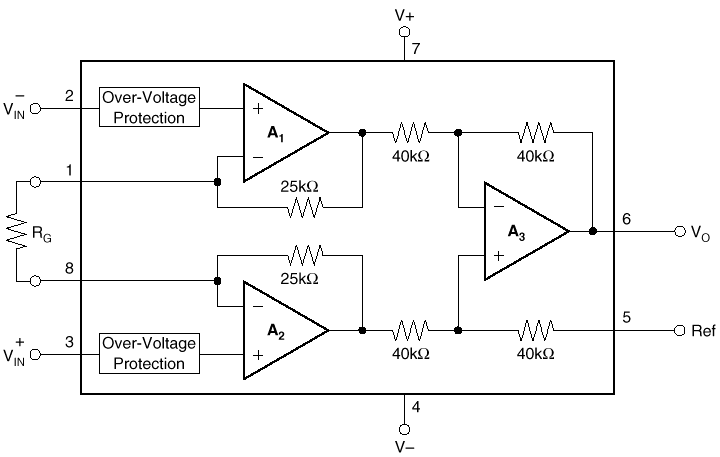


Это соответствует нагрузке на все 8 датчиков

.



При этом остается запас .



**2.3 Внутренняя структура инструментального усилителя INA128.**

**2.4 Аналого - цифровой преобразователь**

Аналого - цифровой преобразователь (АЦП) выбирается исходя из ТЗ.

Нам необходимо взвешивать груз с максимальной весом . При этом необходимо производить измерение с погрешностью 0.1%, что соответствует



.



Здесь - пересчетное значение младшего значащего разряда - шаг квантования. Для того, чтобы покрыть весь диапазон с шагом, равным минимальному шагу квантования необходимо сделать шагов.



Это соответствует . Значит, сигнал необходимо оцифровывать с разрядностью 10. Это минимальное значение разрядности АЦП. В реальных условиях необходимо иметь запас. Поэтому выбираем разрядность АЦП - 12.



Частота дискретизации может быть любой, т.к. процесс измерения происходит в статике.

Выбираем АЦП фирмы BURR BROWN ADS7820.

Передача информации осуществляется в параллельном коде в микроконтроллер по управляющему сигналу с МК. Затем МК мультиплексирует входные сигналы и процесс повторяется.

После обработки всех линий, МК анализирует полученные данные. Причем в это время АЦП не выбран. Это означает, что его выходы находятся в высоко импедансном состоянии и при необходимости могут обслуживать другое устройство.

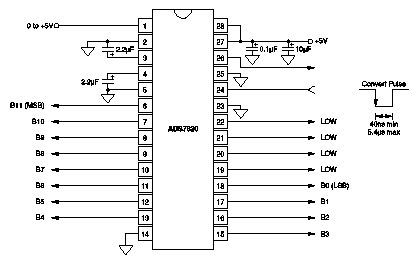


Рис. 2.4.1. Схема включения АЦП ADS7820.

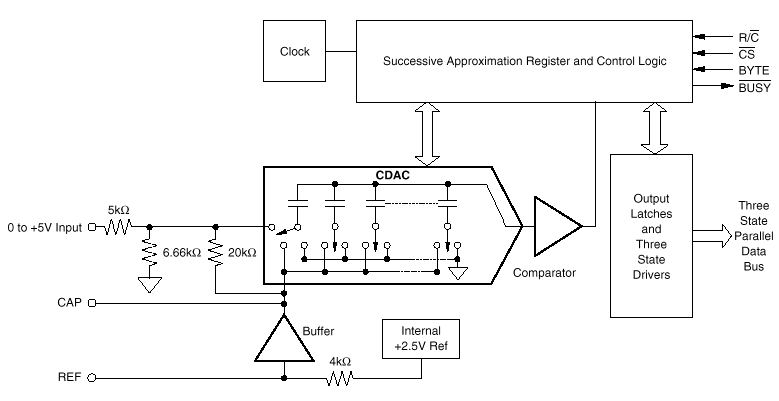


Рис. 2.4.2. Внутренняя структура АЦП.

Для настройки данного АЦП на работу необходимо на выход BYTE подать сигнал низкого уровня. При этом на D0 ... D11 будет поступать код, соответствующий входному аналоговому преобразующему сигналу.

R/C - Чтение/Преобразование. При перепаде из ноля в единицу (положительном фронте) на этом входе разрешается считывание кода с параллельного порта. При перепаде из единицы в ноль (отрицательном фронте) запуск преобразования аналогового сигнала в код.

CS - Выбор кристалла. При логическом нуле на этом входе происходит активация АЦП. Вместе с R/C - запускает преобразование. При логической единице на этом входе происходит перевод всех линий параллельного порта в высокоимпедансное состояние.

BUSY - При запуске преобразования на этом выходе устанавливается логический ноль. Сброс нуля происходит, когда преобразование закончится.

Предлагается подвести от вывода R/C АЦП линию к выводу INT0 и INT1 микроконтроллера через логическую схему.

**2.5 Выбор мультиплексора**

Согласно ТЗ, необходимо работать сразу с восемью аналоговыми сигналами. Для того, чтобы можно было оцифровывать все 8 сигналов одним АЦП необходимо использовать аналоговый мультиплексор. При этом нет требований к его быстродействию.

Необходимо, чтобы он мог пропускать напряжение до 2V.

Выбор канала должен осуществляться с помощью трех управляющих выводов, задающих номер канала. Этим будет заниматься микроконтроллер.

Выбираем мультиплексор фирмы BURR BROWN MPC508A.

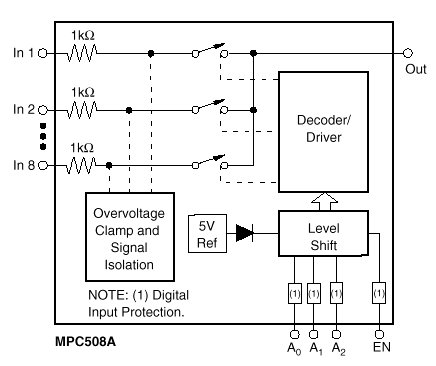


Рис. 2.5.1. Внутренняя структура мультиплексора.

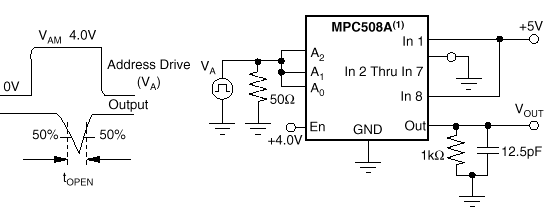


Рис. 2.5.2. Вариант схемы включения мультиплексор MPC508A.

**2.6 Выбор микроконтроллера**

В данном случае, согласно ТЗ, перед проектировщиком не ставится каких либо особых ограничений или требований. Т.к. процесс измерения веса происходит в статике, то нет необходимости в очень высоком быстродействии схемы. По этому не целесообразно проектировать данное устройство на DSP.

Одним из самых распространенных микроконтроллеров на сегодняшний день является МК семейства MCS-8051, первоначально разработанные фирмой INTEL.

INTEL MCS-8051 предназначен для построения контроллеров и микро-эвм различного назначения, отличающихся низкими аппаратными затратами при сохранении универсальности и быстродействия. Область применения MCS-8051 - от локальных систем автоматики до устройств управления бытовыми приборами.

Основными программно-доступными устройствами MCS-8051 являются:

1) 8-разрядный аккумулятор а;

2) 8-разрядный вспомогательный регистр в;

3) триггеры признаков результата: C (переноса), AC(вспомогательного переноса), OV (переполнения), P (четности);

4) триггеры выбора банка рабочих регистров RS0 и RS1;

5) триггер программно-управляемого флага F0;

6) 16-разрядный счетчик команд PC;

7) 16-разрядный регистр указателя данных DPTR;

8) 8-разрядный регистр указателя стека SP;

9) внутренняя память программ емкостью 4 кбайт, расширяемая внешними устройствами до 64 кбайт;

10) внутренняя память данных емкостью 128 байт, в которой размещается от одного до четырех банков рабочих регистров R0-R7, область стека и побитово адресуемая область памяти;

11) внешняя память данных емкостью до 64 кбайт;

12) два программируемых 16-разрядных таймера-счетчика;

13) программируемый двухнаправленный последовательный порт ввода-вывода и соответствующие устройства управления;

14) четыре 8-разрядных двухнаправленных параллельных порта ввода-вывода;

15) двухуровневая приоритетная система прерываний.

Предлагвется использовать микроконтроллер MCS-8051 следующим образом:

Порт Р0 и Р2 будут работать с данными, поступающими с АЦП.

Порт Р1 будет настраивать мультиплексор и АЦП.

Калибровка системы будет производиться по сигналу с компьютера.

Калибровка, также, будет осуществляться по прерыванию INT0.

С помощью RxD и TxD будет осуществляться связь с компьютером.

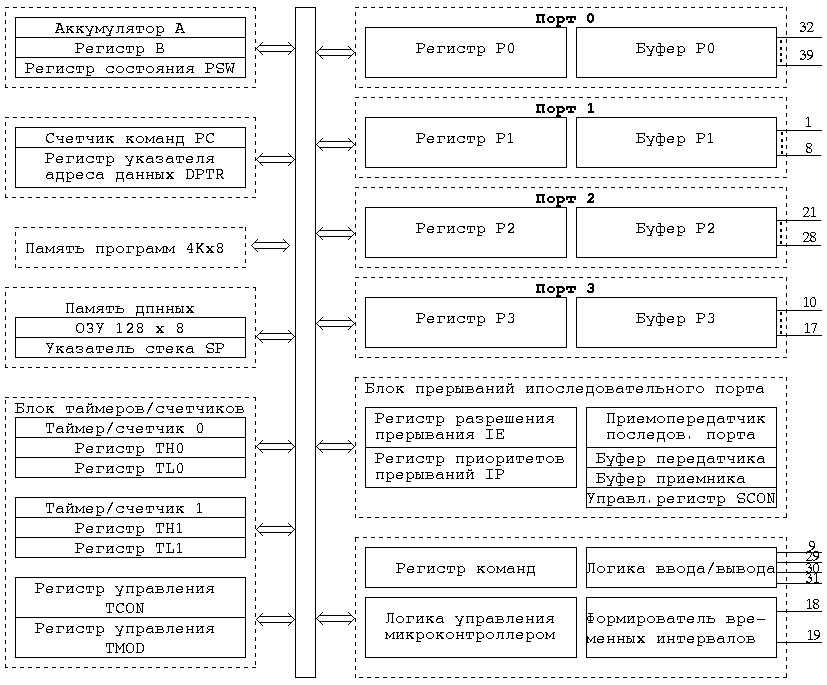


Рис. 2.6.1. Внутренняя структура микроконтроллера INTEL MCS-8051.



Рис. 2.6. Микроконтроллер INTEL MCS-8051

**2.7 Передача информации через последовательный порт.**

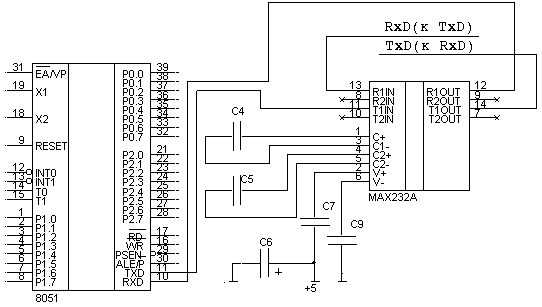


Рис. 2.7.

На рисунке 2.7. изображена электрическая принципиальная схема передачи информации от микроконтроллера в последовательный порт компьютера посредством комплексного преобразования напряжений с помощью драйвера последовательного интерфйса RS232A.

Главная функция устройства RS232A - согласование напряжения между устройствами. Дело в том, что напряжения логической единицы и нуля для микроконтроллера и компьютерного COM - порта различаются по уровням и полярности напряжений.

Табл.2.5.

|  |  |  |
| --- | --- | --- |
| *Логический Уровень* | *Уровень напряжения для МК* | *Уровень напряжения для COM* |
| 0 | 0 ... 0.4 | +12V |
| 1 | 4.75 ... 5.25 | − 12V |

Напряжение питания для RS232A - 5V.

Напряжение питания преобразовывается до уровня ±12V. Затем в зависимости от того, какой логический уровень пришел на вход драйвера, ключи подключают к выходу напряжение необходимой полярности в соответствии с таблицей 2.7. Также, драйвер RS232A выполняет некоторые дополнительные функции - защищта от статического электричества, гальваническая развязка и т.д.

Следуя руководству по эксплуатации данного драйвера необходимо, чтобы номиналы всех конденсаторов С1 − С4 были равны. Причем для RS232A С1 − С4 равно 0.1 μF.

Конденсатор Сst выбираем 10 μF. Он служит для сглаживания случайных скачков напряжения питания RS232A.

**2.8 Настройка контроллера на работу с последовательным портом**

В данном микроконтроллере INTEL MCS-8051 есть специальные выводы RxD и TxD - линии, соответственно ввода и вывода информации, представленной в последовательном коде. Прием или передача информации программой осуществляется с помощью регистра SBUF.

Архитектура контроллера следующая. При передаче информации через последовательный порт в регистре SCON устанавливается флаг Т1,а при приеме информации - флаг R1. В регистре РCON находится бит SMOD, установка которого означает работу с удвоением частоты.

Для установки скорости передачи по последовательному порту необходимо записать в регистр счетчика TH1 и TL1 число *N*, определяемое по следующей формуле:



В этой формуле *Fosc -* частота резонатора (кварца)

*SMOD* - бит удвоения частоты

*BR* - Baud Rate, скорость передачи данных.

В нашем случае мы работаем без удвоения частоты, т.е. *SMOD*=0.

Частота резонатора *Fosc =* 8 MHz.

Скорость передачи данных *BR* = 300 Бод.



Формат посылки изображен на рисунке 2.8.



Рис. 2.8.

1. **Программирование микроконтроллера**

Программа для микроконтроллера состоит из трех основных блоков:

1. Предварительная установка. Запускается по RESET или при включении питания микроконтроллера.
2. Измерение веса (считывания кодов с АЦП, обрабатывающего последовательно 8 аналоговых каналов). Запускается по сигналу INT0, соответствующему сигналу готовности данных на выходе микросхемы аналого-цифрового преобразователя.
3. Вывод информации на компьютер. Запускается по вызову из подпрограмм измерения веса или обработки последовательного порта (по получению управляющего сигнала с компьютера).
4. Калибровка. Считывание кодов с АЦП, обрабатывающего последовательно 8 аналоговых каналов. Причем полученный код считается соответствующим нулевому весу. Данные значения будут вычитаться в последствии из кода, полученного при измерении веса вагона. Запускается по сигналу INT1, соответствующему сигналу готовности данных на выходе микросхемы аналого-цифрового преобразователя.
   1. **Блок-схема предустановок (настроек)**



**3.2 Блок-схема измерения веса**



**3.3 Блок-схема вывода информации о весе в компьютер**



Примечание: здесь DWORD - начальный адрес двух ячеек памяти, в которых содержится младший и старший байты кода, соответствующего измеренному весу для канала(в данном случае - для первого).

**3.4 Блок-схема автоматической калибровки**



Необходимо отметить, что подпрограмма калибровки запускается только по управляющему сигналу от компьютера. Дело в том, что если организовать запуск по включению, может оказаться, что на весах уже стоит вагон. При этом, на выходе устройства будет выдаваться код, соответствующий нулевому весу. Т.к. мы имеем дело с тяжелыми вагонами, может оказаться накладным откатывать вагон только для калибровки.

Калибровка должна производиться при отсутствии грузов (вагона и т.п.) на измерительной площадке.

**3.5 Управление микроконтроллером с компьютера**

При передаче на микроконтроллер управляющего слова, следующим же тактом происходит обработка прерывания, связанного с последовательным портом (у последовательного порта - наивысший приоритет). Далее происходит распознавание управляющего слова и действие согласно нему (см. табл.3.5).

Таблица 3.5.

|  |  |
| --- | --- |
| Управляющее слово | Действия, производимые устройством по УС |
| 00111100 | Обнуление ячеек памяти, содержащих смещение и результат преобразования. |
| 11110000 | Преобразование и передача в компьютер кода, соответствующего весу. |
| 00001111 | Передача в компьютер кода, соответствующего весу измеряемого вагона. |
| 11111111 | Калибровка устройства. |

При необходимости, этот список можно дополнить. Всего возможно запрограммировать 256 действий.

**3.5 Программа на ассемблере для микроконтроллера**

;R0-@ адрес получаемого кода 32-47

;R1-@ адрес смещения 52-67

;R4-счетчик задержки для АЦП

;R5-MUX 1-8

;R6-# младший байт кода 32,34-46

;R7-# старший байт кода 33,35-47

;PSW.1 - программируемый бит: 0 - взвешивание, 1 - калибровка.

;скорость передачи 300 бод, - задаётся ТС1

ORG 00H;начальный адрес трансляции

SJMP BEGIN

ORG 03H;прерывание по INT0

JMP MAIN

ORG 23H;прерывание по последоват. порту

JMP UAPP

ORG 30H;программа инициализации

BEGIN:

MOV IE, #10010001B;разрешение прерываний по последоват.

;порту и по INT0

MOV IP, #00010000B;установка высшего приоритета у прерыв. по ПП

MOV TMOD, #00100000B;режим с перезагр TH1 в TL1

MOV TCON, #00000000B;обнуление битов управления ТС1

MOV PCON, #00000000B;работа МК без удвоения частоты

MOV SCON, #0101000B;настройка UAPP

MOV P0, #0FFH;настройка портов P0 и P2 на ввод

MOV P2, #0FFH

MOV TH1, #187;загрузка и запуск таймера

MOV TL1, #187

SETB TCON.6

MOV R5, #15;очистка смещений

MOV R1, #52

CALL CLEAR

CLR PSW.1;активация подпрограммы взвешивания

STOP:JMPSTOP ;зацикливание

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;Nucleus of the Program

MAIN:

MOV R5, #0

MOV R0, #32;начальный адрес ячейки памяти

;для получаемого кода

MOV R1, #52;начальный адрес смещений

MOV P1, #11111000B;0 - настройка на вывод

MOV P1, R5;выбор канала 1 (MUX).

RETM:MOV P0, #0FFH;прием текущего значения

MOV R6, P0;кода с АЦП. "1"-линия настроена

MOV P2, #0FFH;на ввод

MOV R7, P2;в R6-младший байт кода,R7-старший

JNB PSW.1, MAIN0

MOV A, R6;запись полученных кодов в область смещений

MOV @R1, A;младший байт

INC R1

MOV A, R7

MOV @R1, A;старший байт

JMP COMMON

MAIN0:

CLR PSW.7

MOV A, R6

SUBB A, @R1 ;коррекция OffSet для младшего байта

MOV @R0, A;запись младшего байта

INC R0;вычисление адреса старшего байта

INC R1

MOV A, R7

SUBB A, @R1 ;коррекция OffSet для старшего байта

MOV @R0, A ;запись старшего байта

COMMON:

;подготовка к следующему циклу

INC R0;вычисление нового адреса младшего байта

INC R1

INC R5;вычисление адреса канала MUX

CLR PSW.7

MOV A, R5;проверка -> адрес канала > Last(8).

SUBB A, #8

JNZ GoOn

JB PSW.1, KALIB

ACALL OUT;вызов подпрограммы вывода инф. на ЭВМ

;и реинициализации управл. регистров.

KALIB:

CLR PSW.1

RETI

GoOn:

MOV P1, #11111000B;0 - настройка на вывод

MOV P1, R5;0 - 2 линии порта P1 - управляют MUX

MOV R4,#20 ;небольшая задержка для АЦП

DAC1:DJNZ R4, DAC1

JMP RETM

;End Nucleus of the Program

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;Вывод полученного веса в компьютер через последовательный порт

OUT:

MOV R6, #0;временно - R6-мл. результат

MOV R7, #0 ;R7-ст. результат

;R5-счетчик

MOV R5, #8

MOV R0, #32

SUMM:

MOV A, @R0;суммирование младшего байта

ADD A, R6;и запись его в R6

MOV R6, A

INC R0

MOV A, @R0;суммирование старшего байта

ADDC A, R7;и запись его в R7

CLR PSW.7

MOV R7, A

INC R0

DJNZ R5, SUMM;в регR6-мл. результат

; R7-ст. результат

CLR IE.7;запрет всех прерываний на время передачи

MOV SBUF,R6;передача младшего байта результата

ACALL DELAY;вызов подпрограммы задержки

MOV SBUF,R7;передача старшего байта результата

ACALL DELAY;вызов подпрограммы задержки

CLR SCON.1;сброс флага прерывания по посл. порта

SETB IE.7;разрешение прерываний

MOV R5, #0

MOV R0, #32;начальный адрес ячейки памяти

;для получаемого кода

MOV R1, #52;начальный адрес смещений

CLR IE.0;запрещение прерывания по INT0

RET

;Конец передачи

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;Прерывание по последовательному порту

UAPP:

CLR SCON.1;сброс флага прерывания

MOV R7, SBUF;прием управляющего слова

ACALL DELAY

MOV A, #00111100B;обнуление смещений и результатов

SUBB A, R7

JNZ NEXT0

MOV R5, #36;запуск очистки

MOV R1, #32

ACALL CLEAR

NEXT0:

MOV A, #11110000B;запрос на запуск преобразования и

SUBB A, R7;и на передачу в компьютер

JNZ NEXT1

;активация работы подпрограммы

CLR PSW.1;взвешивания в MAIN

SETB IE.0;разрешение прерывания по INT0

NEXT1:

MOV A, #00001111B;запрос на передачу в компьютер

SUBB A, R7

JNZ NEXT2

ACALL OUT

NEXT2:

MOV A, #0FFH ;запрос на калибровку

SUBB A, R7

JNZ NEXT3;активация работы подпрограммы

SETB PSW.1;калибровки в MAIN

NEXT3:

OTHERWISE:

RETI

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;Подпрограмма задержки для последовательного порта

DELAY:

MOV R2,#25;задержка, небходимая для

A2:MOV R3,#250;принятия/передачи байта

A1:DJNZ R3,A1

DJNZ R2,A2

RET

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CLEAR:

MOV @R1, #0;R5 - колличество

INC R1 ;R1 - начальный адрес

DJNZ R5, CLEAR

RET

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

END;конец трансляции