

Gh

B

Bg

Mlbh

Bfg

Brg

СОДЕРЖАНИЕ

Введение	4
1 Анализ современного состояния разработок по комплексам измерения диаграмм направленностей антенн	6
2 Выбор и обоснование принципов построения комплекса	Ошибка! Закладка не определена.
2.1 Требования к устройству	15
2.2 Выбор элементной базы модуля контроллера	17
2.3 Выбор метода управления шаговыми двигателями	17
2.4 Выбор схемы формирователя для счётчика	18
2.5 Обоснование схемы приёмника	20
2.6 Выбор принципов связи с ПК	21
2.7 Требования к программному обеспечению ПК	23
2.8 Набор команд микроконтроллерного модуля	24
2.9 Обоснование выбора типа операционной системы микроконтроллера	25
2.10 Основные функции программы микроконтроллера	27
2.11 Исследование методов организации полигона для измерений	32
3 Расчет и проверка электрических схем	40
3.1 Расчёт ключей управляющих шаговыми двигателями	40
3.2 Расчёт схемы приёмника	41
3.2.1 Расчёт программно – управляемого логарифмирующего измерительного приёмника	41
3.2.2 Расчёт аппаратного логарифмирующего преобразователя приёмника с кусочно-линейной аппроксимацией	49
Вывод	58
Перечень ссылок	59
Приложение А - программа микроконтроллера на языке ассемблера	60
Приложение Б – макроопределения для программы микроконтроллера на языке ассемблера	77
Приложение В – расчет цепи фильтра компаратора датчика положения платформы в MathCad 2001	80
Приложение Г – программа контроллера приёмника	84
Приложение Д – пример отладочной программы на языке TurboPascal 7.0	88
Приложение Е – схема модуля контроллера в программе OrCad 9.1	97
Приложение Ж - перечень элементов	98

ВВЕДЕНИЕ

Сегодня автоматизированные контрольно - измерительные стенды приобретают всё большую популярность. Причиной тому является развитие относительно нового направления в электронике – программируемых микроконтроллеров. Каждая микросхема контроллера представляет законченную ЭВМ, для работы которой требуется только подать питание. Особо заметный рывок вперёд в микроконтроллерной технике был сделан совсем недавно – около трёх лет назад, когда серийные мощные контроллеры стали производиться с перезаписываемой Flash- памятью. До этого память программ подавляющего числа микросхем программировалась однократно, а контроллеры с перезаписываемой памятью стоили значительно дороже. Поэтому контроллеры практически не применялись для мелкосерийных и несерийных стендов, так как отладка каждого стенда требует поменять память программ порядка сорока раз, что означало испортить сорок контроллеров. Следующим заманчивым достоинством доступных крупносерийных контроллеров стало интегрирование в них АЦП и UART – терминала. Благодаря этому на одной микросхеме теперь можно сделать полноценный измерительный комплекс. Фактически разработке модуля контроллера такого комплекса и посвящён этот проект. В этом измерительном комплексе почти все датчики находятся внутри одного контроллера, остальная схема представляет собой преобразователи-формирователи к виду, в котором контроллер может измерить нужные величины. Разрабатываемый модуль будет «сердцем» системы измерения диаграммы направленности антенн.

Необходимость в автоматизации измерения диаграммы направленности антенны вытекает из значительной длительности и значительного числа отсчётов, получаемых в результате эксперимента, особенно если диаграмму направленности надо измерить не в одной плоскости, а в секторе, да ещё на нескольких частотах. Поэтому сегодня в комплексах данного вида применяются методы, позволяющие значительно сократить время измерения при снятии диаграмм в широкой полосе частот, в частности исследование антенн при подаче сверхкоротких электромагнитных импульсов позволяет строить диаграмму сразу во всей полосе частот. Однако такие комплексы требуют дорогостоящей аппаратуры высокого класса, что не всегда оправдано. Разрабатываемый комплекс использует более традиционные, а значит и более дешёвые методы. Кроме того, используемые методы, в силу своей древности, хорошо согласуются со стандартами измерений подавляющего количества стран.

Итак, разрабатываемый комплекс должен уметь : снять диаграмму направленности антенны в заданном секторе при любом виде поляризации волны, в заданном диапазоне частот. Из этого следует, что комплекс должен обладать набором электронно-коммутируемых облучателей, автоматическим поворотным устройством, набором датчиков положения антенны и уровня сигнала, персональным компьютером для обработки данных, связанных по шине с комплексом и модулем управления комплексом. Теперь можно сформулировать цель данной работы – проектирование модуля микроконтроллера, обеспечивающего преобразование сигнала с датчика положения в цифровую величину, преобразование сигнала с датчика уровня принимаемого сигнала в цифровую величину, обеспечение надёжной связи с компьютером. Дополнительным требованием является достижение универсальности комплекса в плане количества устройств, входящих в комплекс. Так комплекс должен работать с любым генератором, с любым типом облучателя, любым персональным компьютером.

1 АНАЛИЗ СОВРЕМЕННОГО СОСТОЯНИЯ РАЗРАБОТОК ПО КОМПЛЕКСАМ ИЗМЕРЕНИЯ ДИАГРАММ НАПРАВЛЕННОСТЕЙ АНТЕНН

Приведём несколько примеров выдающихся аналоговых комплексов. Прежде всего, хотелось бы отметить комплекс НТП «Тензор»[1]. Измеряемая антенна A_2 расположена на поворотном устройстве управляемом ПЭВМ. Сигнал для стабилизации временного положения с разветвителя подается на первый канал стробоскопического преобразователя. Структурная схема измерителя ДН антенн приведена на рисунке 1.1.

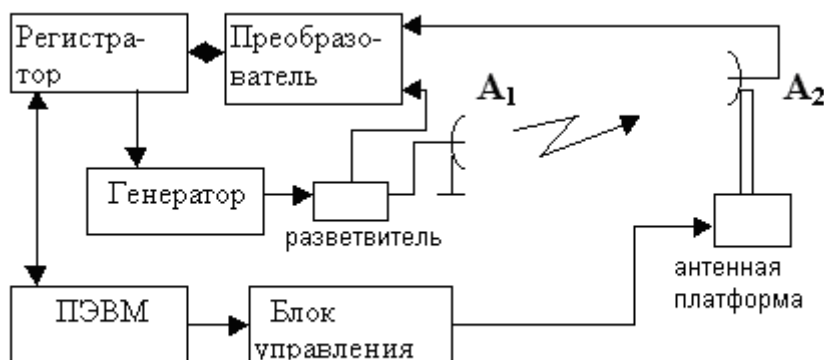


Рисунок 1.1 – Структурная схема измерителя ДН антенн НТП «Тензор».

Особенности метода:

- возможность работы в обычных помещениях с динамическим диапазоном 40 дБ;
- измерения в ближней и дальней зоне;
- высокая производительность;
- сверхширокий частотный диапазон работы: от 100 МГц до 37 ГГц.
- измерения диаграмм направленности, коэффициентов усиления и КСВН

Исследование антенных систем традиционно проводится в частотной области с использованием гармонического сигнала. Это требует значительных затрат времени и большого количества измерительного оборудования, особенно при исследовании широкополосных и сверхширокополосных антенн. В настоящее время развитие техники пикосекундных импульсов позволяет реализовать схему измерения параметров антенн во временной области в диапазоне частот до 18 ГГц.

Данный способ измерений основан на использовании зондирующего видеоимпульса длительностью несколько десятков пикосекунд для получения отклика исследуемой антенны во временной области. Частотная характеристика отклика рассчитывается с помощью алгоритмов БПФ, затем полученный спектр нормируется на спектр сигнала от эталонной антенны с известной зависимостью коэффициента усиления от частоты. Вращая антенну в различных плоскостях, получают амплитудную и фазовую диаграммы направленности антенны.

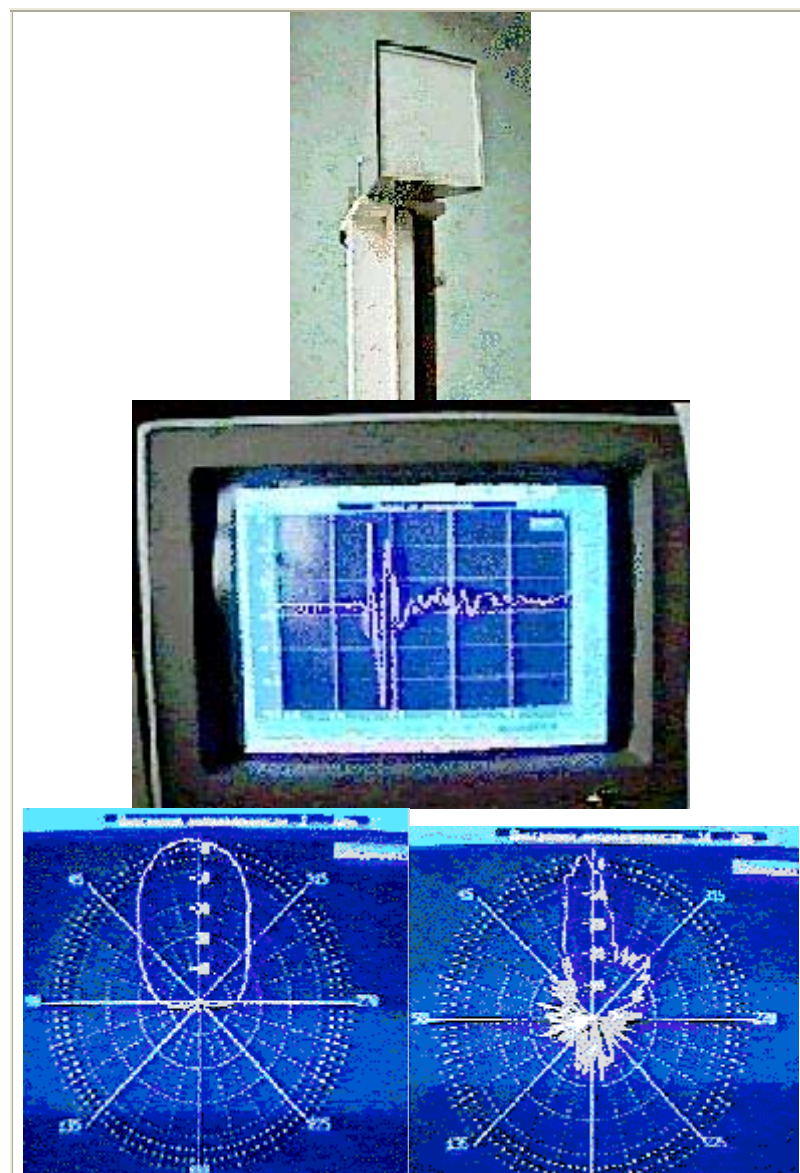


Рисунок 1.2 - Измеряемая рупорная антенна, ее импульсный отклик и диаграммы направленности на частотах 2 и 15 ГГц.

Достоинством метода является возможность одновременного измерения параметров антенн во всем диапазоне частот и временная селекция сигналов, позволяющая обходиться без дорогостоящих безэховых камер. Регистрация данных проводится в промежутке времени (временном окне) между появлением сигнала, распространяющегося по прямой, соединяющей антенны, и появлением отраженных сигналов.

Выбор расстояния между антеннами и расположения отражающих поверхностей относительно излучающей и приемной антенн определяется частотным диапазоном испытываемой антенны f_l или f или f_h , размером апертуры D и фазовой характеристикой $\varphi(\omega)$ тракта приемопередачи. Разность хода прямого и отраженного лучей

$$\Delta R = \sqrt{R^2 + 4H^2} - R,$$

где R – расстояние между антеннами; H – высота расположения антенн над отражающей поверхностью.

Безэховый временной интервал

$$\tau = \frac{\Delta R}{c},$$

где c – скорость света

В случае линейности фазовой характеристики $\varphi(\omega) = \omega \tau$, фазовое «расплывание» импульса отсутствует и время релаксации импульса на выходе антенны

$$\tau_r = \max\{1/f_l, 1/(f_h - f_l)\},$$

Необходимое условие однозначности измерений

$$T \geq \tau_r.$$

Для измерений в дальней зоне

$$R \geq 2 D^2 f_h / c.$$

Тогда минимальное удаление антенны от отражающих плоскостей

$$H_{\min} = \frac{1}{2} \sqrt{(c\tau_r)^2 + 4D^2 f_h \tau_r}.$$

Таким образом, вид импульсной характеристики антенны, ее размеры и диапазон рабочих частот накладывают определенные ограничения на возможность безэхового измерения ее характеристик, однако в этом случае возможно проведение измерений в безэховой камере с сохранением других преимуществ метода – дешевизны измерительного оборудования и оперативности измерений. Так, на снятие диаграммы направленности сверхширокополосной антенны в диапазоне частот 1 – 18 ГГц требуется около 10 мин.

Пример. Время распространения импульса в широкополосной рупорной антенне с полосой частот 1-18 GHz $\tau_r = 1,5$ ns, размер апертуры антенны $D = 0,24$ м. Для этих значений можно получить для частоты $f = 18$ GHz $H_{\min} = 1,27$ м. при расстоянии между антеннами $R = 6,9$ м.

Таким образом, измерения могут быть выполнены в обычной комнате. Длительный импульсный отклик антенны и узкий диапазон ее рабочих частот могут ограничить возможность проведения измерений в обычной комнате. Однако, измерения в безэховой комнате сохраняют другие преимущества метода - низкую цену оборудования,

чрезвычайно низкую среднюю мощность излучения и высокую скорость измерений. Так, измерение диаграммы направленности рупора во всем диапазоне частот требует приблизительно 10 мин времени. Несколько диаграмм рупорной антенны приведены на рисунке 1.3.

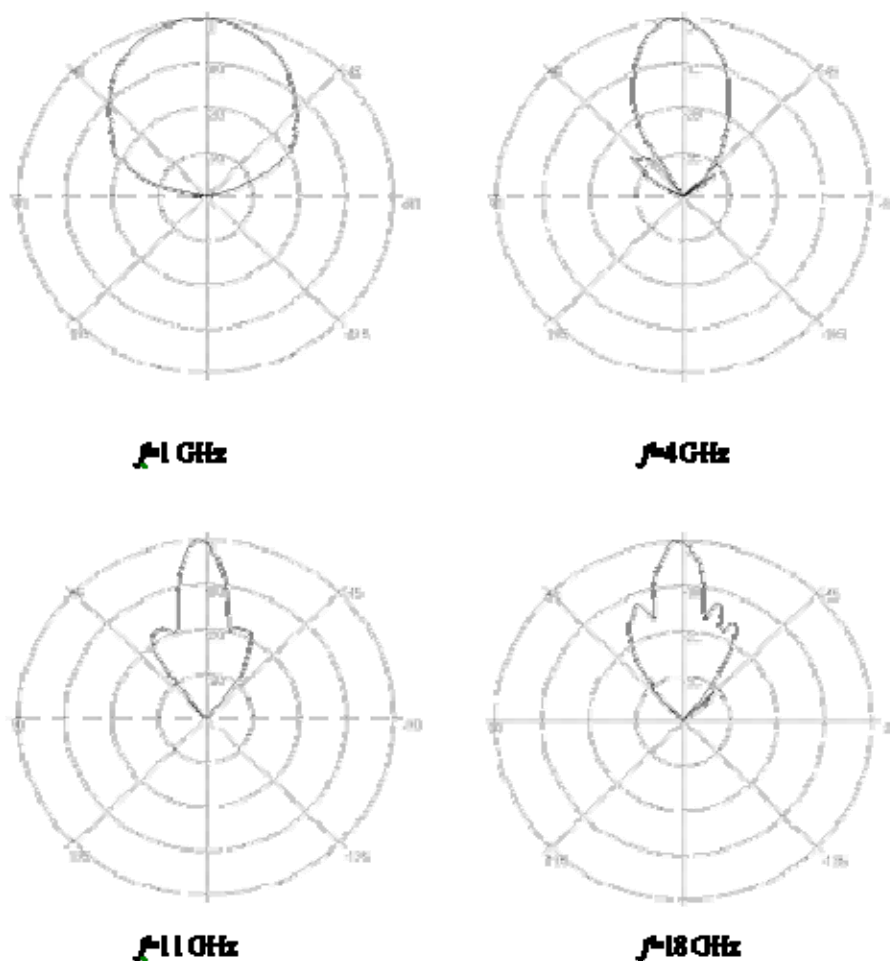


Рисунок 1.3 – Диаграммы направленности антенн на различных частотах.

В состав измерительной установки входят:

- генератор видеоимпульсов;
- регистратор сигналов;
- стробоскопический преобразователь;
- излучающая антенна;
- эталонная антенна;
- треноги;
- координатно-поворотное устройство;
- персональный компьютер;
- специализированное программное обеспечение.

Генератор обеспечивает формирование видеоимпульсов длительностью около 250 пс.

Импульс генератора обрабатывается формирователем для получения импульса длительностью около 30 пс и эффективной шириной спектра 0 – 18 ГГц.

В качестве регистратора используется стробоскопический осциллограф, управляемый от компьютера.

Основные характеристики установки:

- Диапазон частот измерения эффективной поверхности и коэффициента усиления антенн от 0,2 до 18 ГГц.
- Диапазон измерения коэффициента усиления относительно коэффициента усиления эталонной антенны не менее 30 дБ.
- Пределы измерения ослабления тракта приема-передачи в диапазоне частот от 1 до 18 ГГц от 0 до 40 дБ.
- Предел основной погрешности измерения коэффициента усиления и эффективной поверхности антенн ± 1 дБ
- Динамический диапазон измерения диаграммы направленности и поляризации 40 дБ.
- Диапазон измерения диаграммы направленности по азимуту $\pm 180^\circ$.
- Диапазон рабочих температур $+10 - +40^\circ\text{C}$.

Опорно - поворотное устройство (ОПУ) с диэлектрическим столом для измерения параметров антенн, изображённое на рисунках 1.4, 1.5, имеет следующие технические характеристики:

- Диапазон углов поворота в азимутальной плоскости: $\pm 200^\circ$
- Погрешность установки угла: $0,2^\circ$
- Грузоподъемность: 50 кг
- Высота мачты: до 3,5 м
- Масса 90 кг

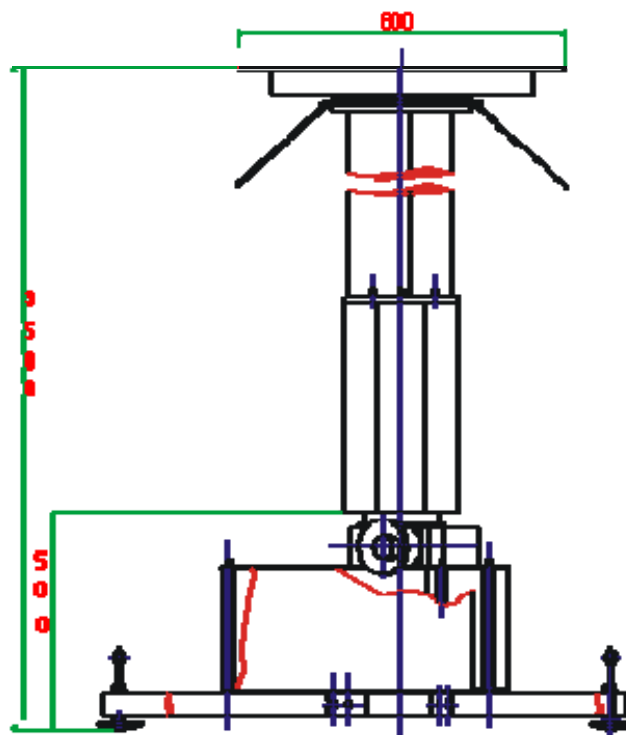


Рисунок 1.4 - Габаритный чертеж ОПУ с мачтой.

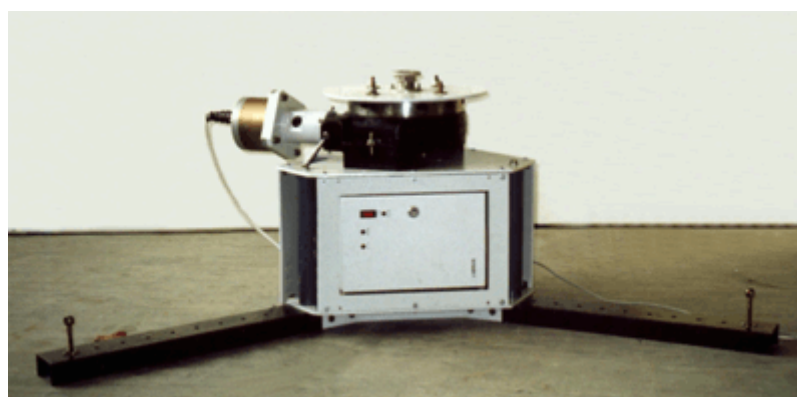


Рисунок 1.5 - Внешний вид ОПУ.

Следующий достойный рассмотрения метод предложил Эл Брай. Устройство включает в себя вспомогательную антенну, которая выносится на определенное расстояние, диодный детектор и соединительную линию, ведущую к регулируемому индикатору, который находится на рабочем месте оператора. Точность измерений зависит от достоверного определения выходной мощности и стабильной работы устройства, что вполне достижимо. Этот метод предполагает наличие регулировки и измерения выходной мощности передатчика и вращающейся антенны с калиброванным индикатором направления излучения. Вспомогательная антенна представляет собой полуволновой диполь для диапазона на котором проводятся измерения. Лучше всего применять диполь в форме INV.VEE, в этом случае достаточной одной опоры, чтобы поднять антенну на достаточную высоту. Устройство показало хорошие результаты на диапазонах 28, 21, 14 и 7 мегагерц, когда расстояние до испытуемой антенны составляло приблизительно 3 длины

волны. Диполь должен быть повернут (перпендикулярен) по направлению к основной антенне. Испытуемая антенна направляется на дополнительную антенну и затем постепенно увеличивается мощность передатчика, на приборе появляются определенные показания. Затем, исходя из показаний прибора М1, антенну направляют точно на дополнительную антенну. У автора этой разработки направление определилось в 105 градусов. Затем прибор настраивается таким образом, чтобы при выходной мощности менее чем 25 Вт прибор М1 давал максимальные показания.

Запаса регулировки чувствительности должно хватать для максимальных показаний М1 при максимальной мощности (у автора — более 500 Ватт). Необходимо заметить, что передатчик должен включаться на время, необходимое для измерений и, конечно же на свободной частоте.

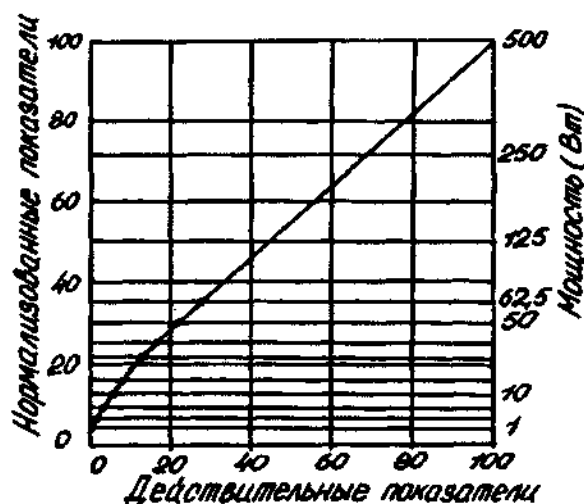


Рисунок 1.6 – Корреляционная кривая.

Для того чтобы избежать неточностей в измерениях, вызванных нелинейностью диода и другими дефектами системы, применяется способ, который называется системой характеристики.

Для этого составляется таблица и используется корреляционная кривая (рисунок 1.6) для компенсации нелинейности системы. Таблица имеет три вертикальных колонки где отмечается мощность (Вт), показания прибора М1 и нормализованные показания. Когда антенна направлена на дополнительную антенну и мощность передатчика максимальна (у автора 500 Вт) потенциометр Р1 устанавливается так, чтобы на приборе были максимальные показания (в случае автора, его 50 мА прибор имел шкалу 1—100 т. е. прибор показывал — 100). Затем мощность передатчика уменьшается на определенную величину и во вторую колонку заносятся показания прибора, соответствующие выходной мощности передатчика.

Последнее измерение должно быть проделано на максимально возможном уровне мощности, который бы соответствовал показанию прибора близкому к нулю.

Направление антенны все это время остается неизменным. После заполнения первых двух колонок приступают к вычислению нормализованных показаний. Вычисления производятся при помощи следующей формулы:

$$НП = \frac{100}{\sqrt{\frac{500}{P}}} ;$$

Где НП — нормализованные показания;

100 — максимальные показания прибора;

500 — максимальная мощность;

P — мощность, используемая для соответствующего показания прибора

Затем готовят график, где показано соотношение между показаниями в колонке 2 и нормализованными показаниями в колонке 3. График поможет избежать дополнительных пересчетов для дальнейших испытаний.

Перед тем, как произвести измерение диаграммы направленности антенны рекомендуется подготовить вторую таблицу для внесения данных измерений.

В таблице должно быть шесть колонок. Первая колонка предназначена для нормализованной схемы диаграммы направленности, которая начинается с 0/360 градусов с интервалами 15 градусов, а вторая колонка для фактического направления антенны, которая начинается (как у автора) со 105 градусов на вершине схемы.

Каждая дополнительная пара колонок используется для записи измерений.

Затем начинаются сами измерения. Перед каждым измерением должна проверяться выходная мощность, которая должна оставаться неизменной в течение всех испытаний.

Направление антенны остается неизменным (т. е. 105 градусов, как у автора) и прибор выставлен на максимальные показания т. е. 100.

Затем антенна меняет свое направление по кругу, через каждые 15 градусов, как указано в колонке 2 и соответствующие показания записываются в колонке 3.

После проведения всех 24 измерений можно записать в следующей колонке нормализованные данные в колонку 4, используя график на рисунке 1.6. Две оставшиеся колонки можно использовать для измерения диаграммы направленности на другой частоте, скажем в SSB-участке, если первые измерения проводились на нижнем конце диапазона. После этого можно составить действительную диаграмму направленности антенны.

В этой главе были приведены наиболее интересные методы измерения диаграммы направленности антенн, первый из них максимально экономит время и средства, второй – даёт максимальную точность измерения. Применительно к имеющимся средствам можно отметить, что первый метод технически почти неосуществим, а вот применение второго метода возможно, но только с помощью специального СВЧ генератора, уровень мощности на выходе которого контролируется с компьютера. Поэтому для расширения универсальности модуля необходимо иметь возможность снимать диаграмму направленности при постоянном уровне мощности генератора. Следовательно разрабатываемый модуль будет нацелен на снятие ДН антенны классическим методом вращающейся исследуемой антенны при неподвижной эталонной передающей антенне с постоянным уровнем мощности излучения. При этом модуль будет полностью совместим с методом вращающейся исследуемой антенны при неподвижной эталонной передающей антенне с переменным уровнем мощности излучения.

2 РАЗРАБОТКА ПРИНЦИПОВ ПОСТРОЕНИЯ КОМПЛЕКСА

2.1 Разработка структурной схемы

Здесь надо заметить, что автор данной работы не является разработчиком всего комплекса в целом. Целью работы является не выбор принципов функционирования всего устройства, а использование имеющихся ресурсов с максимальной эффективностью.

Приступаем к проектированию структурной схемы модуля контроллера, уточним задачу.

Микроконтроллерный блок Антенного измерительного комплекса должен обеспечивать:

- связь с персональным компьютером,
- управление генератором СВЧ,
- управление двумя шаговыми двигателями (далее ШД) комплекса,
- управление поляризацией ЭМВ,
- преобразование цифрового сигнала с датчиков угла и азимута в число, соответствующее углу поворота,
- преобразование аналогового уровня напряжения с детекторного избирательного усилителя в число.

В соответствии с этим блок должен иметь следующие аппаратные средства:

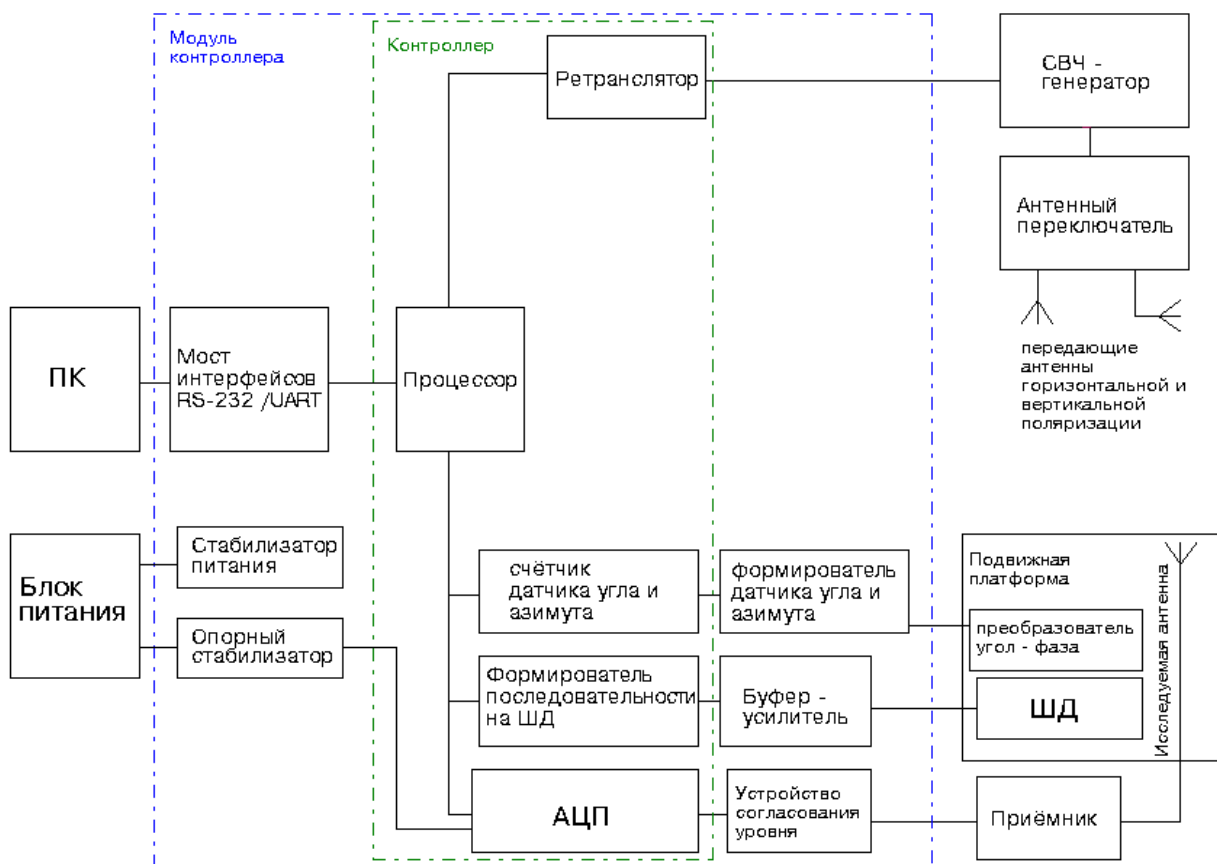
- мост коммуникационных интерфейсов RS232-UART,
- «ретранслятор» цифрового кода на генератор СВЧ,
- формирователь последовательности напряжений на ШД,
- мощные цифровые ключи для коммутации ШД,
- формирователь и счётчик для измерения положения,
- возможно нелинейный усилитель – преобразователь уровня для расширения диапазона детектора,
- АЦП.

Эти функции в значительной части реализуются на контроллере ATmega8535, который имеет в своём составе следующую используемую периферию:

- UART терминал,
- 4 8-битных порта,
- 2 аппаратных счётчика,
- АЦП.

Мост интерфейсов можно выполнить на ИМС MAX232, которая является специализированным мостом интерфейсов с минимальным числом внешних компонентов.

Усилитель на ключи можно выполнить на микросхеме- буфере 74НС245, которая кроме функции усиления-развязки нагрузки и контроллера производит защиту от одновременного включения обоих двигателей, чтобы не перегрузить блок питания. Структурная схема комплекса приведена на рисунке 2.1.



Принятые сокращения: ПК – персональный компьютер, СВЧ – сверх высокочастотный, АЦП – аналого-цифровой преобразователь, ШД – шаговые двигатели.

Рисунок 2.1- Структурная схема комплекса и модуля контроллера.

Pin- диодный мост позволяет управлять поляризацией ЭМВ: диоды коммутируют напряжение генератора или на антенну горизонтальной поляризации, или на антенну вертикальной поляризации (эти антенны неподвижны). Принятый исследуемой антенной сигнал детектируется СВЧ детектором, фильтруется и оцифровывается. Выбранная структурная схема соответствует методу «вышки». В данном случае исследуемая антенна является приёмной, однако можно поставить её в качестве передающей. Такой метод используется реже, поскольку в таком случае тяжело предотвратить попадание отражённого луча на измерительный зонд

2.2 Выбор элементной базы модуля контроллера

Для реализации поставленных задач выбираем современный контроллер ATmega8535 корпорации ATMEL [2]. Он имеет полный набор необходимой периферии. Выбираем тактовую частоту 4 МГц поскольку это позволит производить пошивку МК, не доставая с платы. Корпус выбираем типа DIP – с ним удобнее работать. Такая конфигурация потенциально позволит производить:

- оценку положения подвижной части комплекса с точностью
$$dFi = Fd/Ft = 400 / 4\,000\,000 = 0.01 \%,$$
где $Fd = 400$ Гц-частота на датчиках ,
$$Ft = 4\,000\,000 \text{ Гц} - \text{тактовая частота контроллера};$$
- измерение напряжения с датчика мощности сигнала с точностью
$$dU = 1/R = 1/1024 = 0.1 \%,$$
где R- разрядность АЦП;
- производить обмен данными с ПК на скорости до 19200 Бод, поскольку при больших скоростях вероятность ошибки становится больше 1%;
- производить on-board обработку до 4 млн. ассемблерных команд в сек;
- хранить переменные в 512 байтах ОЗУ;
- хранить до 256 байт в ПЗУ данных;
- производить загрузку до 8 кБ кода команд.

Кроме контроллера на модуль установим ИМС трёх типов: буфер 74НС245, «мост интерфейсов RS232-UART» MAX232 [3], прецизионный стабилизатор MAX 195 для стабилизации опорного напряжения АЦП. Все микросхемы включим по схемам, рекомендованным производителем. Теперь рассмотрим элементы, требующие особого внимания: ключи, формирователи с датчика положения и с датчика уровня сигнала.

2.3 Выбор метода управления шаговыми двигателями

Задача следующая: обеспечить работу в ключевом режиме при работе на индуктивную нагрузку, ограничивать ток не требуется, реактивная энергия во время выключения рассеивается на сопротивлении мотора. Напряжение 25 В, ток 4 А. Потребление от контроллера – не более 5 мА. Продолжительность импульса – 80 мс, скважность – 4. Такая продолжительность импульсов была выбрана экспериментально, так как меньшая продолжительность импульсов вызывает резонансные явления в конструкции устройства. Решение : поскольку катушки мотора имеют значительную индуктивность и сопротивление порядка 10 Ом, необходимо защитить ключи. Это лучше всего

осуществить с помощью диодов, включённых параллельно двигателям. Электрический расчет схемы приведён в третьем разделе.

2.4 Выбор схемы формирователя для счётчика

Это устройство особо ответственное – сигнал на выходе формирователя должен будет измеряться с точностью 0,05%. Поэтому возьмём обычную схему усилителя – ограничителя на операционном усилителе и добавим к ней фильтры. Лучше всего в такой схеме работают ОУ с полевыми транзисторами на входе, такие как MCP 601 [2]

Принципиальная схема приведена на рисунке 2.2.

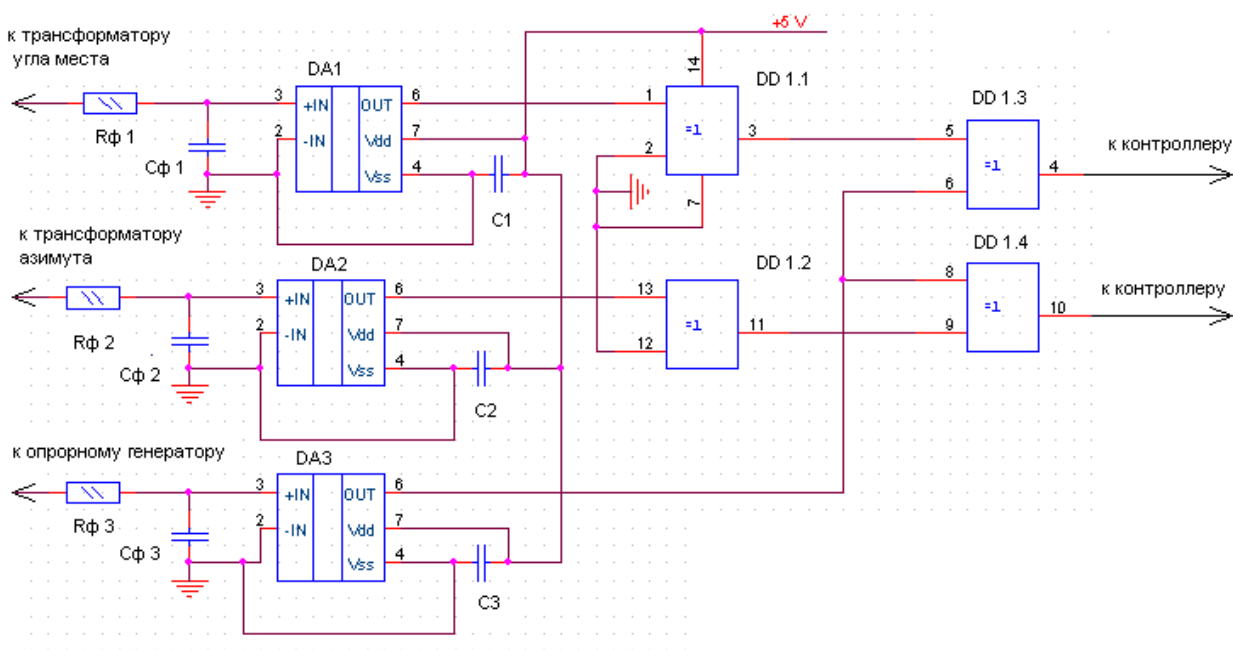


Рисунок 2.2 – Формирователь датчика положения антенны.

Расчет параметров фильтра $R_{\phi}C_{\phi}$ этого усилителя в MathCad приведён в приложении В. Конденсаторы $C_{\phi 1} - C_{\phi 3}$ должны быть высокостабильные, например с полиэтилен-сульфидной изоляцией. Зависимость отклонения ёмкости от температуры и частоты представлены на рисунках 2.3 и 2.4. Такие конденсаторы выпускает фирма «Evox» [3].

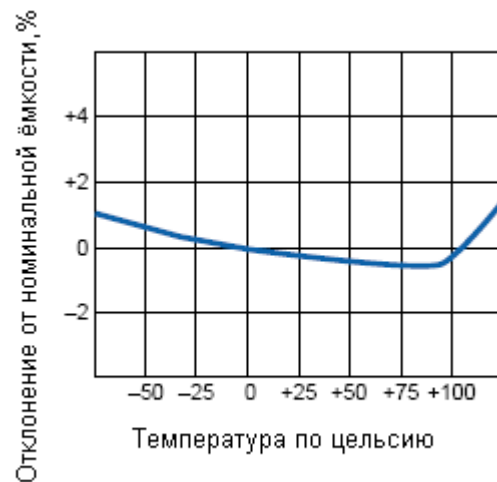


Рисунок 2.3 - Отклонения ёмкости от температуры.

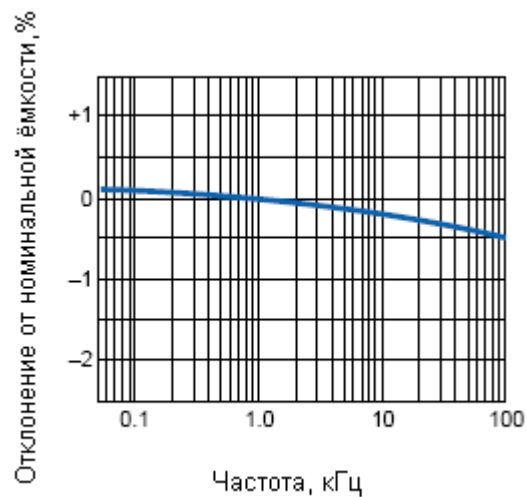


Рисунок 2.4 - Отклонения ёмкости от частоты.

Конденсаторы С1-С3 – керамические, 1 мкФ (в соответствии с рекомендацией производителя операционных усилителей), микросхема DD1 – CD4070 – четыре логических элемента «исключающее или». Операционные усилители МСР601 – с полевыми транзисторами на входе и возможностью работы при отрицательных входных напряжениях.

Выбор такого датчика обусловлен высокой линейностью получаемых результатов. экспериментальная зависимость количества отсчётов датчика от угла поворота платформы представлен на рисунке 2.5.



Рисунок 2.5 – Экспериментальная зависимость количества отсчётов датчика от угла поворота платформы .

В соответствии с представленным графиком, для определения положения требуется произвести поворот на несколько шагов и сравнить показания с начальными. Не очень удобно, но допустимо.

2.5 Обоснование схемы приёмника

Приёмник должен преобразовать СВЧ колебания сантиметрового диапазона модулированные по амплитуде сигналом 1 кГц. Приёмник строится по принципу прямого усиления. СВЧ колебания без усиления детектируются на СВЧ детекторе и затем огибающая усиливается активным полосовым фильтром настроенным на 1 кГц и детектируются. Дополнительная задача – обеспечение высокой линейности преобразования уровень входной мощности (дБ) – выходное квазипостоянное напряжение. Дополнительным требованием является лёгкость согласования с контроллером комплекса, возможность замены на другой приёмник, возможность работы отдельно от комплекса. Выбранная функциональная схема приёмника изображена на рисунке 2.6.

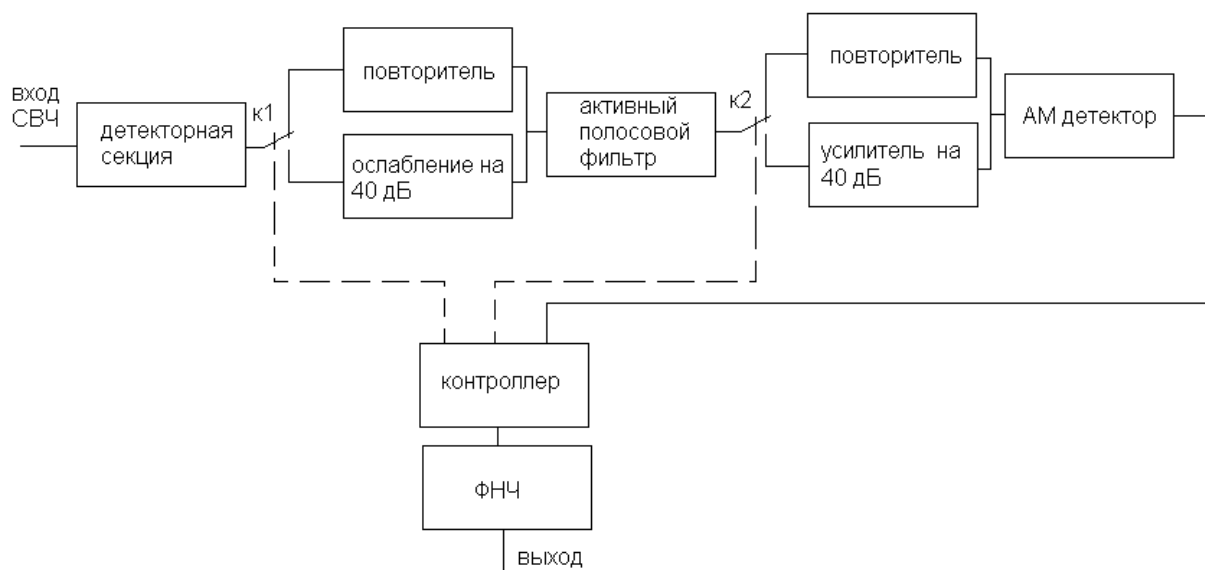


Рисунок 2.6 – функциональная схема приёмника.

2.6 **Выбор принципов связи с ПК**

Связь осуществляется через COM- port. Контроллер посылает ответ после приёма заданного количества байт, входящих в пакет, проверки и выполнения команд. Форма пакетов приведена в таблице 1 и 2.

Таблица 2.1 - Пакет от ПК:

Байт	0	1	2	3	4	5	6	7	8
Имя	A	K	P_nomber	Move_dir	Number_of_steps	Get_set	Data_H	Data_L	Con_sum

Байт 0,1: заголовок;

Байт 2: номер пакета;

Байт 3: направление движения комплекса(вверх/вниз,вправо/влево) (0 .. 4)

Байт 4: число шагов (1..255)

Байт 5:команда снять информацию с датчиков/ установить поляризацию/частоту

Байт 6,7: информационные байты

Байт 8: контрольная сумма байтов 0-7- они последовательно проходят через хог с переменной, начальное значение которой было 0xff.

Таблица 2.2 - Пакет от МК.

Байт	0	1	2	3	4	5	6	7	8	9	10	11	12
Имя	A	K	P_number	H1	L1	H2	L2	H3	L3	H4	L4	Err	C_sum

Байт 0,1: заголовок;

Байт 2: номер пакета;

Байт 3-10: старшие и младшие байты результата

- если используется усреднение $X = (256 * (H1 + \dots + H4) + (L1 + L2 + \dots + L4)) / 4$

где X- измеряемая величина

- если не используется усреднение $X = 256 * H1 + L1$, остальные H и L равны нулю.

Байт 11: =Err =254 если обнаружена ошибка, ОК = 0

Байт 12: контрольная сумма пакета, аналогично принимаемому пакету.

Такая система команд позволяет минимальное время занимать COM-port и при этом достаточно честно находить ошибки. Здесь используется проверка по контрольной сумме а не CRC для того, чтобы сэкономить память контроллера.

Работает связь следующим образом:

A – инициализация:

-посылается команда «проверка оборудования», компьютер ждёт 1 секунду.

Если ответ не пришёл, выводится сообщение «комплекс не подключён»,

если приходит пакет с сообщением о неисправности, выводится сообщения об этих неисправностях.

Б – работа:

основной командой, используемой при работе является : повернуться до десяти шагов и измерить напряжение с датчика, которое выполняется за время, практически равное времени движения, поэтому таймаут устанавливается на время $N * 80$ мс (N- число шагов) если ответ не пришёл, посылается ещё один (с тем же номером), после третьей попытки выдаётся сообщение о потере связи. Номер пакета нужен для того, чтобы ПК и МК выявляли не принятые и потерянные пакеты (в случае обнаружения ошибки в контрольной сумме производится повторный запрос (пакет с тем же номером от ПК- контроллер повторяет пакет, а при потере пакета производится повторная его отправка через время, равное таймауту).

В свою очередь контроллер производит следующую обработку пакета : во-первых – поиск заголовка, с которого начинается запись пакета в ОЗУ. Во-вторых -

проверка контрольной суммы принятого пакета, если она не верна, то ответ не отсылается. После окончания принятия пакета начинается проверка его команд, после чего выполняются команды move_dir, после этого выполняются команды Get_Set, затем формируется пакет, считается контрольная сумма и отправляется. Следует отметить, что проверка номера пакета осуществляется не только ПК, но и контроллером, для того, чтобы выявлять повторные пакеты и не изменять положения комплекса при этом.

2.7 **Требования к программному обеспечению ПК**

В диалоговом окошке необходимо предусмотреть следующие меню(стандартные):

- Файл:
 - новый
 - открыть
 - сохранить
 - печать
- Сервис:
 - Сохранить график в формате BMP?
- Помощь : (иметь возможность подключить текстовый файл).
- Настройки соединения:
 - COM x (1 по умолчанию)
 - Скорость →19200 Kbaud по умолчанию

В диалоговом окошке необходимо предусмотреть следующие меню(не стандартные):

- настройки
 - 1,Частота x xxx ,x MHz (2 500, 0 MHz по умолчанию)
 - 2,Поляризация
 - вертикальная (по умолчанию)
 - горизонтальная
 - 3,Сектор : измерения производить :
 - только в горизонтальной плоскости (по умолчанию)
 - угол по вертикали xx degrees(0 degrees по умолчанию)
 - диапазон по горизонтали xx .. xx degrees (-180 ..+180 по умолчанию)
 - только в вертикальной плоскости

угол по горизонтали xx degrees(0 degrees по
умолчанию)

диапазон по вертикали xx .. xx degrees
(-60 ..+90 по умолчанию)

- в секторе

диапазон по горизонтали xx .. xx degrees(предел-180 ..+180)

диапазон по вертикали xx .. xx degrees
(-60 ..+90 default)

4,Число повторения измерений: x(1- по умолчанию)

5,Сетка: {установка шага сетки

- по углу поворота, x degrees(0,4 degrees по умолчанию)

- по уровню сигнала, y dB(5 dB по умолчанию)

установка границ сетки

- по углу : x .. y degrees(360 degrees по умолчанию)

-по уровню : 0 .. x dB(60 dB по умолчанию)}

- нормировано /не нормировано к максимальному значению по
уровню(нормировано по умолчанию)

6,Система координат:

-декартовая(по умолчанию)

-полярная

7,Точность -низкая(по умолчанию)

-высокая(работа с усреднением)

- Результаты: (то что не поместилось в окне)

В диалоговом окошке необходимо поместить график с результатами

измерения(зависимость напряжения с датчика от угла/азимута поворота):

график должен быть максимальных размеров и строиться по ходу выполнения
измерений;

на графике должны определяться точка максимума по уровню; точки, отстоящие по
уровню от максимума на 3 dB: для этих точек должны выводиться соответствующие им
углы.

2.8 Набор команд микроконтроллерного модуля

В соответствии с п.4.2 команды делятся на команды движения комплекса(Move_dir) и
команды считывания датчиков и установок (Get_set).

команды Move_dir:

- | знач. | команда |
|-------|--------------|
| 0- | нет движения |
| 1- | STEP_UP |
| 2- | STEP_DOWN |
| 3- | STEP_LEFT |
| 4- | STEP_RIGHT |

команды Get_set:

- | знач. | команда |
|-------|--|
| 0- | нет команды |
| 1- | GET_U1- снимается уровень сигнала(без усреднения) |
| 2- | SET_FREQ- установка частоты генератора $F = (Data_L + 256 * Data_H) / 10$;MHz |
| 3- | SET_VERT_POL -установка вертикальной поляризации волны генератора. |
| 4- | SET_HOR_POL -установка горизонтальной поляризации волны генератора. |
| 5- | GET_UGOL -опросить датчик угла(без усреднения) |
| 6- | GET_AZIMUT - опросить датчик азимута(без усреднения) |
| 7- | EQ_TEST -проверить работоспособность датчиков азимута/угла (если не работают, Rpacket (ERR) = Err) |
| 8- | SET_LEVEL- установка уровня мощности генератора, в децибелах, $LEVEL = - DATA_L$, дБ. |
| 9- | GET_U1_U- снимается уровень сигнала(с усреднением) |
| 10- | GET_UGOL_U- опросить датчик угла(с усреднением) |
| 11- | GET_AZIMUT_U- опросить датчик азимута(с усреднением) |
- усреднение производится в течении 15 msec, снимается 4 отсчёта через каждые 5 msec, таким образом исключается наводка 50Гц.

2.9 **Обоснование выбора типа операционной системы микроконтроллера**

Операционная система контроллера является ОС типа Foreground/Background. Это означает, что контроллер находится в обработке вечного цикла кода (Background), который изредка прерывается прерываниями. Это было определено тем, что

контроллер комплекса занят выполнением простейших функций, которые всегда успешно завершаются без заикливания и всяких там «мёртвых хваток», контроллер общается исключительно с ПК, и никакой многозадачности от него не требуется. Поэтому такая ОС позволяет сэкономить время и память контроллера и значительно упростить систему, что немаловажно для её надёжной работы.

Работает МК следующим образом : по включению питания происходит инициализация – определяется адрес начала стека, далее идёт назначение портов на вход/выход, после чего инициализируется UART и АЦП. Далее начинается тот самый вечный цикл, естественно с команды «принять пакет», которая занимается вопросами приёма/обработки пакета от ПК.

Далее принятый пакет обрабатывается – производится выполнение команд `move_dir`, после которой выполняется команда `get_set`. Стоит отметить, что в этих функциях также производится проверка принятого пакета на правильность, а функции работают напрямую с регистрами отправляемого пакета, записывая в него данные. После обработки принятого пакета вызывается функция передачи, которая считает контрольную сумму и отправляет пакет. Теперь цикл работы закончен – поэтому начинается следующий. Блок – схема алгоритма работы контроллера приведена на рисунке 2.7. Программа контроллера состоит из 3 частей – собственно программы контроллера (в приложении А), макроопределений комплекса (приведены в приложении Б) и стандартных макроопределений ресурсов контроллера.

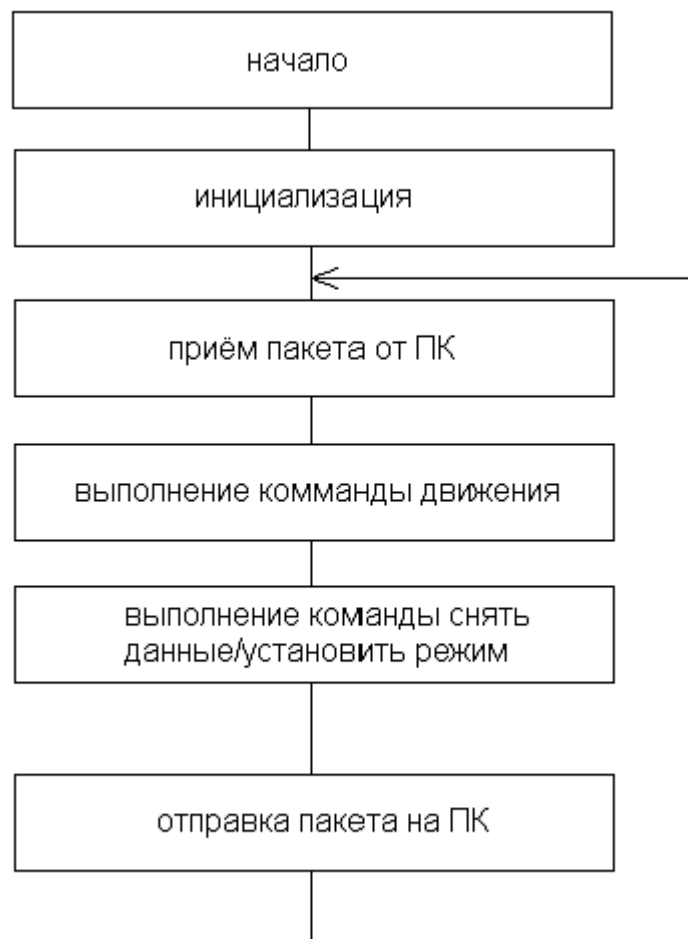


Рисунок 2.7 – блок – схема алгоритма работы контроллера.

2.10 Основные функции программы микроконтроллера

2.10.1 Функция приёма пакета от ПК

Блок – схема алгоритма приведена на рисунке 2.8. В задачу функции входят всё, что связано с приёмом пакета : поиск заголовка, приём и сохранение данных в ОЗУ, проверка контрольной суммы. Если ПК посылает неправильный заголовок или контрольную сумму, работа функции не прекращается никогда, и узнать, подключён комплекс или нет не представляется возможным.

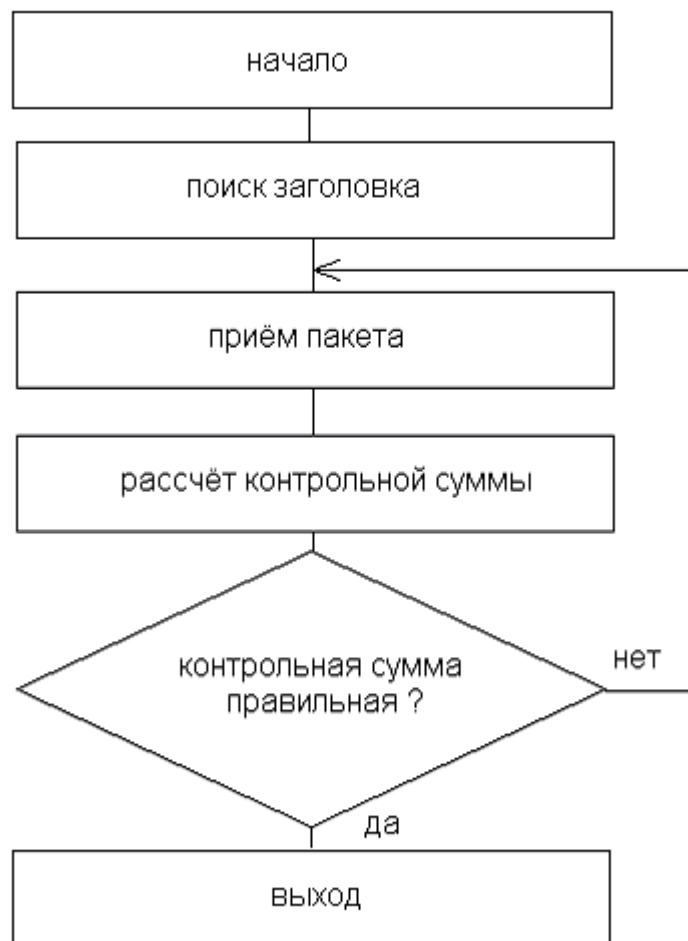


Рисунок 2.8 - блок – схема алгоритма функции приёма.

2.10.2 Функция передачи пакета на ПК

В задачу функции передачи данных на ПК входит : определение заголовка, расчет контрольной суммы, отправка пакета. Пакет формируется в ОЗУ во время выполнения команд снять данные/ установить режим. Блок – схема алгоритма приведена на рисунке 2.9

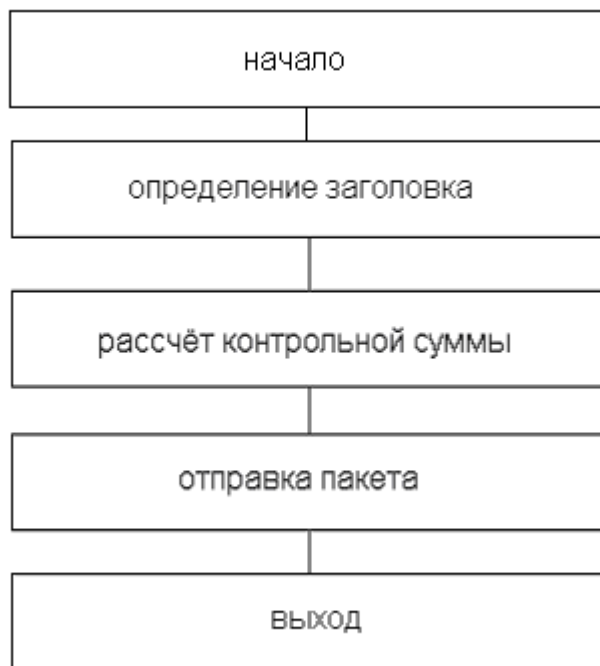


Рисунок 2.9 - блок – схема алгоритма функции передачи.

2.10.3 Функция определения положения платформы

Задачей этой функции является определение положения платформы с высокой точностью.

Основной проблемой, связанной с измерением является малая помехозащищённость формирователя, в результате чего на выходе формирователя появляется сигнал, похожий на дребезг контактов. Поэтому работа без программной фильтрации невозможна. блок – схема алгоритма функции определения положения приведена на рисунке 2.10. На выходе формирователя появляется сигнал с широтно - импульсной модуляцией, длительность сигнала линейно связана с углом поворота. В данной системе измеряется длительность сигнала логической «1», что и отражено на блок – схеме.

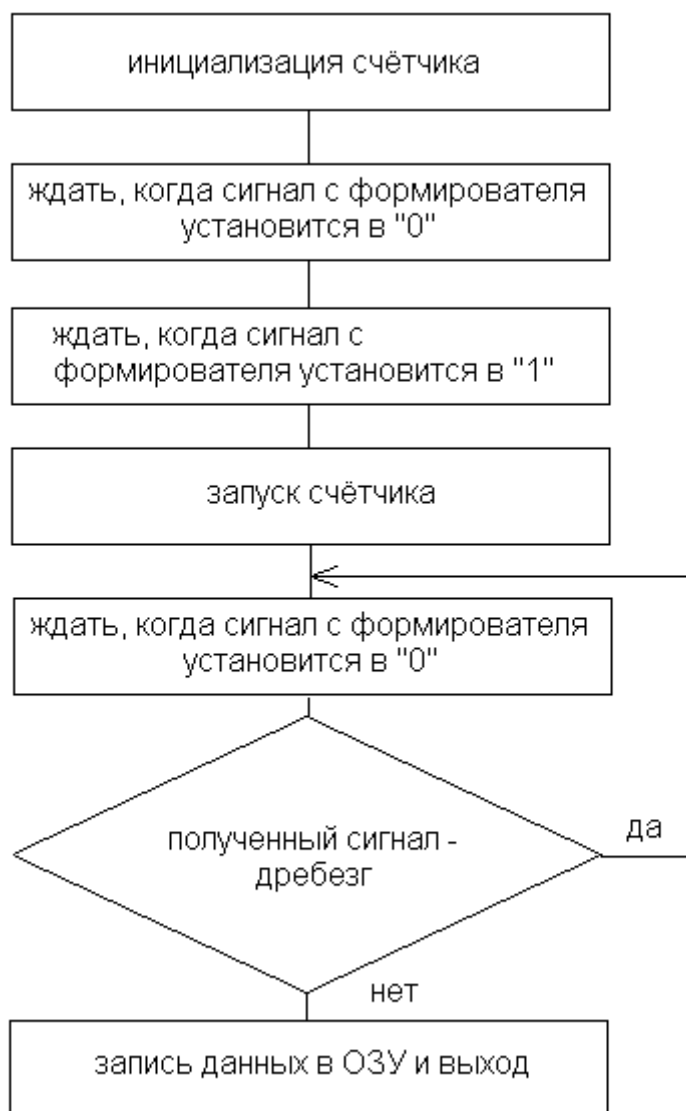


Рисунок 2.10- блок – схема алгоритма функции определения положения.

2.10.4 Функция формирования последовательности на шаговые двигатели

Управление шаговыми двигателями осуществляется с помощью четырёх похожих функций: поворот вверх, вниз, вправо, влево. Функции отличаются друг от друга только порядком выдачи управляющей последовательности, поэтому рассмотрим одну из них – поворот вверх. В соответствии с приведённой блок – схемой на шаговые двигатели для поворота на один шаг подаётся последовательность из четырёх сигналов, длительность каждого из которых 20 мс. После окончания движения двигателя отключаются от блока питания поскольку потребляют значительную мощность – 80 Вт.

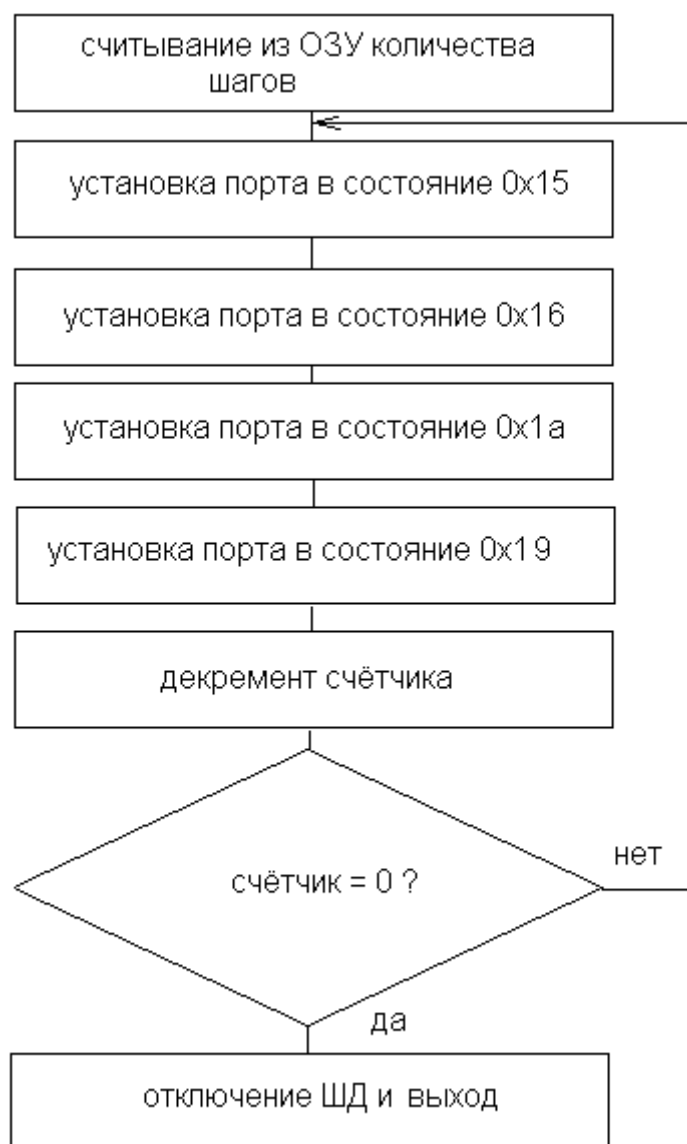


Рисунок 2.11 - блок – схема алгоритма функции поворота вверх.

2.10.5 Функция АЦП

В данном проекте АЦП производится по необычному принципу : во время работы АЦП процессорное ядро контроллера отключается, чем исключается его влияние на работу АЦП. После завершения работы АЦП контроллер просыпается по прерыванию. Блок – схема алгоритма функции АЦП приведена на рисунке 2.12.



Рисунок 2.12 - блок – схема алгоритма функции АЦП.

2.11 Исследование методов организации полигона для измерений

При измерении диаграммы направленности антенны одним из самых сложных проблем является соблюдение методики выполнения измерений для определения параметров по полю, соответствующей международным стандартам. Основным нормативным документом, содержащим требования к параметрам разрабатываемого комплекса, является [6],[7],[8]. Этот стандарт требует для комплексов I класса неравномерность ДН измерительного зонда в пределах раскрытия испытываемой антенны не более 0,8 дБ. Рассмотрим основные методы взаимного расположения элементов измерительной системы.

Наиболее простой метод – расположить измерительный зонд и испытываемую антенну таким образом, чтобы исключить отражённый от земли сигнал. Например при аттестации первичных эталонных антенн на государственных полигонах используется рельеф местности – между исследуемой антенной и зондом находится бездонная пропасть, из которой не выбирается ни одна отражённая волна. Этот метод требует наличия таких географических аномалий местности и транспортировку антенн до них. На менее «крутых» станциях используется безэховая камера с обеспечением двухкратных отражений с суммарным поглощением не менее 35 дБ. Однако такие камеры крайне дороги, и по этой причине недоступны рядовым разработчикам. Целью исследования является поиск наиболее простых и при этом достаточно достоверных технологий измерения диаграмм направленности.

Наиболее простой метод для измерения характеристик поля антенн СВЧ диапазона – использование измерительного зонда совместно с параболическим зеркалом, что позволяет выровнять амплитудно-фазовое распределение в раскрыве исследуемой антенны. Однако, несмотря на свою простоту, метод требует уделить большое внимание конструкции зеркало- измерительный зонд. Неправильная конструкция повлечёт интерференцию излучаемых и отражённых волн в исследуемой антенне. Также к отрицательным чертам этого метода является наличие большого параболического зеркала, которое трудно хранить в целости и сохранности.

Следующий признанный метод – измерение ДН при облучении её импульсами малой длительности с фильтрацией отраженных волн по времени прихода на антенну. Имеет много преимуществ и один недостаток – высокая стоимость оборудования.

Исследуем возможность использования ещё нескольких перспективных экономичных методов, которые не нашли широкого применения.

Прежде всего исследуем методы, которые не требуют специальной подготовки лаборатории – с фильтрацией отражённых волн по времени прихода. Как уже было замечено, в настоящее время широко используется метод облучения короткими импульсами. Поскольку быстродействующая импульсная техника дорога, попробуем воспользоваться менее дорогостоящим методом – измерения фазового сдвига на частоте модулирующего колебания с последующим расчетом мощности несущего колебания. Суть метода состоит в том, что зная фазы полезной и отражённой волн, а также амплитуды первой гармоники напряжения на детекторе, можно вычислить амплитуды полезного и отражённой волн. Структурная схема исследуемой модели представлена на рисунке 2.13. В данном случае рассматривается одна отражающая поверхность – пол. Итак, полезное излучение доходит до исследуемой антенны за время t_1 , в то время как отражённая волна доходит за время $t_1 + t_2$, где t_2 – задержка, обусловленная разностью хода лучей. При первом рассмотрении кажется, что напряжение на детекторе будет равно сумме напряжений, создаваемых мощностью прямых и отражённых радиоволн в соответствии с рисунком 2.14(Поскольку ток на выходе детектора пропорционален входной мощности). При этом происходит изменение амплитуды и фазы первой гармоники модулирующего колебания на выходе детектора. Зная фазу излучённого колебания и задержку t_2 (измерив расстояние до отражающей поверхности) можно в значительной степени уменьшить погрешность измерений, выделив из этой смеси

амплитуду полезного колебания.

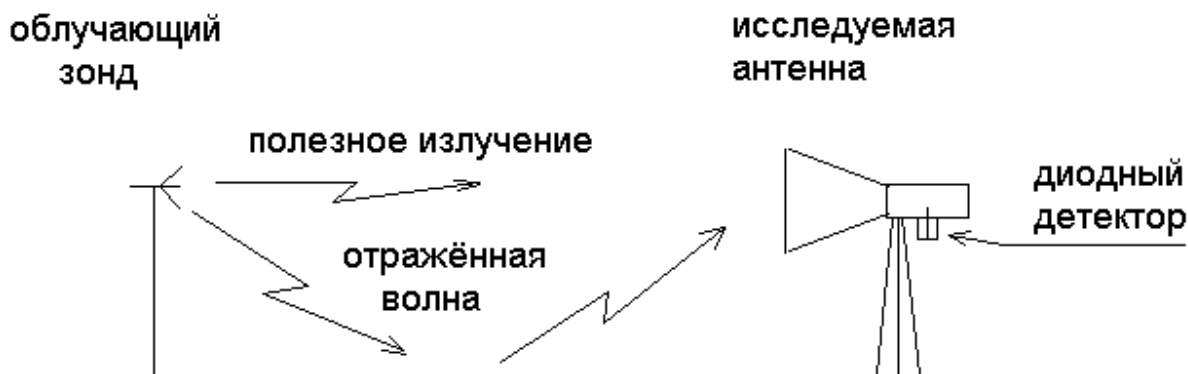


Рисунок 2.13 – модель полигона для измерений с учётом отражения радиоволн.

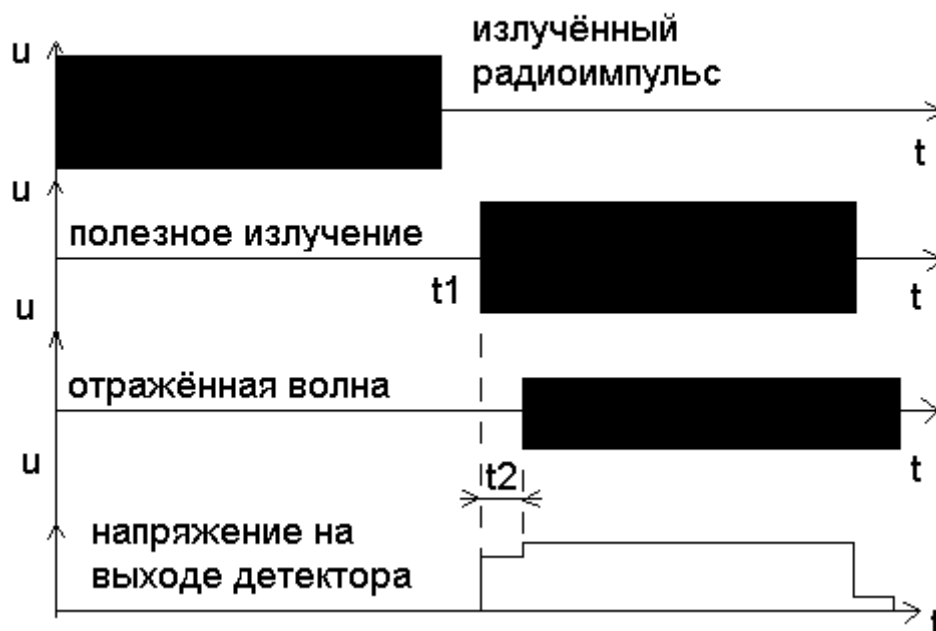


Рисунок 2.14 – графики зависимости напряжений в излучающем зонде, и в исследуемой антенне, созданных полезным излучением, отражённой волной и напряжения на выходе детектора исследуемой антенны (вариант 1).

Однако при более детальном рассмотрении оказывается, что из-за того, что полезная и отражённая волна имеют одинаковую частоту, в приёмной антенне происходит интерференция этих колебаний. Допустим, что отражённая и полезная волна приходит к антенне в противофазе, тогда напряжение на детекторе даже уменьшается. При этом графики соответствуют приведённым на рисунке 2.15. А если в фазе – увеличивается – как на рисунке 2.14. Таким образом, фаза и амплитуда первой гармоники модулирующего колебания на выходе детектора зависит не только от фаз модулирующих колебаний, но и от фаз несущих колебаний, что делает невозможным применение данного метода.

Следующий метод – менее практичный – учёт отражённого сигнала на несущей частоте. Для этого требуется знать фазу излучённого колебания, приходящего колебания и амплитуду колебаний в приёмной антенне. Что в свою очередь требует крайне чувствительного фазомера.

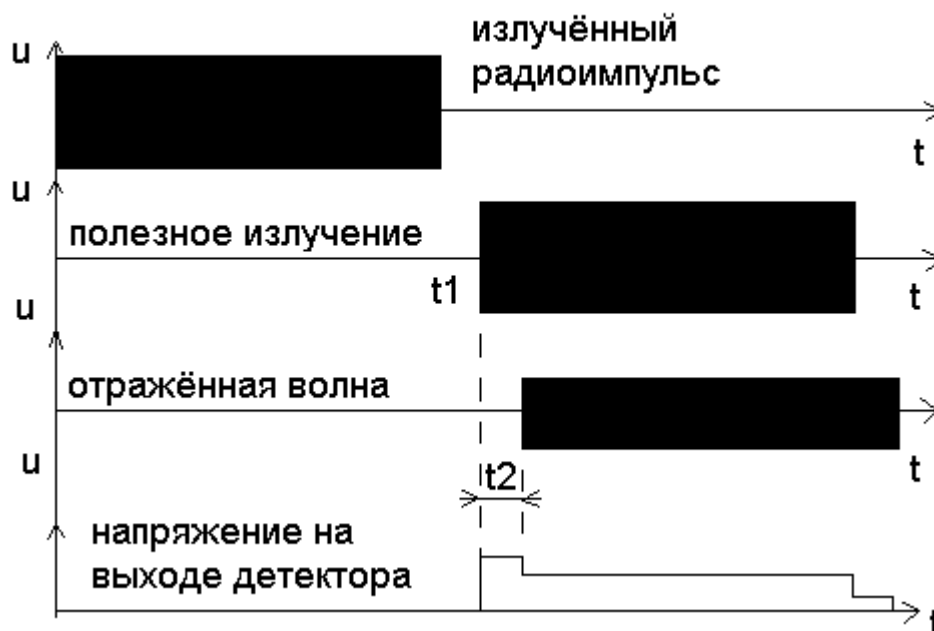


Рисунок 2.15 – графики зависимости напряжений в излучающем зонде, и в исследуемой антенне, созданных полезным излучением, отражённой волной и напряжения на выходе детектора исследуемой антенны (вариант 2) когда полезная и отражённая волна приходят в противофазе.

Кроме того, требуется прокладывать кабель от передающей антенны к приёмной и с высокой точностью вычислять фазу отражённого колебания, кроме того, если габариты исследуемой антенны превышают длину волны данный метод неприемлим, поскольку придётся учитывать фазы и амплитуды всех волн, попадающих в антенну.

Следующая группа методов посвящена принципам оборудования лаборатории, позволяющим исключить влияние отражённых волн. Как уже было замечено, применяемые в настоящее время методы очень эффективны и очень дороги.

Наиболее простым для понимания методом является установка отражающих экранов на пути предполагаемых отражённых волн, что наглядно показано на рисунке 2.16.

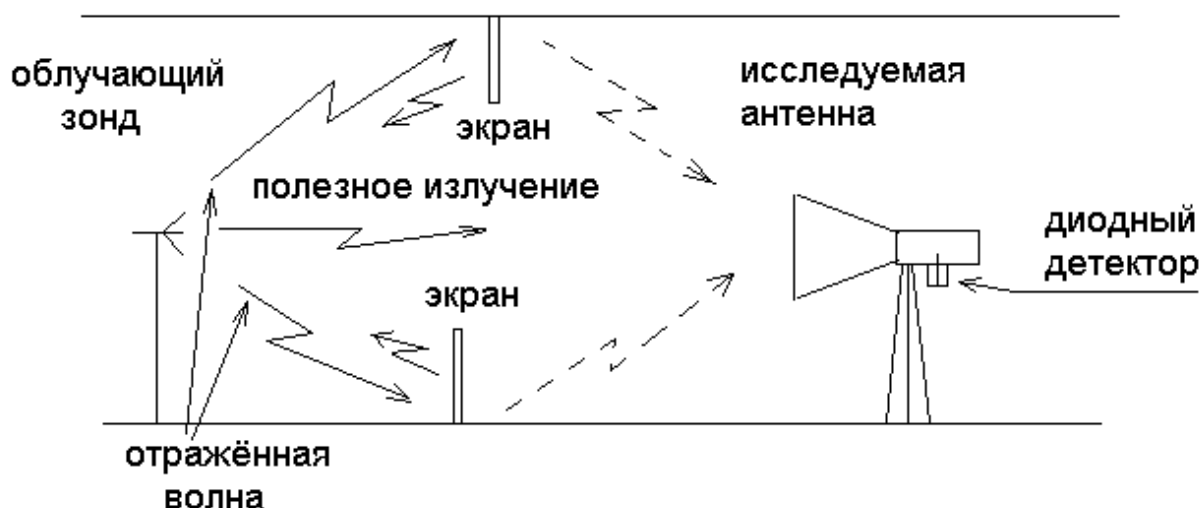


Рисунок 2.16 – уменьшение влияния отражённых волн методом установки экранов.

Как правило, в помещении присутствуют пол и потолок, следовательно, требуется установить два экрана. Кроме того, необходимо, чтобы на полу и потолку были установлены проводящие плиты. Это нужно для того, чтобы волна, приходящая на поверхность, отражалась по одному направлению, а не по всем. Недостатком данного метода является большие отражающие площади на полу и потолке.

Следующий метод от них свободен. Метод измерения ДН по отражённому лучу предполагает использования только одного экрана. Методика расположения экрана показана на рисунке 2.17.

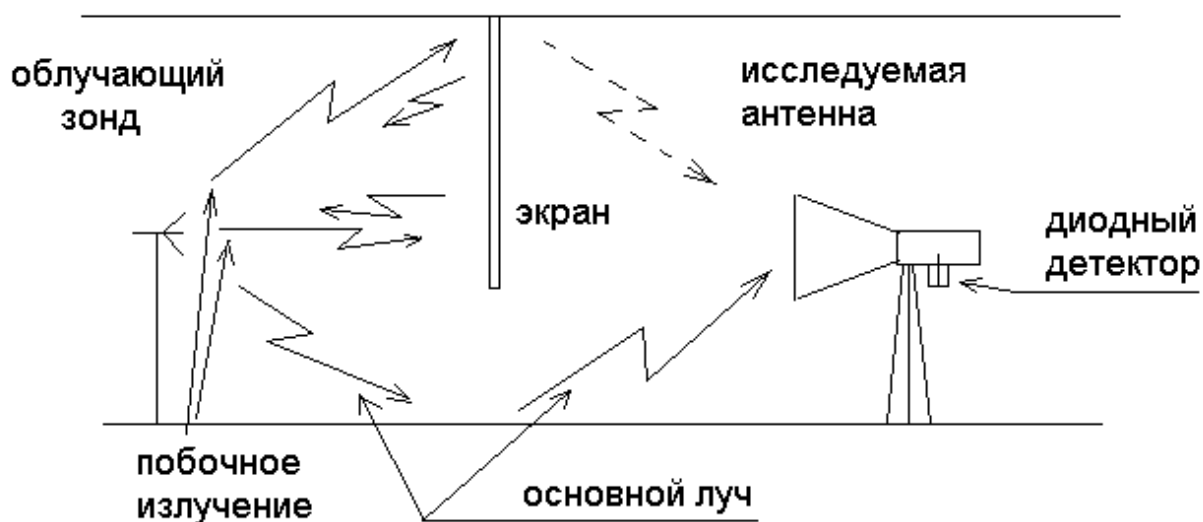


Рисунок 2.17 – схема измерения ДН по отражённому лучу.

В данном случае прямой луч подавляется экраном, который располагается между антеннами. Поскольку экран подвешивается к потолку, то отражённый от потолка луч

также подавляется этим экраном. Кроме того, луч, который потенциально мог отразиться от потолка, уменьшен благодаря ДН передающего зонда. Таким образом, его можно не учитывать. Остаётся только прямой и основной луч. В ходе проведённого математического моделирования в программе «Mmana», установлено, что подавление излучения экраном составляет более 10 дБ. Модель и результаты моделирования приведены на рисунках 2.18 и 2.19.

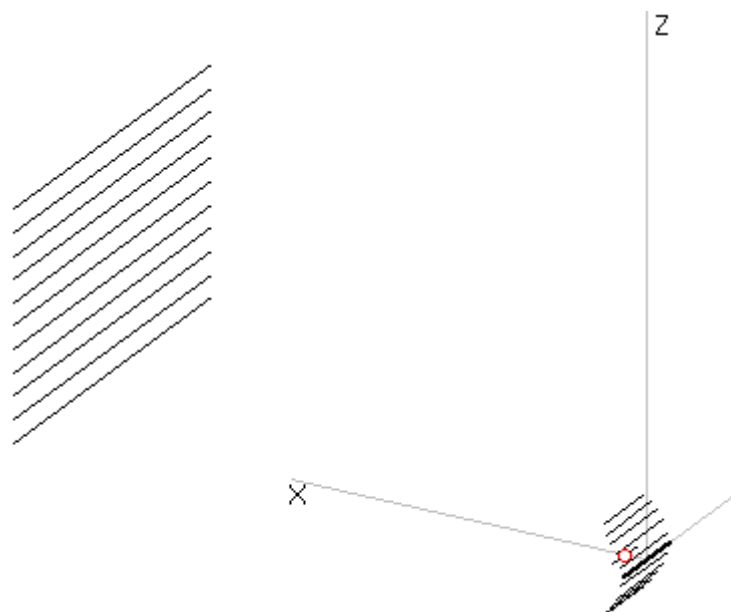


Рисунок 2.18 – модель системы в программе «Mmana».

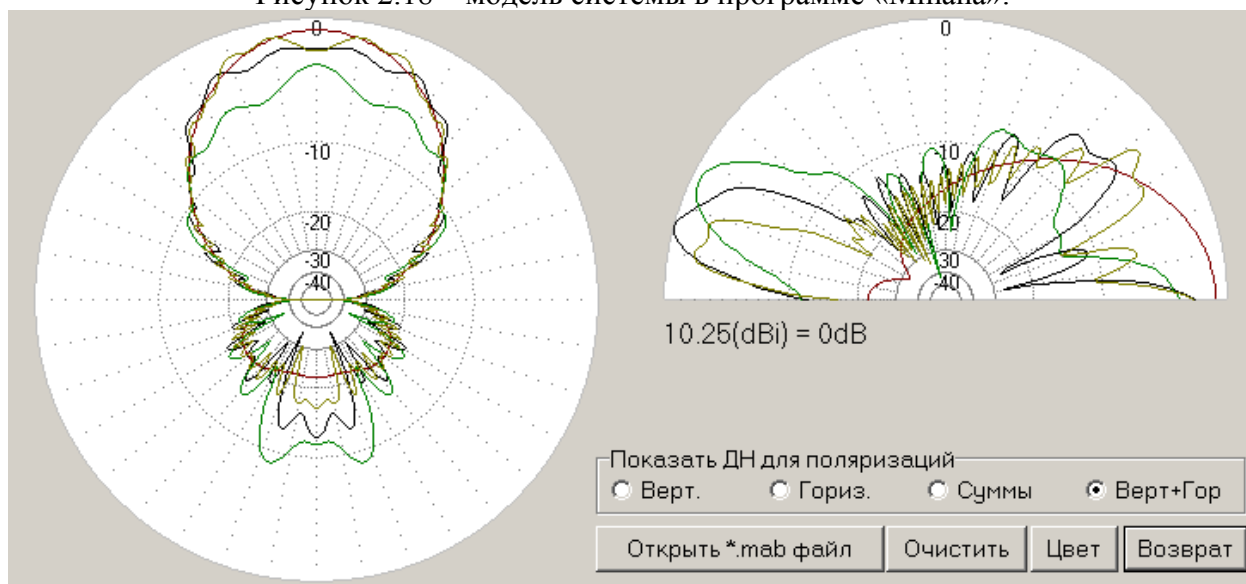


Рисунок 2.19 - ДН антенн в горизонтальной(слева) и вертикальной плоскостях.

Моделирование производилось в программе моделирования антенн «ММАНА». К сожалению, вычисление конструкций с экранами крайне сложен и занимает объём оперативной памяти исчисляемой десятками ГигаБайт, поэтому размер экрана был значительно снижен, что позволило сократить объём используемой ОЗУ до 250 Мбайт и

время вычисления – до 15 минут (на Pentium 4- 2.2 ГГц). Результаты моделирования сведены в таблицу 1: изменение сопротивления активного вибратора антенны приведена для возможности оценки корректности модели. В соответствии с таблицей, для сохранения свойств антенны (в том числе сопротивления вибратора) требуется размещать экран не ближе шести длин волн.

Для исследования выбран случай с расстоянием до экрана 9 длин волн, что позволяет «увеличить» площадь экрана. При расстоянии 12 длин волн до экрана, экран оказывается слишком маленьким, чтобы подавить побочное излучение. В исследуемой модели нижний край экрана располагался на 15 градусов выше центральной оси, поэтому ослабление, вносимое экраном, оценивалось именно в этом направлении.

Таблица 2.1 – результаты моделирования.

Расстояние до экрана, λ	Ослабление в направлении 15град. по вертикали, дБ	Изменение сопротивления активного вибратора антенны, %	Цвет диаграммы на графиках
Без экрана	1	0	коричневый
3	3	9	зелёный
6	10	2	чёрный
12	7	1.5	салатный

В соответствии с таблицей, для сохранения свойств антенны (в том числе сопротивления вибратора) требуется размещать экран не ближе шести длин волн.

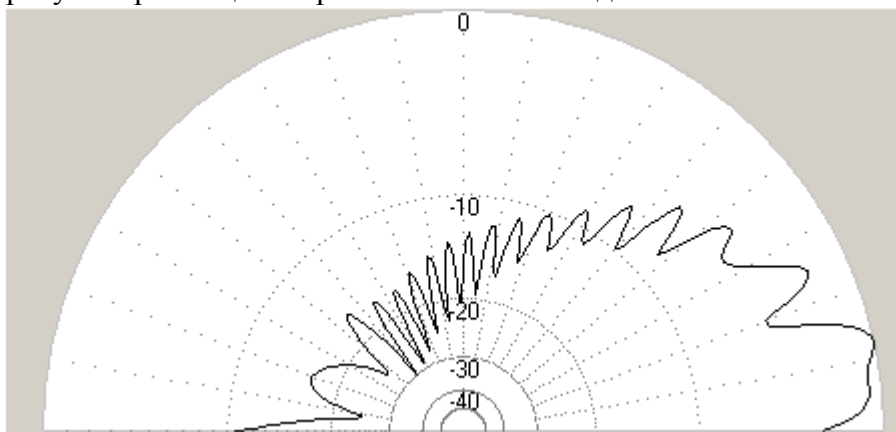


Рисунок 2.20 – ДН при изменении положения максимума излучения при установке экрана.

Естественно, максимум излучения при установке экрана смещается в другую сторону, что и показано на рисунке 2.20. Следует заметить, что неравномерность ДН в направлении максимума $\pm 5^\circ$ составляет $\pm 0,5$ дБ.

Недостатком данной математической модели является относительно малый размер экрана и относительно редкое расположение проводников экрана, однако имеющиеся ресурсы не позволяют его увеличить. В реальных условиях экран будет работать

значительно лучше, чем в данной модели. В первую очередь, уменьшится изрезанность ДН, что в значительной мере уменьшит неравномерность ДН облучающей части стенда.

Теперь определим минимально допустимые размеры экрана, который придётся положить на пол для обеспечения качественного отражения и отсутствия отражённых волн. Определимся с расположением ближнего к передающему зонду края экрана. Поскольку неизвестно, как отразится луч, попавший до экрана, будем считать, что он не доходит до исследуемой антенны. Теперь исследуем судьбу луча, попавшего на передний край экрана. Этот луч отразившись от экрана на полу далее попадёт либо на экран со стороны облучающего зонда, либо на потолок за экраном. В первом случае можно считать, что он будет подавлен при многократном отражении. Во втором – после отражения от потолка часть энергии обязательно попадёт на приёмную антенну. Поскольку антенны СВЧ имеют значительное подавление в направлении 90 и более градусов от направления максимума луча, допустим точку падения на потолок не ближе места установки исследуемой антенны. Из этих условий можно примерно рассчитать положение нижнего края подвесного экрана. В соответствии с рисунком 2.21 луч ABF попадает на передний край экрана, луч ACD – на задний, луч AE – проходит до места установки исследуемой антенны без отражений. Из геометрического положения луча ABF можно найти положение нижнего края вертикального экрана и переднего края горизонтального экрана. При этом нижний край экрана должен располагаться настолько низко, чтобы луч AE проходил ниже рабочей поверхности антенны. А дальний край горизонтального экрана должен находиться достаточно далеко, чтобы луч ACD также проходил ниже рабочей поверхности антенны.

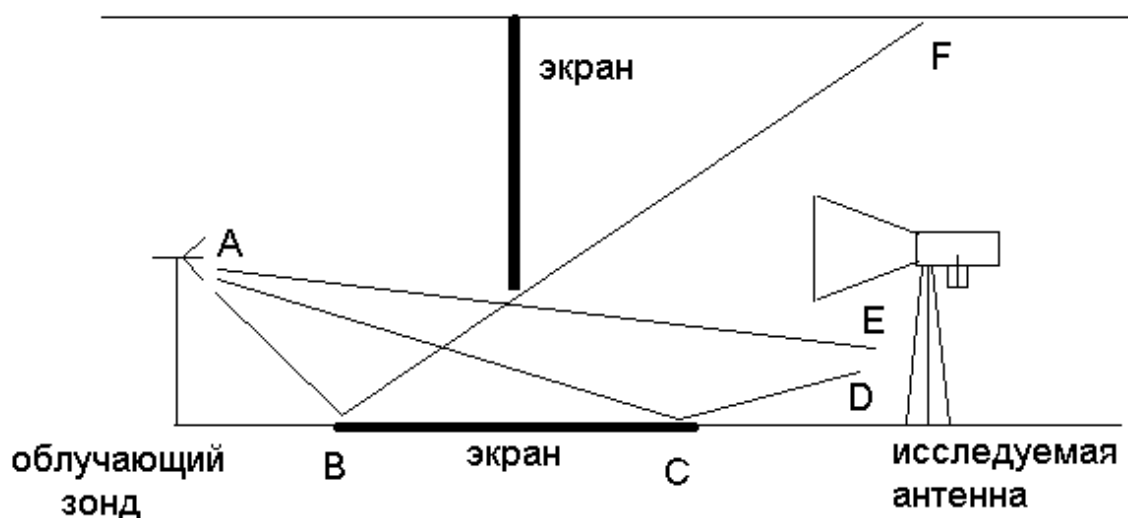


Рисунок 2.22 – местоположение экранов.

3.1 Расчёт ключей управляющих шаговыми двигателями

Далее рассчитываем ключи, модуль понижающего преобразователя в данной работе не рассчитывается.

1. определяем коэффициент усиления по току:

$$K := \frac{I_2}{I_1} ;$$

где $I_2 = 4 \text{ A}$ – ток ключа;

$I_1 = 0.005 \text{ A}$ – ток управления.

Получаем коэффициент усиления 800, которого нельзя достичь на одинарном мощном биполярном транзисторе, поэтому поставим полевой транзистор.

2. Выбираем распространённый полевой транзистор IRLR024 [5], используемый в блоках питания. Его основные характеристики :

максимально допустимое напряжение сток- исток – 55 В;

максимально допустимый ток стока – 17 А;

максимально допустимая рассеиваемая мощность – 45 Вт.

3. Определяем рассеиваемую мощность : по сток- затворной характеристике (рисунок3.1) находим напряжение на транзисторе при управляющем напряжении 4 В и токе в 4 А.

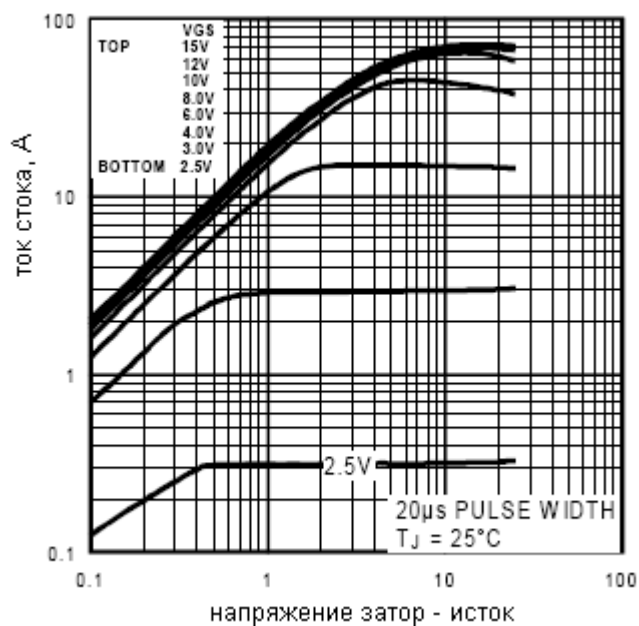


Рисунок 3.1 – Сток – затворная характеристика транзистора IRLR024.

Напряжение в импульсе – 0,3 В. Находим рассеиваемую на транзисторе мощность:

$$P := \frac{1}{N} \cdot U \cdot I ;$$

где $N = 4$ – скважность,

$U = 0,3$ В – напряжение во время импульса, В,

$I = 4$ А - ток в импульсе.

Получаем рассеиваемую мощность 0,3 Вт.

3.2 Расчёт схемы приёмника

3.2.1 Расчёт программно – управляемого логарифмирующего измерительного приёмника

В соответствии с функциональной схемой была разработана принципиальная схема.

Наиболее сложная её часть – активный фильтр. Его принципиальная схема представлена на рисунке 3.2. Резисторы R3, R1 образуют делитель на 2 и создают виртуальную среднюю точку для работы ОУ DA1. Элементы C3, R2, C4, C6 - C8, R4-R8 образуют частотно-зависимую цепь ООС. Её расчет приведён в приложении.

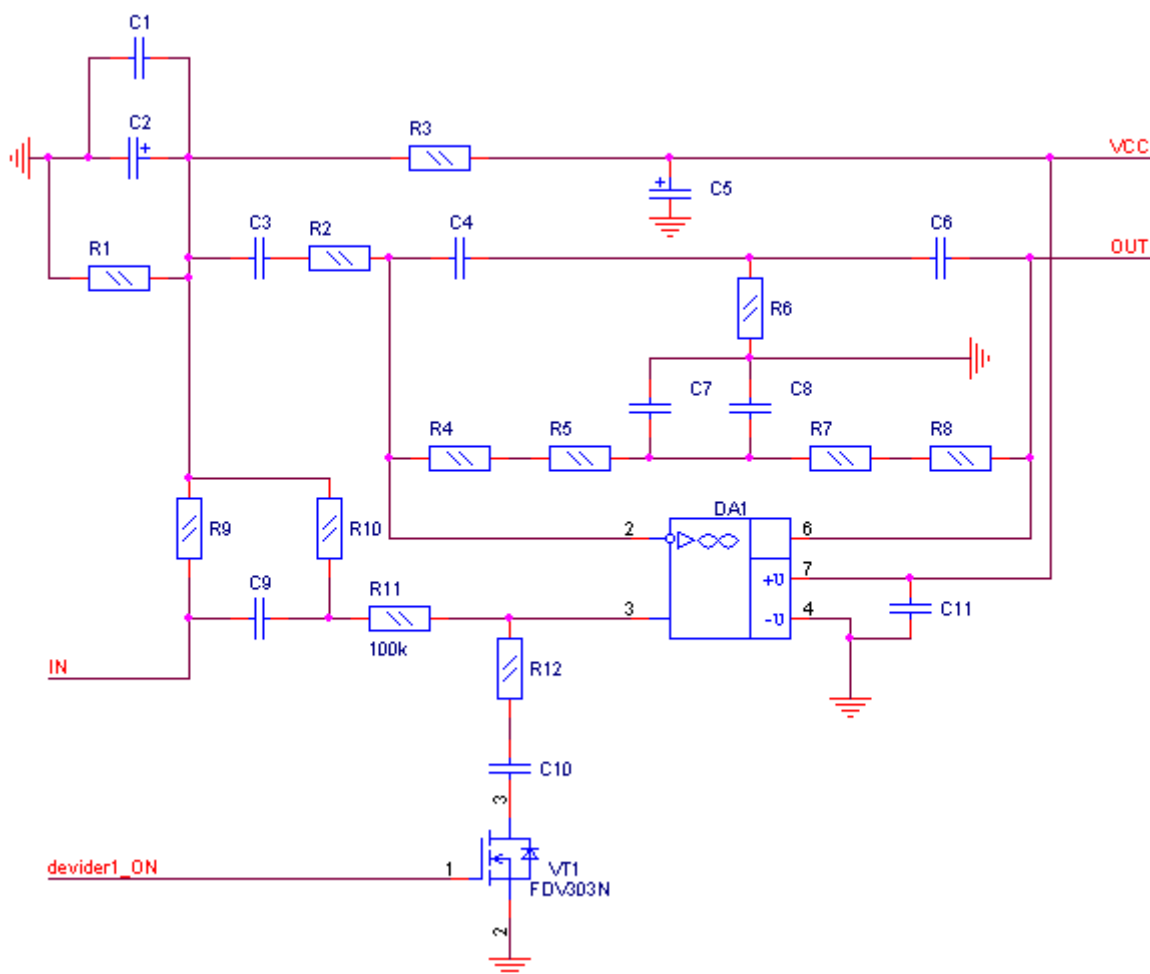


Рисунок 3.2 - принципиальная схема активного фильтра.

R9 предназначен для смещения в прямом направлении детекторного диода. Исходя из документации на применяемый детекторный диод Д603, оптимальным смещением является ток $I_{opt} = 20$ мкА. Рассчитаем номинал резистора R9, учитывая, что падение на нём $U_{vd}=0,5$ В, а напряжение на делителе R3 R1 – $U_{in}=2.5$ В.

$$R9 = \frac{U_{in} - U_{vd}}{I_{opt}} = \frac{2.5 - 0.5}{20 \cdot 10^{-6}} = 10^5 \text{ Ом.}$$

Часть схемы, состоящая из элементов R11, R12, C10, VT1, - это входной управляемый делитель на 100. Если на затвор VT1 подан логический ноль, то коэффициент передачи делителя равен 1, если логическая единица – 0,01. Номиналы резисторов R11 – 100 кОм, R12- 1 кОм.

Расчетная АЧХ фильтра приведена на рисунке 3.3. Лёгкое несовпадение максимума расчетной АЧХ с требуемой будет устранено в процессе настройки фильтра.

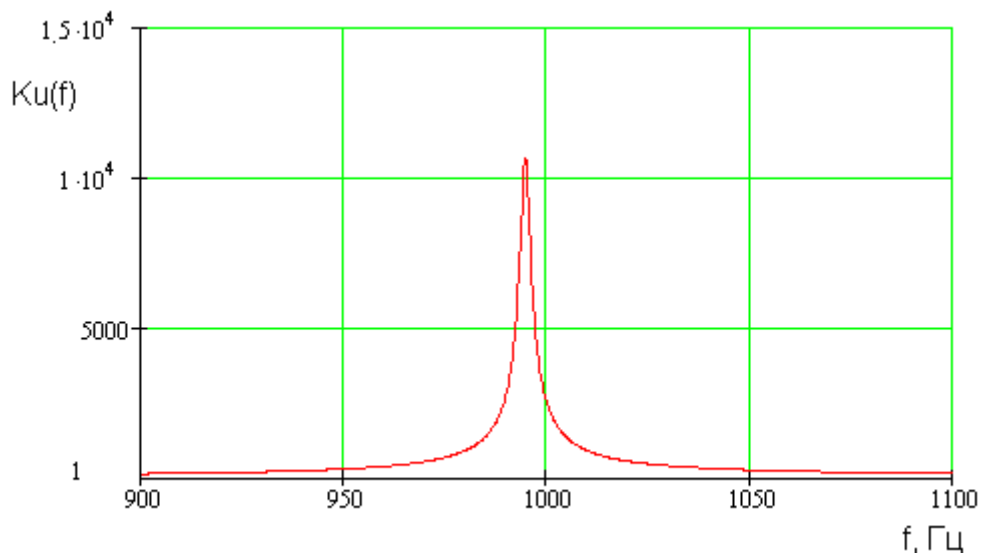


Рисунок 3.3 – расчетная АЧХ активного фильтра.

После настройки фильтра его АЧХ стала выглядеть так, как показано на рисунке 3.4.

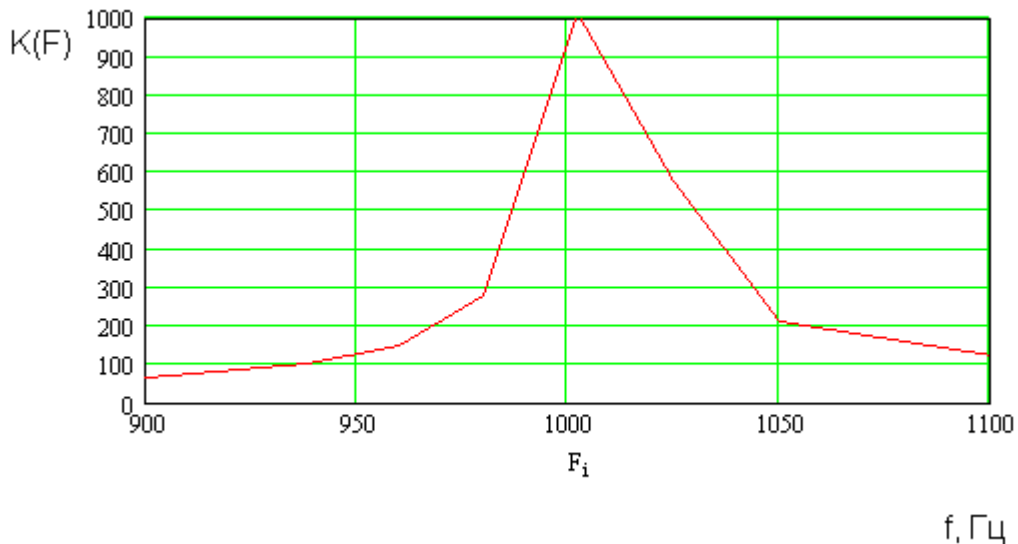


Рисунок 3.4 – измеренная АЧХ фильтра.

Рассчитаем уровень собственных шумов фильтра. Для этого найдём спектральную плотность шумов на входе фильтра, перемножим на нормированную АЧХ и проинтегрируем по частоте. Спектральную плотность шумов на входе фильтра определяют три составляющих: напряжение тепловых шумов на сопротивлении источника сигнала, уровень шумов ОУ по напряжению и току. Сопротивление источника сигнала для ОУ задаёт резистор R11. Найдём его спектральную плотность тепловых шумов. Как известно,

$$En = \sqrt{4 \cdot k \cdot T \cdot R \cdot \Delta F} ;$$

при полосе 1 Гц и сопротивлением 1 кОм уровень тепловых шумов En1 равен 4 нВ.

Тогда на сопротивлении R11 спектральную плотность тепловых шумов

$$ER11 = En1 \cdot \sqrt{R11} = 4 \cdot 10^{-9} \cdot \sqrt{100} = 40 \cdot 10^{-9} B / \sqrt{Гц} ;$$

Уровень собственных шумов операционного усилителя En = 29 нВ/Гц. Найдём уровень шума, обусловленного входным током ОУ. По документации спектральная плотность входного шумового тока составляет $In = 0.6 \cdot 10^{-15} A$, тогда спектральная плотность входного напряжения EIn, им обусловленная, составит

$$EIn = In \cdot R11 = 0.6 \cdot 10^{-15} \cdot 10^5 = 0.06 \cdot 10^{-9} B / \sqrt{Гц}.$$

Тогда результирующая спектральная плотность входного шумового напряжения En_in

$$En_in = \sqrt{ER11^2 + En^2 + EIn^2} = \sqrt{(40 \cdot 10^{-9})^2 + (29 \cdot 10^{-9})^2 + (0.06 \cdot 10^{-9})^2} = 49 \cdot 10^{-8} B / \sqrt{Гц};$$

теперь найдём шумовое напряжение Eш, приведённое ко входу приёмника

$$Eш = En_in \cdot \sqrt{\int_0^{\infty} K(f)^2 df} = 0.6 мкВ .$$

Измеренное напряжение шума у собранного приёмника оказалось 2,5 мкВ. Повышенный уровень шума можно объяснить помехами по цепям питания и отсутствием экранирования приёмника. Теперь перейдём к следующей части приёмника – управляемому усилителю.

Управляемый усилитель изображён на рисунке 3.5.

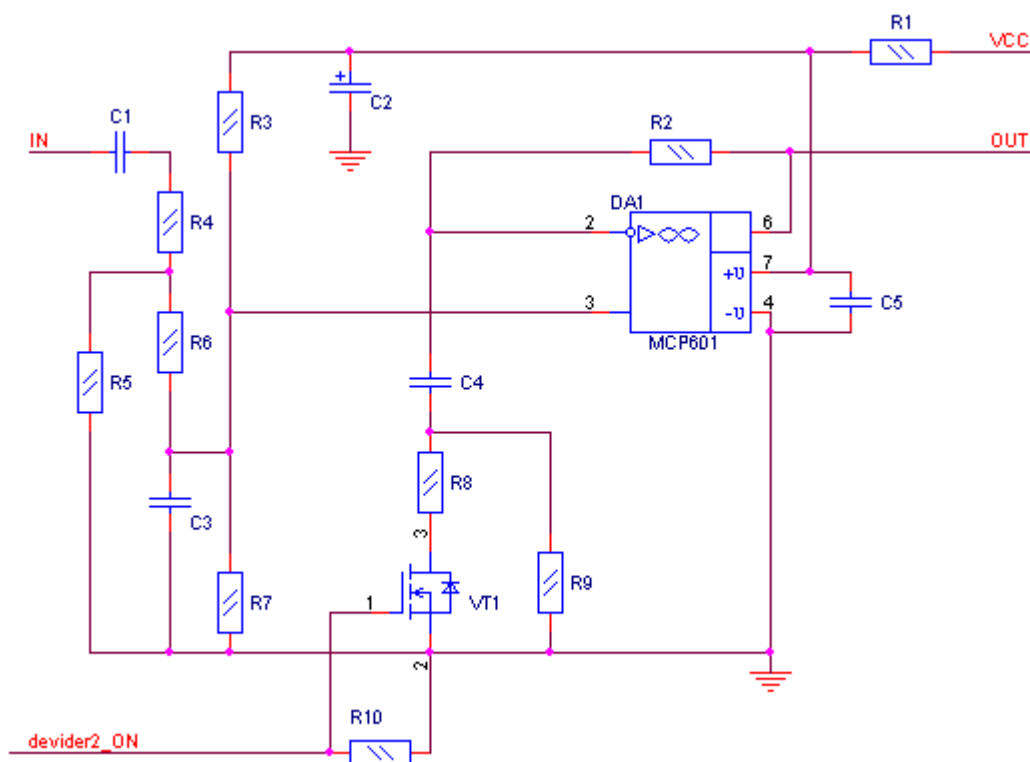


Рисунок 3.5 - управляемый усилитель.

C1- развязывающий конденсатор. R4-R6 и C3 обеспечивают развязку с предыдущим каскадом и одновременно делит входной сигнал на 10. Элементы R2, R8, R9, C4, VT1 входят в управляемую ООС. При подаче логической единицы на затвор VT1 он открывается и коэффициент деления в цепи ООС возрастает, в результате чего коэффициент усиления возрастает до 100. При закрытом VT1 коэффициент усиления близок к одному. Элементы R1 C2 обеспечивают развязку по питанию. Следующая часть схемы – квазипиковый детектор.

Схема квазипикового детектора показана на рисунке 3.6.

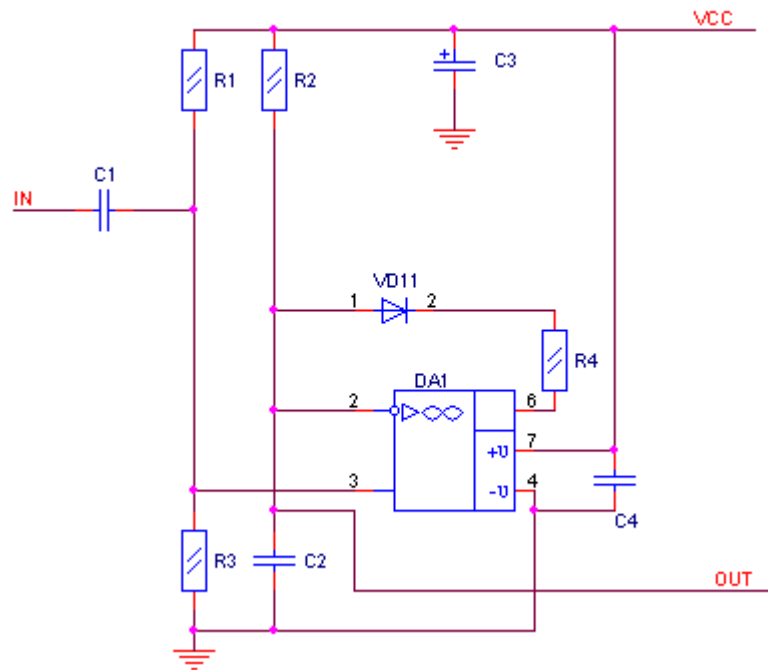


Рисунок 3.6 – схема квазипикового детектора.

C1 – разделяющий конденсатор; R1, R2 – задаёт начальное напряжение на выходе детектора. C2, R2 и R4 определяют постоянную времени разряда и заряда детектора. Схема квазипикового детектора была выбрана так как она обладает широким диапазоном входных напряжений, высокой помехозащищённостью и проста в построении. С выхода квазипикового детектора сигнал поступает на АЦП контроллера. Там сигнал усредняется, производится логарифмирование отсчётов, после чего результат выводится в виде широтно-импульсного цифрового сигнала. Пройдя через фильтр НЧ он превращается в постоянное напряжение, которое может быть измерено АЦП главного контроллера, либо вольтметром постоянного тока при использовании этого приёмника отдельно от комплекса. Измеренная зависимость постоянного выходного напряжения от уровня входной мощности приведена на рисунке 3.7.

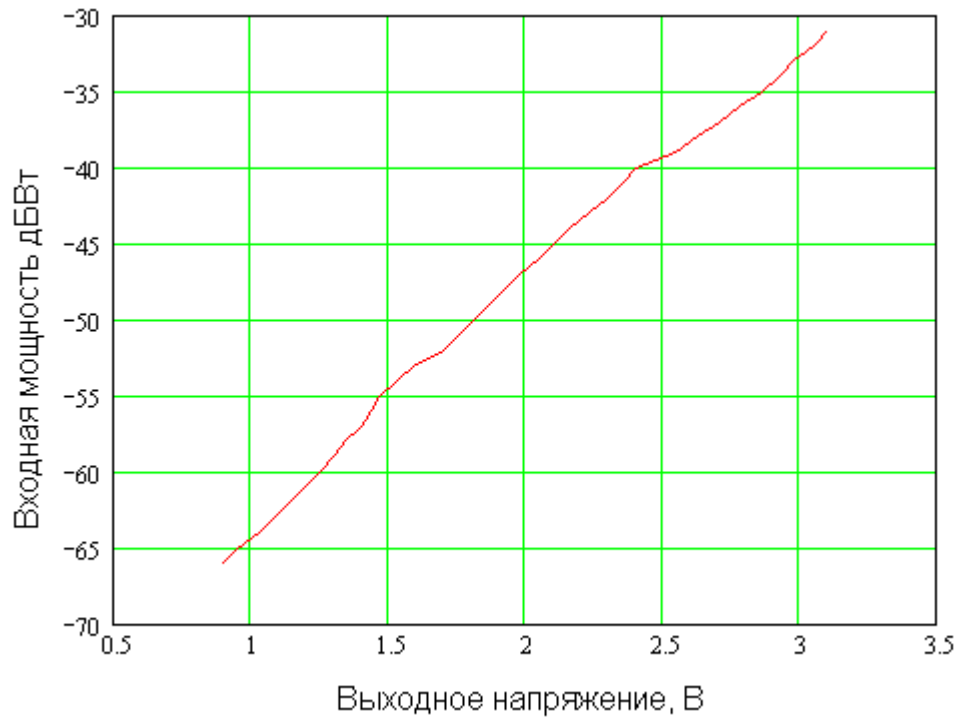


Рисунок 3.7 - зависимость постоянного выходного напряжения от уровня входной мощности приёмника.

В качестве управляющего приёмником применён submodule контроллера ИТЕА.468243.008. Кроме того, предусмотрена возможность подачи переменного напряжения, поступающего на вход детектора, непосредственно на вход АЦП контроллера. Таким образом при высоком уровне шумов можно будет производить цифровую фильтрацию и программное детектирование, таким образом повысив чувствительность приёмника. Предполагается в качестве фильтра использовать рекурсивный фильтр, имитирующий колебательный контур. Таким образом снизить затраты времени процессора на вычисление. Более оптимальные фильтры требуют больше переменных для построения, кроме того применение выбранной модели позволит быстро изменять его характеристики. При расчете были получены следующие коэффициенты системной функции:

$$A0 = \frac{Yk}{\frac{\Delta}{2 \cdot L} + \frac{2 \cdot C}{\Delta} + Yk};$$

$$A1 = 0;$$

$$A2 = -A0;$$

$$B0 = \frac{\Delta}{2 \cdot L} + \frac{2 \cdot C}{\Delta} + Yk;$$

$$B1 = -\frac{\frac{\Delta}{L} - \frac{4 \cdot C}{\Delta}}{B0};$$

$$B2 = -\frac{\frac{\Delta}{2 \cdot L} - \frac{2 \cdot C}{\Delta} - Yk}{B0}.$$

Эта системная функция

$$H(z) = \frac{A0 \cdot z^2 + A1 \cdot z + A2}{z^2 - B1 \cdot z - B2}$$

имеет передаточную характеристику

$$K(\omega) = \frac{A0 \cdot (e^{j \cdot \omega \cdot \Delta})^2 + A1 \cdot e^{j \cdot \omega \cdot \Delta} + A2}{(e^{j \cdot \omega \cdot \Delta})^2 + B1 \cdot e^{j \cdot \omega \cdot \Delta} + B2}.$$

В этих формулах Yk - проводимость контура в резонансе

$$Yk = \frac{1}{(\omega \cdot L \cdot Q)};$$

Δ – период дискретизации. L , C и Yk рассчитываются исходя из добротности Q и центральной частоты фильтра ω . При заданной добротности фильтра 100 получена передаточная характеристика, изображённая на рисунке 3.8.

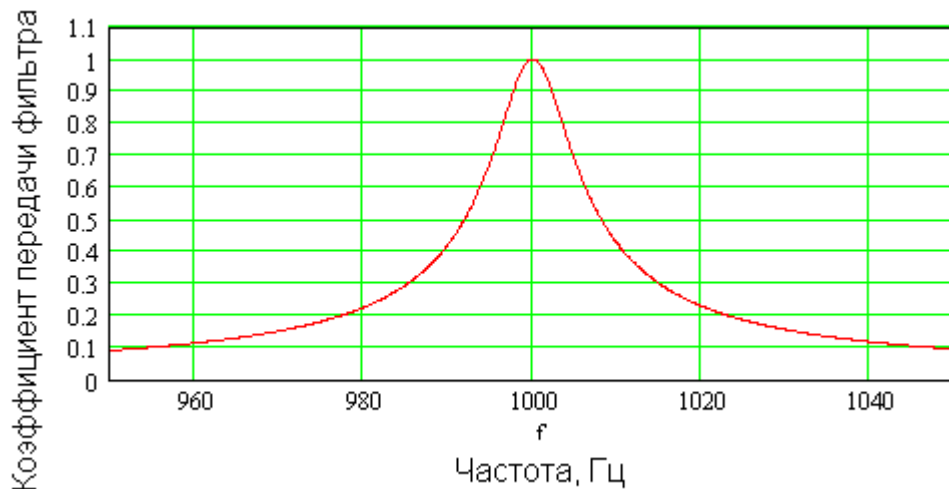


Рисунок 3.8 - передаточная характеристика модели цифрового фильтра.

При этом была произведена оптимизация коэффициентов функции. Для уменьшения времени вычисления все коэффициенты по возможности приведены к степени по основанию 2. Так как деление и умножение на 2 производится в целом виде сдвигом вправо или влево на 1 бит.

$$A0 = \frac{3}{2^{10}};$$

$$A1 = 0;$$

$$A2 = -A0;$$

$$B1 = 1 + \frac{1}{2^1} + \frac{1}{2^3} - \frac{3}{2^8};$$

$$B2 = \frac{3}{2^9} - 1.$$

При этом затраты на вычисление каждого выходного отсчёта ЦФ будет производится примерно в $T_r=100$ тактов процессора. При тактовой частоте $F_t=8$ МГц максимальная частота дискретизации составит

$$F_{\Delta} = \frac{F_t}{T_r} = \frac{8 \cdot 10^6}{100} = 80 \text{ кГц}.$$

Что вполне допустимо при дискретизации сигнала частотой 1 кГц.

Для программного логарифмирования в контроллере применяется следующий алгоритм: полученное значение амплитуды представляется в виде сомножителя степени числа 2 и целого числа в диапазоне от 0 до 32. В таблице находятся 32 значения, соответствующих переводу 32 младших значений в дециБеллы. Программа делит исходную величину на 2 методом сдвига влево в процессоре, что соответствует увеличению результата на 3 дБ. Когда переменная становится меньше 32, ячейка матрицы с номером, соответствующей значению переменной считывается из памяти и добавляется к результату перевода. Далее в зависимости от того, были ли включены усилитель и аттенюатор, добавляется ими вносимое значение усиления или ослабления. Текст программы приведён на рисунке 3.9.

```

dB = 0;
if (ADCR > 31)
{
    while (ADCR > 31)
    {
        ADCR = ADCR >> 1;
        dB += 48; // - 3 dB
    }
}
dB += ADC32[ADCR];
if (KU1_ON) dB -= 259; // - 18 dB
if (KU2_ON) dB += 237; // + 18 dB

```

Рисунок 3.9 – фрагмент программы, отвечающей за логарифмирование.

После логарифмирования результат выводится в виде широтно-импульсного цифрового сигнала, который после фильтрации подаётся на АЦП. К достоинствам такого метода построения относится простота и минимальные размеры приёмника, возможность работы в составе комплекса или вне него. При работе отдельно, приёмник может быть подключён по USB и питаться от неё, так как питание приёмника осуществляется от +5В.

3.2.2 Расчёт аппаратного логарифмирующего преобразователя приёмника с кусочно-линейной аппроксимацией

Первый вариант приёмника имеет выполняющий программное логарифмирование, в этой главе рассматривается аппаратный метод реализации логарифмирования. При разработке модуля логарифмирующего преобразователя-усилителя с кусочно – линейной аппроксимацией (далее усилителя) стояла задача кроме логарифмирования выполнить согласование фильтра-детектора и АЦП контроллера. Модуль детектора имеет диапазон выходных напряжений 0,1-14 В, диапазон входного напряжения АЦП - 0-5 В. Кроме того, необходима нелинейная характеристика для частичной компенсации нелинейности диодного фильтра-детектора, характеристика которого приведена на рисунке 3.10.



Рисунок 3.10 – Зависимость нормированного выходного напряжения модуля детектора от относительного уровня мощности входного сигнала (красная сплошная линия) и идеальная зависимость (обозначена синими точками).

Передаточная характеристика корректирующего усилителя изображена на рисунке 3.11.



Рисунок 3.11 – Зависимость требуемого нормированного выходного напряжения корректирующего усилителя от нормированного входного напряжения.

Как видно из этого рисунка, характеристика корректирующего усилителя должна иметь характер, близкий к логарифмическому. Однако имея однополярное питание на плате модуля, нельзя поставить традиционную схему логарифмического усилителя.

Поэтому исследуем схему усилителя, представленную на рисунке 3.10. В качестве диода используем малогабаритный диод Шотки BAS-70. Примем максимальный выходной ток модуля детектора 3 мА. По графику на рисунке 3.11. Находим, что при прямом токе 3 мА на диоде падает 0,5 В. Находим номинал резистора R1:

$$R1 = \frac{U_{вых} - U_{пр}}{I_{макс}} ;$$

где $U_{вых} = 14 \text{ В}$ – максимальное выходное напряжение с модуля детектора;

$U_{пр} = 0,5 \text{ В}$ – прямое напряжение на диоде при токе 3 мА;

$I_{макс} = 3 \text{ мА}$ – максимальный ток через диод.

Получаем $R1 = 4,5 \text{ кОм}$, ближайший номинал – 4,7 кОм.

Далее определяем резисторы в цепи ОС ОУ. Необходимый коэффициент усиления

$$K := \frac{U_{вых_макс}}{U_{вх_макс}} ;$$

где $U_{вых_макс} = 5 \text{ В}$ – максимальное выходное напряжение усилителя,

$U_{вх} = 0,5$ – максимальное напряжение на диоде.

Получаем коэффициент усиления 10. Примем номинал резистора R3 1кОм.

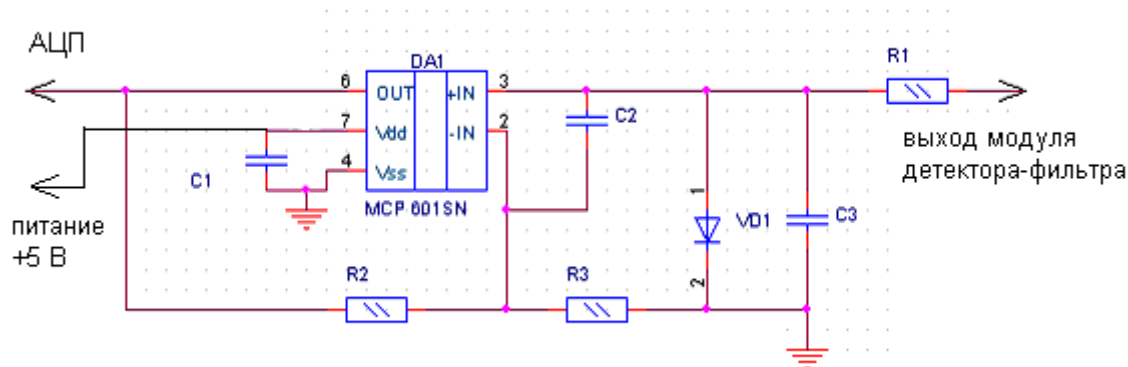


Рисунок 3.12- Схема нелинейного усилителя.

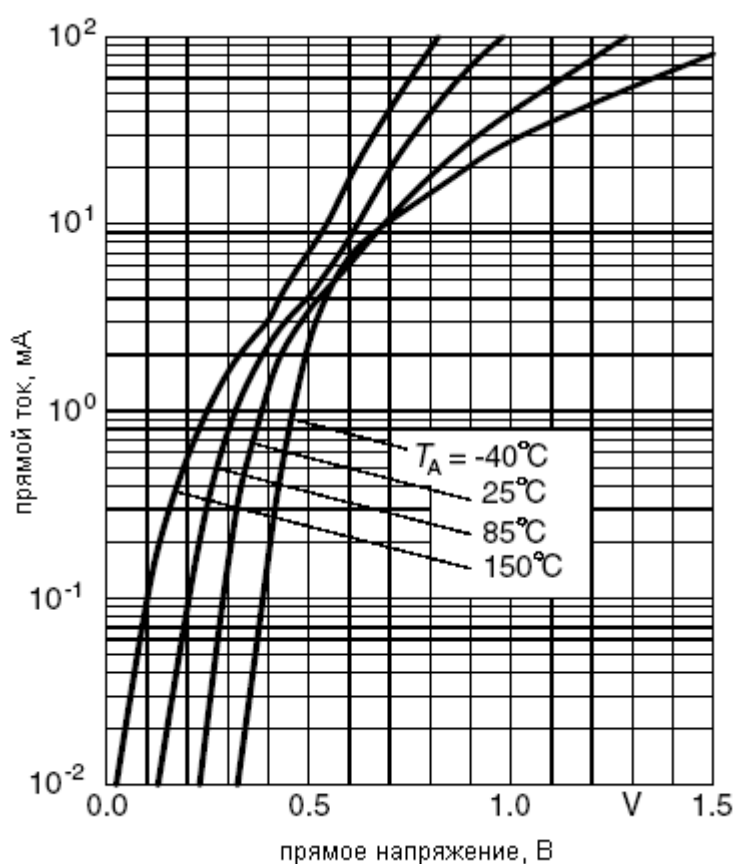


Рисунок 3.13 - Вольт – амперная характеристика диода BAS-70.

Для такого включения операционного усилителя резистор R2 рассчитывается по формуле

$$R2 = (K - 1) \cdot R3 ;$$

где K = 10 – необходимый коэффициент усиления;

R3 = 1 кОм – номинал резистора R3.

Получаем R2 = 9 кОм, ближайший номинал 9,1 кОм.

Определяем ёмкости фильтра C3. Поскольку время установления составляет порядка 1мс , то необходимая ёмкость

$$C = \frac{T_{уст}}{4 \cdot R} ;$$

где $T_{уст} = 1$ мс – время установления,

$R = 4,7$ кОм - сопротивление фильтра.

Получаем $C3 = 53$ нФ, ближайший номинал – 47нФ.

Конденсатор $C2$ берём в 100 раз меньше $C3$, т.е. 470 пФ. Она необходима для дополнительной фильтрации.

Конденсатор $C1$ берём 1 мкФ, в соответствии с рекомендацией производителя микросхемы.

Проведём моделирование в PSpice simulator 9.1. График зависимости выходного напряжения от входного напряжения схемы представлен на рисунке 3.14.

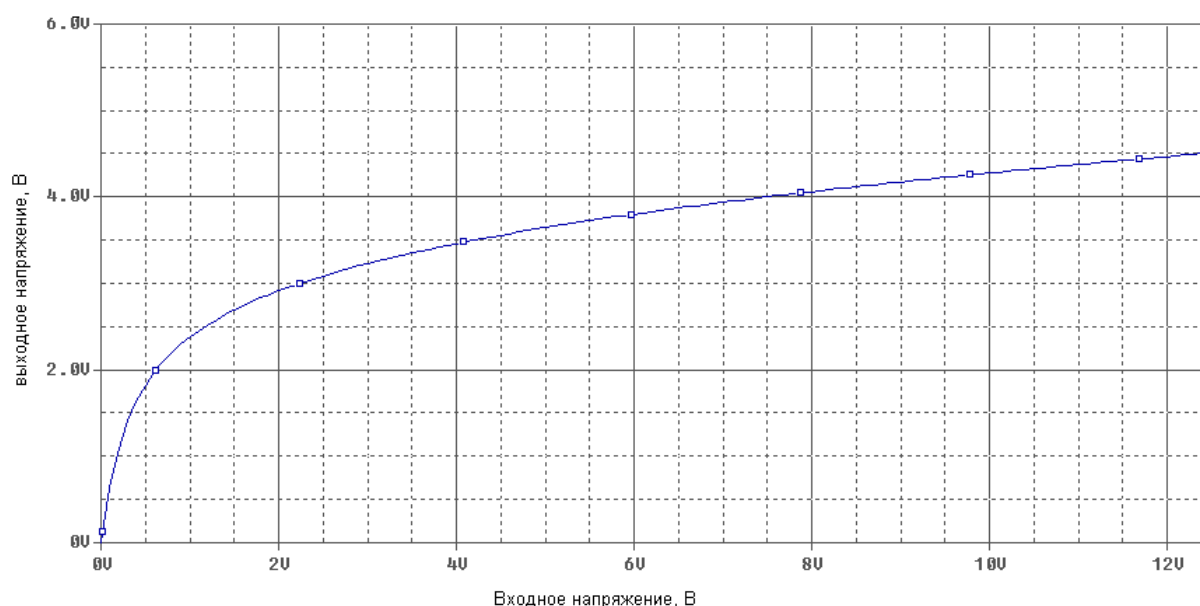


Рисунок 3.14 - График зависимости выходного напряжения от входного напряжения нелинейного усилителя с диодом Шоттки, результат моделирования в PSpice simulator 9.1.

Как видно из графика, рассчитанный усилитель действительно преобразует 12 – вольтный диапазон входных напряжений в 5 – вольтный диапазон выходных, теперь проверим температурную стабильность схемы. Проведя моделирование в PSpice simulator 9.1 находим, что в интервале температур от 0 до 40 градусов по Цельсию точность выходного напряжения составляет от 5 до 12 % (в зависимости от входного напряжения), поэтому для наших целей она непригодна. По результатам моделирования получается, что точность выходного напряжения сохраняется в пределах 1% при условии, что температура меняется в пределах 1,5 градуса. Если мы будем использовать эту схему, то придётся калиброваться каждый раз. Попробуем заменить кремниевый диод Шоттки на арсенид –

галлиевый светодиод, который потенциально имеет большую температурную стабильность. График зависимости возможного относительного отклонения выходного напряжения от входного напряжения приведён на рисунке 3.15.

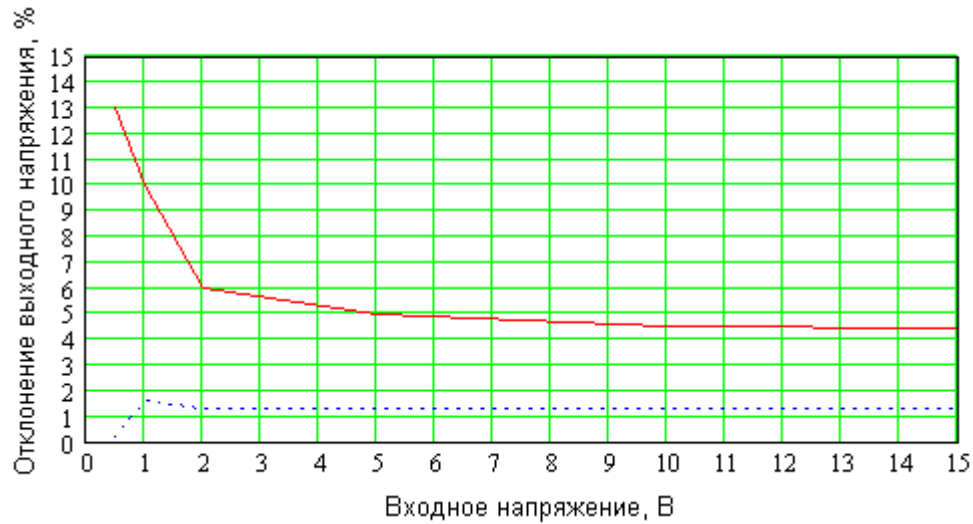


Рисунок 3.15 – График зависимости возможного отклонения выходного напряжения от входного напряжения при изменении температуры на 20 градусов.

Проведя моделирование в PSpice simulator 9.1 находим, что в интервале температур от 0 до 40 градусов по Цельсию точность выходного напряжения составляет от 0,1 до 1,6 % (в зависимости от входного напряжения), поэтому для наших целей он может подойти. Однако имеет очень крутой наклон характеристики.

График зависимости выходного напряжения от входного напряжения такой схемы представлен на рисунке 3.16.

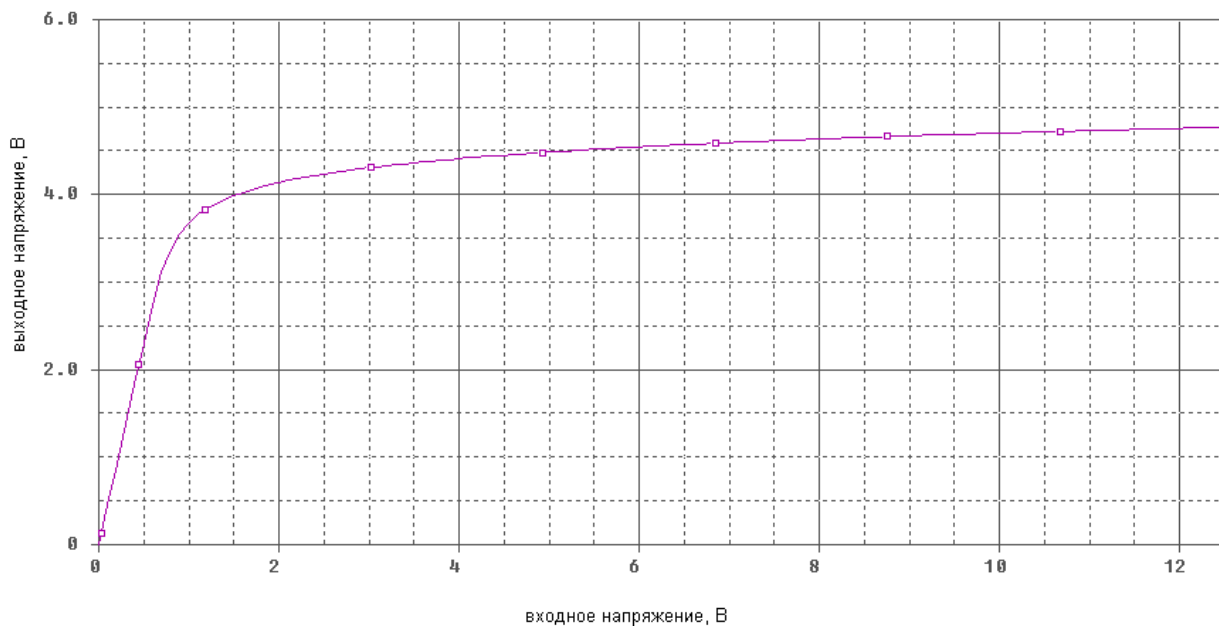


Рисунок 3.16 - График зависимости выходного напряжения от входного напряжения нелинейного усилителя с арсенид – галлиевым диодом, результат моделирования в PSpice simulator 9.1.

Заметим, что использование такого НЛУ также нецелесообразно, поскольку крутой наклон характеристики приведёт к потере точности. Поэтому ограничим крутизну характеристики, добавив последовательно светодиоду резистор. Получившаяся схема приведена на рисунке 3.17.

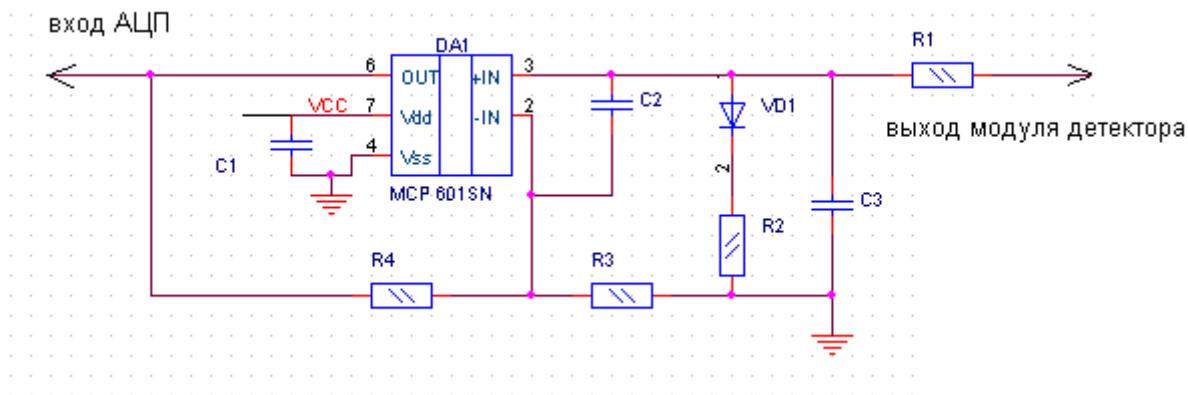


Рисунок 3.17 – Нелинейный усилитель с арсенид-галлиевым диодом и последовательно включённым сопротивлением.

Рассчитаем параметры элементов. Поскольку арсенид – галлиевый диодоткрывается при напряжении 1,5 В, зададимся коэффициентом усиления 2. Примем номинал R4 равным 10 кОм, номинал R1 равным 4,7 кОм. Определим R3:

$$R3 = (K - 1) \cdot R4 ;$$

где R4 = 10 кОм – номинал резистора R3,

K = 2 – коэффициент усиления.

Получаем R3 = 10 кОм.

Резистор R2 подобран экспериментально, R2 = 220 Ом.

Характеристика усилителя, снятая экспериментально , показана на рисунке 3.18.



Рисунок 3.18 – Зависимость выходного напряжения от входного для усилителя с диодом и резистором.

Теперь исследуем результаты применения нелинейного усилителя с диодом Шотки и арсенид – галлиевым диодом с последовательно включённым сопротивлением.

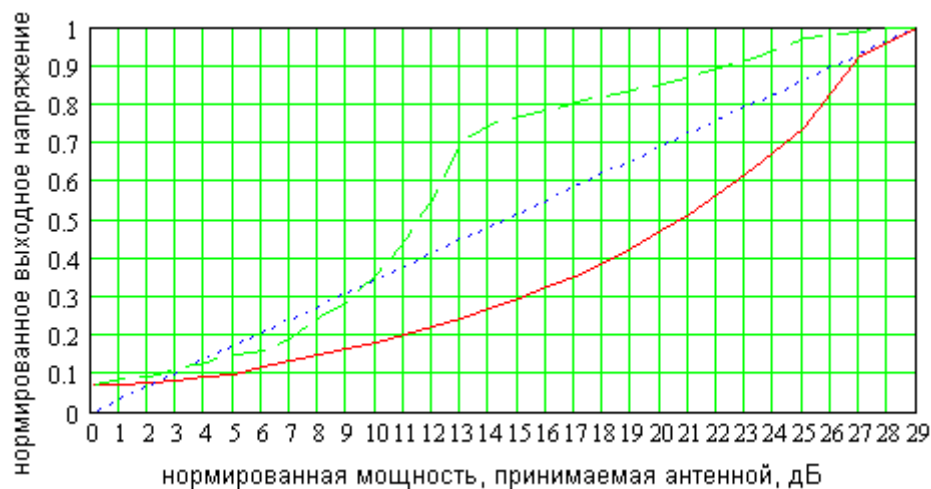


Рисунок 3.20 – Зависимость нормированного выходного напряжения от нормированной входной мощности для «идеального» усилителя (синяя прямая), усилителя с диодом Шотки (красная кривая) и усилителя с арсенид – галлиевым диодом.

Как видно из этого графика, характеристика системы при использовании усилителя с диодом Шотки более линейна. Следующий график – зависимость количества разрядов АЦП, приходящееся на один дециБел при отсутствии коррекции и с обоими видами коррекции, представлен на рисунке 3.21.



Рисунок 3.21 – Зависимость количества разрядов АЦП на один дБ входной мощности для системы без коррекции (красный график) , системы с коррекцией на диоде Шотки (чёрный график) и арсенид – галлиевом диоде (коричневый график).

Практически, высокую точность нельзя обеспечить при малом разрешении, поэтому система без коррекции будет плохо работать при высоком уровне входного сигнала , в

диапазоне до 10дБ. В то же время резкие перепады в количестве разрядов также отрицательно сказываются на точности, поэтому в данном применении имеет смысл использовать либо систему с коррекцией на арсенид – галлиевом диоде, либо комбинированную : первые 10 дБ оценивать по выходу усилителя с коррекцией на арсенид – галлиевом диоде, остальные – прямо с выхода модуля детектора .

Теперь оценим ошибку нелинейного усилителя при изменении температуры на 20 градусов для системы с коррекцией на арсенид – галлиевом диоде и при изменении температуры на 2 градуса для усилителя с диодом Шотки, график данной зависимости приведён на рисунке 3.22.

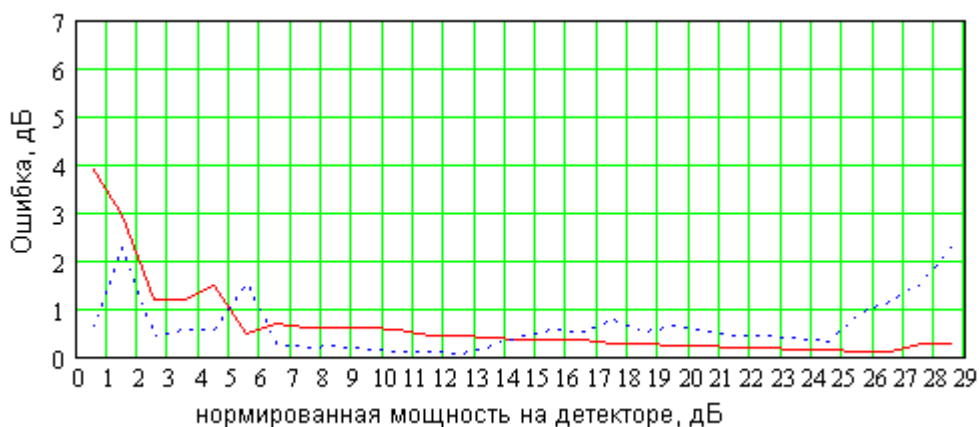


Рисунок 3.22 – Зависимость ошибки в дБ при изменении температуры для усилителя с арсенид - галлиевым диодом (синий график) и диодом Шотки (красный график).

Как видно из этого графика при изменении температуры на 20 градусов ошибка системы с коррекцией на арсенид – галлиевом диоде составит 2 дБ, при изменении температуры на 2 градуса коррекция с диодом Шотки обеспечит ошибку 4 дБ.

Проведя исследование систем коррекции можно сделать вывод, что наиболее подходит система с коррекцией на арсенид – галлиевом диоде с последовательно включённым сопротивлением, так как с её помощью можно добиться не только высокого разрешения, но и стабильности параметров. К сожалению, отсутствие данных по температурной стабильности модуля детектора не позволяют сделать вывод по температурной стабильности комплекса в целом, однако можно с уверенностью сказать, что разработанный модуль имеет высокую температурную стабильность.

Очевидно, что приёмник, описанный в предыдущей главе имеет большой динамический диапазон и большую линейность. Поэтому он и останется в качестве основного.

ВЫВОД

В процессе разработки дипломного проекта были изучены современные методы измерения диаграммы направленности антенн, предложена структурная схема комплекса, рассчитана принципиальная схема модуля, также были отмакетированы наиболее ответственные узлы разрабатываемого модуля, протестировано программное обеспечение контроллера, промоделирована работа схемы на температурную нестабильность.

По результатам проделанной работы можно сделать вывод о технических параметрах разработанного модуля :

- температурный диапазон (при однократной калибровке), °C +10..+30
- точность датчика уровня сигнала (при однократной калибровке), дБ ± 1
- точность установки местоположения платформы, градусов ± 0,2
- время снятия диаграммы в области ± 180 градусов, с 100
- совместимость формата данных комплекса MathCad
- максимальная мощность,
потребляемая от сети питания 220В 50Гц, Вт 80
- вид комплекса – стационарный.

Таким образом, можно сказать, что разработка данного проекта оказалась успешной и он может достойно соперничать со своими аналогами.

Перечень ссылок

1. Измеритель ДН антенн НТП «Тензор» // www.innov.ru/tenzor - kompl1.html
2. ATmega8535, AT8535L : preliminary data // www.Atmel.com/slup2108.pdf
3. Microchip MAX232: Technical Description // <http://www.maxim-ic.com/ic/rs232.html>
4. SMR capacitors: technical data // www.evov.com/SMR.pdf
5. IRLR/U2905 Power MOSFET // www.Irf.com/PD91334.pdf
6. ГОСТ 8.309 – 78. «Антенны остронаправленные. Методика выполнения измерений для определения параметров по полю в раскрыве».
7. ГОСТ 22261 – 94 «Измерители электрических и магнитных величин. Общие ТУ».
8. ГОСТ 24314-80 «Приборы электронные измерительные. Термины и определения, способы выражения и общие условия испытаний.»
9. ГОСТ 14014-84 «Информационно-измерительные системы. Требования к параметрам»
10. Сидоренко Д.А. Система позиционирования двухкоординатного поворотного устройства / А.А. Щекатурин, А.Ю. Никищенко // Проблемы повышения эффективности электромеханических преобразователей в электроэнергетических системах: Материалы междунар. научн.-техн. Конф., г. Севастополь: издательство СевНТУ, 2003.- 104 с., С.48-49.
11. Сидоренко Д.А. Система управления положением антенны для автоматизированного измерительного комплекса / А.А. Щекатурин, В.Н. Осаулко // Стан и розвиток ВМС Збройних Сил України на сучасному стані: Тези доповідей на III наук.- техн. Конф. – Севастополь: СВМІ ім. Н.С. Нахімова, 2003. – 248с., С.135-136.
12. Сидоренко Д.А. Система управления положением антенны для автоматизированного измерительного комплекса / А.А. Щекатурин, В.Н. Осаулко // Збірник наукових праць Вип. 1(4), Севастополь: Севастопольський ВМІ ім. П.С. Нахімова, 2004. – С. 189-191.
13. Сидоренко Д.А. Цифровой блок устройства измерения полного сопротивления / А.А. Щекатурин, А.Ю. Никищенко // 8-й Международный молодежный форум «Радиоэлектроника и молодежь XXI века» Сб. научных трудов. Ч.І. – Харьков: ХНУРЭ, 2004. – 500с.
14. Сидоренко Д.А. Микроконтроллерный блок автоматизированного измерительного комплекса для измерения характеристик вооружения ВМС Украины / А.А. Щекатурин, И.Л. Афонин // Сб. научн. Трудов, Включённый ВАК в перечень специальных изданий в области технич. наук , 2004.
15. С.А. Ясинский Прикладная «Золотая математика» и её приложения в электросвязи.- М.: Горячая линия – Телеком, 2004. – 239 с.
16. Г.Я. Мирский Микропроцессоры в измерительных приборах.- М.: Радио и связь, 1984. – 160 с.
17. Т. Williams EMC for product designers. – Oxford.: Planta Tree, 2001. – 360 с.
18. М. О’Hara EMC at component and PCB level. – Oxford.: Planta Tree, 2001. – 200 с.
19. Т. Мандел Разработка пользовательского интерфейса: пер. с англ.- М.: ДМК Пресс, 2001. – 416с.
20. Ускова О.Ф. Программирование алгоритмов обработки данных.- СПб.: БХВ-Петербург, 2003.-192с.
21. J. J. Labrosse MicroC/OS-II .-Lawrence, Kansas.: CMP Books, 2002.-620с.

Приложение А - программа микроконтроллера на языке ассемблера

```
*****
;*      Schekaturin's komplex host kontroller firmware      *
;*      written by Sidorenko D.A.                            *
;*      first edition:  18.02.2004                            *
;*      last changes:  27.04.2004  by Sidorenko D.A.         *
;*      controller: ATmega8535/AT90S8535, 4 MHz crystal      *
;*      working with Software by:  Eduard                    *
;*
;*      RS-232 speed: 19.2 kBaud                             *
;*      p.computer hardware resources used : com-port 1/2,usb(virtual com) *
;*
*****
.include "m8535def.inc"      ;standart definitions included
.include "AIKdef.inc";standart definitions included
```

```

;//////////////////// int vectors  //////////
.cseg
.org 0x00
    rjmp reset
.org $001      ;for azimuth and ugol
    rjmp int0_i

.org $002      ;for azimuth and ugol
    rjmp int1_i
.org adccaddr   ;adc interrupting
    rjmp adc_i

; .org $00B      ;not used
;    rjmp uart_i
.org $020
;//////////////////// main  //////////
reset:
;//////////////////// initialisation //////////
    ldi temp,0x00
    mov zero,temp      ;temp = 0
    ldi temp,10
    mov ten,temp      ;ten = 10
    mov rh,temp      ;rh = 10
    mov rl,r0      ;rl = 0
    mov ul,r0      ;ul = 0
    mov uh,temp      ;uh = 10
    mov rd,r0      ;rd = 0
    mov azl,zero      ;azl = 0
    mov azh,temp      ;azh = 10

    ldi temp,0x0f      ;temp = 0x0f
    out ddrb,temp      ;port b-out
    ldi temp,0xff
    out ddrc,temp      ;port c-out

```

Приложение А - продолжение

```

    out portd,temp;pull up resistors on
    out ddrd,zero; ;port d-in

    ldi r31,LOW(RAMEND)
    out spl,r31
    ldi r31,HIGH(RAMEND)
    out sph,r31      ;stek initc
    rcall pause10      ; 10 msec pause
    rcall uart_init      ; uart init
    ldi temp,64
    out mcucr,temp      ;sleep enable
    rcall adc_init      ;ADC init
    sei
    out udr,temp

;----- transmit memory -----
    ldi temp,CHAR_A

```

```

        sts T_A_RAM,temp
        ldi temp,CHAR_K
        sts T_K_RAM,temp
;----- initialisation complete -----
;----- main loop -----
main_loop:
        ldi temp,0x0
        sts H1_RAM,temp
        ldi temp,0x0
        sts L1_RAM,temp
        ldi temp,0x0
        sts H2_RAM,temp
        ldi temp,0x0
        sts L2_RAM,temp
        ldi temp,0x0
        sts H3_RAM,temp
        ldi temp,0x0
        sts L3_RAM,temp
        ldi temp,0x0
        sts H4_RAM,temp
        ldi temp,0x0
        sts L4_RAM,temp
        ldi temp,0x0
        sts P_ERR_RAM,temp

        rcall receive_p
        rcall move_n_steps
        rcall get_set_f
        rcall transmit_f
rjmp main_loop
;----- functions ----
;*****pause0.5mSECONDS
pause05:
        ldi temp,5

```

Приложение А - продолжение

```

m0105:ldi temp1,130
m10105:dec temp1
brne m10105
dec temp
brne m0105

ret
;*****pause1mSECONDS
pause1:
        ldi temp,10
        m01:ldi temp1,130
        m101:dec temp1
        brne m101
        dec temp
        brne m01

ret
;*****pause5mSECONDS

```

```

pause5:
    ldi temp,50
    m01h5:ldi temp1,130
    m101h5:dec temp1
    brne m101h5
    dec temp
    brne m01h5
ret
;*****pause2mSECONDS
pause2:
    ldi temp,20
    m02:ldi temp1,130
    m102:dec temp1
    brne m102
    dec temp
    brne m02
ret
;*****pause10mSECONDS
pause10:
    ldi temp,100
    m:ldi temp1,130
    m1:dec temp1
    brne m1
    dec temp
    brne m
ret
Pause30:;*****pause30mSECONDS
    ldi temp2,3
    m1p_ch12:rcall pause10
    dec temp2
    brne m1p_ch12
ret
Pause20:;*****pause20mSECONDS
    ldi temp2,2

```

Приложение А - продолжение

```

    m1p11_ch12:rcall pause10
    dec temp2
    brne m1p_ch12
ret
pause250:;*****pause250mSECONDS
    ldi temp2,25
    m1p_ch121:rcall pause10
    dec temp2
    brne m1p_ch121
ret

pause1000:;*****p1000mSECONDS
    ldi temp2,100
    pause101:
    ldi temp,100
    mx:ldi temp1,130

```

```

        m1x:dec temp1
        brne m1x
        dec temp
        brne mx
        dec temp2
        brne pause101
ret
adc_init:;*****adc_init
        ldi temp,0x86
        out ADCSRA,temp
ret
adc_off:;*****adc off
        ldi temp,0
        out ADCSRA,temp
ret
;*****adc0*****
adc0:
        out admux,zero           ;channel 0
        rcall pause05
        ldi temp,64
        out mcucr,temp           ;idle
        ldi temp,0x8e
        out ADCSRA,temp
        sbi ADCSRA,6             ;start
        sleep
        ldi temp,0x86
        out ADCSRA,temp
        in temp,adcl
        in temp1,adch
ret    ;adc0
uart_init:;*****uart_init:
        ldi temp,12      ;19200
        out UBRRL,temp    ;speed 19200 bod
        ldi temp,24

```

Приложение А - продолжение

```

        out UCSRB,temp    ;enable transmit and interrapping txc
        mov flag,zero
        nop
        ret
step_left1:;*****step_left1****
        ldi temp,0x29
        out portc,temp
        rcall pause20
        ldi temp,0x2a
        out portc,temp
        rcall pause20
        ldi temp,0x26
        out portc,temp
        rcall pause20
        ldi temp,0x25
        out portc,temp

```

```

        rcall pause20
        ldi temp,0x0
        out portc,temp
ret
step_right1:;*****1step_right1****
        ldi temp,0x25
        out portc,temp
        rcall pause20
        ldi temp,0x26
        out portc,temp
        rcall pause20
        ldi temp,0x2a
        out portc,temp
        rcall pause20
        ldi temp,0x29
        out portc,temp
        rcall pause20
        ldi temp,0x0
        out portc,temp
ret
step_up1:;*****step_up1****
        ldi temp,0x15
        out portc,temp
        rcall pause20
        ldi temp,0x16
        out portc,temp
        rcall pause20
        ldi temp,0x1a
        out portc,temp
        rcall pause20
        ldi temp,0x19
        out portc,temp
        rcall pause20
        ldi temp,0x0

```

Приложение А - продолжение

```

        out portc,temp
ret
step_down1:;*****step_down1****
        ldi temp,0x19
        out portc,temp
        rcall pause20
        ldi temp,0x1a
        out portc,temp
        rcall pause20
        ldi temp,0x16
        out portc,temp
        rcall pause20
        ldi temp,0x15
        out portc,temp
        rcall pause20
        ldi temp,0x0

```

```

        out portc,temp
ret

;*****
;*      perevod
;*****
;*****
perevod:
    mov byte1,r0
    mov byte2,r0
    mov byte3,r0
    mov counter,r0
    dec byte1
    dec counter
perevod1:
    inc counter
    inc byte1
    ldi temp,10
    eor temp,byte1
    brne perevod2
    inc byte2
    mov byte1,r0
perevod2:
    ldi temp,10
    eor temp,byte2
    brne perevod3
    inc byte3
    mov byte2,r0
perevod3:
    mov temp,byte
    eor temp,counter
    brne perevod1
    nop
    ldi temp,0x30

```

Приложение А - продолжение

```

    or byte1,temp
    or byte2,temp
    or byte3,temp
    nop
ret    ;perevod
;////////// move_n_steps //////////
move_n_steps:
    lds counter,Number_of_steps_RAM ;counter = Number_of_steps_RAM
    lds rd,Move_dir_RAM             ;rd = Move_dir_RAM
    cp counter,zero                  ;
    breq move_n_steps100            ; if (counter == no moveing)goto (
move_n_steps100:)
    ldi temp,STEP_UP
    cp temp,rd                      ;rd == STEP_UP ?
    brne move_n_steps101            ; if (rd != STEP_UP)goto (
move_n_steps101:)

```

```

        move_n_steps102:
            dec counter
            rcall step_up1
            tst counter                ;if (counter != 0) goto(move_n_steps102:)
            brne move_n_steps102
    move_n_steps101:
        ldi temp,STEP_DOWN
        cp temp,rd
        brne move_n_steps103          ; if (rd != STEP_DOWN)goto (
move_n_steps103:)
        move_n_steps104:
            dec counter
            rcall step_down1
            tst counter                ;if (counter != 0) goto(move_n_steps104:)
            brne move_n_steps104

```

```

    move_n_steps103:
        ldi temp,STEP_LEFT
        cp temp,rd
        brne move_n_steps105          ; if (rd != STEP_LEFT)goto (
move_n_steps105:)
        move_n_steps106:
            dec counter
            rcall step_left1
            tst counter                ;if (counter != 0) goto(move_n_steps106:)
            brne move_n_steps106

```

```

    move_n_steps105:

```

```

        ldi temp,STEP_RIGHT
        cp temp,rd

```

Приложение А - продолжение

```

        brne move_n_steps107          ; if (rd != STEP_RIGHT)goto (
move_n_steps107:)
        move_n_steps108:
            dec counter
            rcall step_right1
            tst counter                ;if (counter != 0) goto(move_n_steps108:)
            brne move_n_steps108

```

```

        move_n_steps107:
    move_n_steps100:
        ret        ;move_n_steps
;***** get_set *****
;*      get/set from Get_set_RAM
;*****
get_set_f:

```

```

lds rd,Get_set_RAM      ;rd = Get_set_RAM
cp rd,zero              ;
    breq get_set_f100    ; if (counter == no moveing)goto ( move_n_steps100:)
    ldi temp,GET_U1
    cp temp,rd           ;rd == GET_U1 ?
    brne get_set_f101    ; if (rd != GET_U1)goto ( move_n_steps101:)
        rcall adc0
        sts L1_RAM,temp
        sts H1_RAM,temp1
    get_set_f101:
    ldi temp,SET_FREQ
    cp temp,rd
    brne get_set_f102    ; if (rd != SET_FREQ)goto ( move_n_steps103:)
        rcall set_generator_freq
    get_set_f102:
    ldi temp,SET_VERT_POL
    cp temp,rd
    brne get_set_f103    ; if (rd != SET_VERT_POL)goto (
move_n_steps105:)
        rcall set_generator_vert_pol
    get_set_f103:
    ldi temp,SET_HOR_POL
    cp temp,rd
    brne get_set_f104    ; if (rd != SET_HOR_POL)goto (
move_n_steps107:)
        rcall set_generator_hor_pol
    get_set_f104:
    ldi temp,GET_UGOL
    cp temp,rd
    brne get_set_f105    ; if (rd != GET_UGOL)goto ( move_n_steps107:)
        rcall get_platform_ugol
        sts L1_RAM,ul
        sts H1_RAM,uh
    get_set_f105:

```

Приложение А - продолжение

```

    ldi temp,GET_AZIMUT
    cp temp,rd
    brne get_set_f106    ; if (rd != GET_AZIMUT)goto ( move_n_steps107:)
        rcall get_platform_azimut
        sts L1_RAM,azl
        sts H1_RAM,azh
    get_set_f106:
    ldi temp,EQ_TEST
    cp temp,rd
    brne get_set_f107    ; if (rd != EQ_TEST)goto ( move_n_steps107:)
        rcall get_eq_test
    get_set_f107:
    ldi temp,GET_U1_U
    cp temp,rd
    brne get_set_f108    ; if (rd != GET_U1_U)goto ( move_n_steps108:)
        rcall get_u1_u_func

```

```

        get_set_f108:
        ldi temp,GET_UGOL_U
        cp temp,rd
        brne get_set_f109          ; if (rd != GET_UGOL_U)goto ( move_n_steps109:)
            rcall get_ugol_u_func
        get_set_f109:
        ldi temp,GET_AZIMUT_U
        cp temp,rd
        brne get_set_f110          ; if (rd != GET_AZIMUT_U)goto (
move_n_steps110:)
            rcall get_azimut_u_func
        get_set_f110:
        ldi temp,SET_LEVEL
        cp temp,rd
        brne get_set_f111          ; if (rd != SET_LEVEL)goto ( move_n_steps111:)
            rcall set_level_func
        get_set_f111:
get_set_f100:
ret      ;Get_set
;//////////          set_generator_freq      //////////
set_generator_freq:
ret      ;set_generator_freq
;//////////          set_generator_vert_pol      //////////
set_generator_vert_pol:
ret      ;set_generator_vert_pol
;//////////          set_generator_hor_pol      //////////
set_generator_hor_pol:
ret      ;set_generator_hor_pol
;//////////          get_platform_ugol      //////////
get_platform_ugol:
        ldi temp,0
        out tccr1a,temp;counter -internal
        ldi temp,192
        out tccr1b,temp;setting count

```

Приложение А - продолжение

```

        ldi temp,0
        out tcnt1h,temp;
        ldi temp,0
        out tcnt1l,temp;counter :=0

        ldi temp,67
        out mcucr,temp;enable int 1,idle,sleep
        ldi temp,0
        out timsk,temp;counters int disabled
        ldi temp,64
        out GICR,temp
        nop
        nop
        nop
        ldi temp1,192
        ldi temp,66

```

```

sei
mov rez1,ten
ldi symbol,193
sleep
urw3m1:
dec rez1
brne urw3m1
out tccr1b,symbol
;start counting
out mcucr,temp;enable int 1 down,idle,sleep
ldi temp1,192
sleep;stop counting
in ul,tcnt1l
in uh,tcnt1h
ldi temp,0
out GICR,temp
ldi temp,64
out mcucr,temp;enable int 1,idle,sleep
ret ;get_platform_ugol
;////////// get_platform_azimut //////////
get_platform_azimut:
ldi temp,0
out tccr1a,temp ;counter -internal
ldi temp,192
out tccr1b,temp ;setting count
ldi temp,0
out tcnt1h,temp ;
ldi temp,0
out tcnt1l,temp ;counter :=0

ldi temp,77
out mcucr,temp ;enable int 1,idle,sleep
ldi temp,0
out tmsk,temp ;counters int disabled

```

Приложение А - продолжение

```

ldi temp,128
ldi temp1,64
out GICR,temp
nop
nop
nop
ldi temp1,192
ldi temp,72
ldi symbol,193
mov rez1,ten
sei
sleep
aurw3m1:
dec rez1
brne aurw3m1
out tccr1b,symbol

```

```

;start counting
out mcucr,temp;enable int 1 down,idle,sleep
ldi temp1,192
sleep;stop counting
in azl,tent1l
in azh,tent1h

ldi temp,0
out GICR,temp
ldi temp,64
out mcucr,temp;enable int 1,idle,sleep
ret ;get_platform_azimut
;***** get_eq_test *****
;* result ul: if ul==1+2+4+8 then allthing works
;* else if u1==15-1 or u1 == 15-4 then azimuth not working
;* if u1==15-2 or u1 == 15-8 then ugol not working
;*****
get_eq_test:
ldi temp,0
mov ul,temp
ldi byte,1
ldi adress,2
ldi symbol,4
ldi temp2,8

ldi temp,100
mpr:ldi temp1,130
m1pr:sbic pind,2
or ul,byte
sbic pind,3
or ul,adress
sbis pind,2
or ul,symbol
sbis pind,3

```

Приложение А - продолжение

```

or ul,temp2

dec temp1
brne m1pr
dec temp
brne mpr
sts L1_RAM,ul
ret ;get_eq_test
;//////////////////// rcall get_u1_u_func //////////////////
get_u1_u_func:
rcall adc0
sts L1_RAM,temp
sts H1_RAM,temp1
rcall pause5
rcall adc0
sts L2_RAM,temp

```

```

        sts H2_RAM,temp1
        rcall pause5
        rcall adc0
        sts L3_RAM,temp
        sts H3_RAM,temp1
        rcall pause5
        rcall adc0
        sts L4_RAM,temp
        sts H4_RAM,temp1
ret      ;get_u1_u_func
;//////////      get_ugol_u_func      //////////
get_ugol_u_func:
        rcall get_platform_ugol
        sts L1_RAM,ul
        sts H1_RAM,uH
        rcall get_platform_ugol
        sts L2_RAM,ul
        sts H2_RAM,uH
        rcall get_platform_ugol
        sts L3_RAM,ul
        sts H3_RAM,uH
        rcall get_platform_ugol
        sts L4_RAM,ul
        sts H4_RAM,uH

ret      ;get_ugol_u_func
;//////////      get_azimut_u_func      //////////
get_azimut_u_func:
        rcall get_platform_azimut
        sts L1_RAM,azl
        sts H1_RAM,azh
        rcall get_platform_azimut
        sts L2_RAM,azl
        sts H2_RAM,azh

```

Приложение А - продолжение

```

        rcall get_platform_azimut
        sts L3_RAM,azl
        sts H3_RAM,azh
        rcall get_platform_azimut
        sts L4_RAM,azl
        sts H4_RAM,azh
ret      ;get_azimut_u_func
;//////////      set_level_func      //////////
set_level_func:
ret      ;set_level_func
;//////////      interruptings      //////////
;***** reseive_p *****
;*reseive packet from PC :
;*      0      = 'A'= 0x41
;*      1      = 'K'= 0x4b
;*      2      P_nomber - packet nomber

```

```

;*      3      Move_dir - move direction
;*      4      number of steps
;*      5      Get_set
;*      6      Data_H
;*      7      Data_L
;*
;*
;*      resources used:
;*          functions:      PC_byte_wait
;*          registers:      temp,rd
;*                          RAM:0x0060..0x006F-receive pocket
;*      changes:RAM:0x0060..0x006F-receive pocket
;*              RAM:0x0070..0x007F-transmit pocket(P_ERR_RAM)
;*****
receive_p:
    in rd,udr                ;delay receive flag
; byte 0- 'A'
    rcall PC_byte_wait      ;receive data
    sts R_A_RAM,rd          ;R_A_RAM = received data
    ldi temp,CHAR_A
    cp temp,rd              ;receive data == 'A' ?
    brne receive_p         ; if (rd != 'A')goto (receive_p:)
; byte 1- 'K'
    rcall PC_byte_wait      ;receive data
    sts R_K_RAM,rd          ;R_K_RAM = received data
    ldi temp,CHAR_K
    cp temp,rd              ;receive data == 'K' ?
    brne receive_p         ; if (rd != 'K')goto ( receive_p:)
    sts P_ERR_RAM,zero      ;P_ERR = 0
; byte 2- P_number
    rcall PC_byte_wait      ;receive data
    sts R_P_number_RAM,rd   ;R_P_number_RAM = received data
; byte 3- Move_dir
    rcall PC_byte_wait      ;receive data
    sts Move_dir_RAM,rd     ;Move_dir_RAM = received data

```

Приложение А - продолжение

```

; byte 4- Number_of_steps
    rcall PC_byte_wait      ;receive data
    sts Number_of_steps_RAM,rd ;Number_of_steps_RAM = received data
; byte 5- Get_set
    rcall PC_byte_wait      ;receive data
    sts Get_set_RAM,rd      ;Get_set_RAM = received data
; byte 6- Data_H
    rcall PC_byte_wait      ;receive data
    sts Data_H_RAM,rd       ;Data_L_RAM = received data
; byte 7- Data_L
    rcall PC_byte_wait      ;receive data
    sts Data_L_RAM,rd       ;Data_H_RAM_RAM = received data
; byte 8- c_sum
    rcall PC_byte_wait      ;receive data
    sts c_sum_ram,rd        ;Data_H_RAM_RAM = received data
ldi temp,0xff

```

```

lds tr,R_A_RAM                ; calculate c_sum
eor temp,tr
lds tr,R_K_RAM
eor temp,tr
lds tr,R_P_nomber_RAM
eor temp,tr
lds tr,Move_dir_RAM
eor temp,tr
lds tr,Number_of_steps_RAM
eor temp,tr
lds tr,Get_set_RAM
eor temp,tr
lds tr,Data_H_RAM
eor temp,tr
lds tr,Data_L_RAM
eor temp,tr
lds temp1,c_sum_ram
    cp temp1,temp                ;receive data with no err ?
    breq reseive_p111            ; if (c_sum == transmitted c_sum)goto ( reseive_p111:)
        ldi temp,ERR
        sts P_ERR_RAM,temp
    reseive_p111:
;if received the packet with the same number then transmit packet once more
lds temp1,T_P_nomber_RAM
lds temp,R_P_nomber_RAM
cp temp1,temp
    brne reseive_p112            ; if (T_P_nomber_RAM != R_P_nomber_RAM )goto (
reseive_p112:)
        rcall transmit_f        ; else
        rjmp receive_p
    reseive_p112:
ret        ;reseive_p

```

Приложение А - продолжение

```

;***** PC_byte_wait *****
;
;* wait byte from PC and loads it to rd
;* resources : UART :no interruptings!
;*****
PC_byte_wait:
    in temp,UCSRA
    sbrs temp,7
    rjmp PC_byte_wait    ;
    in rd,udr            ;
ret        ;PC_byte_wait
;***** transmit_f *****
;
;* transmits T_PACKET (RAM0x070-0x07b)
;* resources :UART-no interruptings
;* mov R_P_nomber_RAM to T_P_nomber_RAM
;* calculates c_sum
;*****
;

```

```

transmit_f:
; byte 0- 'A'
    lds tr,T_A_RAM
    rcall transmit_byte    ;transmit data
; byte 1- 'K'
    lds tr,T_K_RAM
    rcall transmit_byte    ;transmit data
; byte 2- T_P_number
    lds tr,R_P_number_RAM
    sts T_P_number_RAM,tr
    nop
    lds tr,T_P_number_RAM
    rcall transmit_byte    ;transmit data
; byte 3- H1_RAM
    lds tr,H1_RAM
    rcall transmit_byte    ;transmit data
; byte 4- L1_RAM
    lds tr,L1_RAM
    rcall transmit_byte    ;transmit data
; byte 5- H2_RAM
    lds tr,H2_RAM
    rcall transmit_byte    ;transmit data
; byte 6- L2_RAM
    lds tr,L2_RAM
    rcall transmit_byte    ;transmit data
; byte 7- H3_RAM
    lds tr,H3_RAM
    rcall transmit_byte    ;transmit data
; byte 8- L3_RAM
    lds tr,L3_RAM
    rcall transmit_byte    ;transmit data
; byte 9- H4_RAM
    lds tr,H4_RAM
    rcall transmit_byte    ;transmit data

```

Приложение А - продолжение

```

; byte 10- L4_RAM
    lds tr,L4_RAM
    rcall transmit_byte    ;transmit data
; byte 11- P_ERR_RAM
    lds tr,P_ERR_RAM
    rcall transmit_byte    ;transmit data
; byte 12- c_sum          ; calculate and send c_sum
    ldi temp,0xff
    lds tr,T_A_RAM
    eor temp,tr
    lds tr,T_K_RAM
    eor temp,tr
    lds tr,T_P_number_RAM
    eor temp,tr
    lds tr,H1_RAM
    eor temp,tr

```

```

    lds tr,L1_RAM
    eor temp,tr
    lds tr,H2_RAM
    eor temp,tr
    lds tr,L2_RAM
    eor temp,tr
    lds tr,H3_RAM
    eor temp,tr
    lds tr,L3_RAM
    eor temp,tr
    lds tr,H4_RAM
    eor temp,tr
    lds tr,L4_RAM
    eor temp,tr
    lds tr,P_ERR_RAM
    eor temp,tr
    mov tr,temp
    sts T_c_sum_ram,temp
    rcall transmit_byte    ;transmit data
ret    ;transmit_f
;***** transmit_byte *****
transmit_byte:
    in temp,UCSRA
    sbrs temp,6
    rjmp transmit_byte    ;
    out udr,tr
transmit_byte2:
    in temp,UCSRA
    sbrs temp,6
    rjmp transmit_byte2    ;
    rcall pause05          ;wait 05 ms
ret    ;transmit_byte

```

Приложение А - продолжение

```

;----- interrupting vectors -----
adc_i:
    reti
;uart_i:
;in rd,udr
;reti
int1_i:
int0_i:
    out tccr1b,temp1
reti

```

Приложение Б – Макроопределения для программы микроконтроллера на языке ассемблера

;AIK definitions file

```
.def zero=    r0      ;= 0
.def ten=     r1      ;= 10
.def byte1=   r2
.def byte2=   r3
.def byte3=   r4
.def tr=      r5
.def rd=      r6
.def flag=    r7
.def rh=      r8
.def rl=      r9
.def uh=      r10
.def ul=      r11
.def azh=     r12
.def azl=     r13
.def rez1=    r14
.def rez2=    r15
```

```

.def temp=    r16
.def temp1=   r17
.def temp2=   r18
.def counter= r19
.def lcounter= r20
.def hcounter= r21
.def status=  r22
.def symbol=  r23
.def adress=  r24

.def byte=    r25
;////////// constants //////////

;////////// pause val //////////
.equ pause_val=    79    ;10ms
;////////// receive pocket constants \\\\\\\\\\\\\\\\\\\\\\\
;////////// bit numbers  //////////
.equ R_A=          0x00  ;0
.equ R_K=          0x01  ;1
.equ R_P_number=   0x02
.equ Move_dir=     0x03
.equ Number_of_steps= 0x04
.equ Get_set=      0x05
.equ Data_L=       0x06
.equ Data_H=       0x07

```

Приложение Б - продолжение

```

;    /// location in RAM
.equ R_A_RAM=      0x060
.equ R_K_RAM=      0x061
.equ R_P_number_RAM= 0x062
.equ Move_dir_RAM= 0x063
.equ Number_of_steps_RAM=0x064
.equ Get_set_RAM= 0x065
.equ Data_L_RAM= 0x066
.equ Data_H_RAM= 0x067
.equ c_sum_ram=    0x068
;////////// transmit pocket constants \\\\\\\\\\\\\\\\\\\\\\\
;////////// bit numbers  //////////
;    .equ A=          0x00  ;0
;    .equ B=          0x01  ;1
;    .equ P_number=   0x02
.equ L1=    0x03
.equ H1=    0x04
.equ L2=    0x05
.equ H2=    0x06
.equ L3=    0x07

```

```

.equ H3=      0x08
.equ L4=      0x09
.equ H4=      0x0A
.equ P_ERR= 0x0B
; /// location in RAM
    .equ T_A_RAM=      0x070
    .equ T_K_RAM=      0x071
    .equ T_P_nomber_RAM=0x072
    .equ L1_RAM=      0x073
    .equ H1_RAM=      0x074
    .equ L2_RAM=      0x075
    .equ H2_RAM=      0x076
    .equ L3_RAM=      0x077
    .equ H3_RAM=      0x078
    .equ L4_RAM=      0x079
    .equ H4_RAM=      0x07A
    .equ P_ERR_RAM= 0x07B
    .equ T_c_sum_ram=0x07c
;///////// Packets constants: ///////////
    .equ CHAR_A=      0x41  ;'A'
    .equ CHAR_K=      0x4b  ;'K'
    .equ P_LENGTH=    4
    .equ ERR=      254
    .equ OK=      0
;///////// move constants  ///////////
    .equ STEP_UP=      1
    .equ STEP_DOWN=    2
    .equ STEP_LEFT=    3
    .equ STEP_RIGHT=4
;///////// get_set constants  ///////////

```

Приложение Б - продолжение

```

.equ GET_U1=      1
.equ SET_FREQ=    2
.equ SET_VERT_POL= 3
.equ SET_HOR_POL= 4
.equ GET_UGOL=    5
.equ GET_AZIMUT=6
.equ EQ_TEST=     7
.equ SET_LEVEL=   8
.equ GET_U1_U=    9
.equ GET_UGOL_U= 10
.equ GET_AZIMUT_U=11

```

Приложение В – расчет цепи фильтра компаратора датчика положения платформы в MathCad 2001

Примим защитный резистор номиналом в 100 кОм,

$$R_{\phi} := 10^5 \text{ Ом};$$

Передаточная функция цепи фильтра имеет вид:

$$K(f) := \frac{\frac{1}{1i \cdot 2 \cdot \pi \cdot f \cdot C}}{\frac{1}{1i \cdot 2 \cdot \pi \cdot f \cdot C} + R_{\phi}}$$

Исследуем влияние старения пассивных элементов на точность датчика. В процессе старения элементы могут изменить свои значения на 0,25 от указанной на них точности, это значит, что при точности конденсатора 5% изменение ёмкости не превысит 1,25%. Резистор имеет смысл поставить с точностью 1%, тогда изменение его сопротивления составит 0,25%, т.е. в 5 раз меньше, чем у конденсатора.

Теперь оценим влияние изменения смещения операционного усилителя. По документации, изменение смещения не может превысить 0,5 мВ (в течении первых 10 лет). Найдём смещение платформы, вызванное этим изменением :

$U_m := 5$ В - амплитуда сигнала с датчика;

$U_{sm} := 0.5 \cdot 10^{-3}$ В - максимальное смещение ОУ;

$dP := \frac{U_{sm}}{U_m \cdot 2 \cdot \pi}$ $dP = 1.592 \times 10^{-5}$ - максимальное смещение фазы;

Уточним, что 1 шаг платформы соответствует 0,4 градуса, следовательно точность установки - 0,2 градуса. Найдём требуемую точность датчика:

$dg := 0.2$ точность установки в градусах,

$d_{ust} := \frac{dg}{360}$ $d_{ust} = 5.556 \times 10^{-4}$ -точность установки платформы.

Из приведённых выкладок видно, что изменение смещения ОУ не может вызвать ошибки, в то время как изменение параметров резистора и конденсатора может вызвать ошибку.

Найдём значение ёмкости, при изменении значения которой на 2% (что возможно при применении существующих конденсаторов с точностью 5%) не произойдёт ошибки позиционирования.

Это значит, что фаза в этой цепочке не должна сдвигаться более, чем на 0,2 точности установки, т.е. на 0,0001.

$f := 400$ Гц - опорная частота датчика;

$C := 1 \cdot 10^{-12}, 2 \cdot 10^{-12} \dots 10^{-9}$ задаём диапазон исследования

$dC := 0.02$ -задаём максимальное изменение ёмкости

$$Y(C) := \frac{\frac{1}{i \cdot 2 \cdot \pi \cdot f \cdot C \cdot (1 + dC)}}{\frac{1}{i \cdot 2 \cdot \pi \cdot f \cdot C \cdot (1 + dC)} + R\phi}$$
 -передаточная функция при изменении ёмкости

Приложение В - продолжение

$$Y1(C) := \frac{\frac{1}{i \cdot 2 \cdot \pi \cdot f \cdot C}}{\frac{1}{i \cdot 2 \cdot \pi \cdot f \cdot C} + R\phi} \quad \text{передаточная функция при номинальной ёмкости}$$

$$\Theta(C) := \left| \frac{(\arg(Y(C)) - \arg(Y1(C)))}{2 \cdot \pi} \right| \quad \text{- относительное отклонение ФЧХ}$$

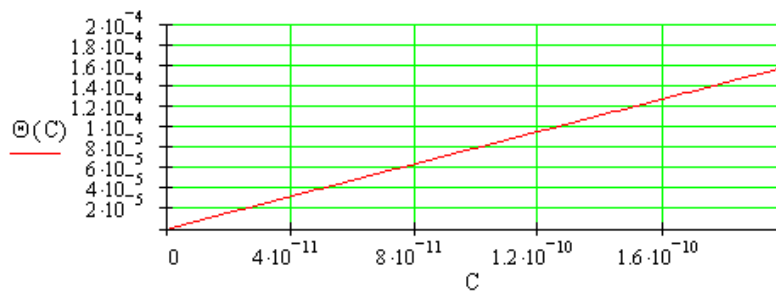


Рисунок 1 - график зависимости отклонений показаний датчика от номинала ёмкости конденсатора фильтра.

$$d\Theta(C) := \arg(Y1(C) \cdot \Theta(C))$$

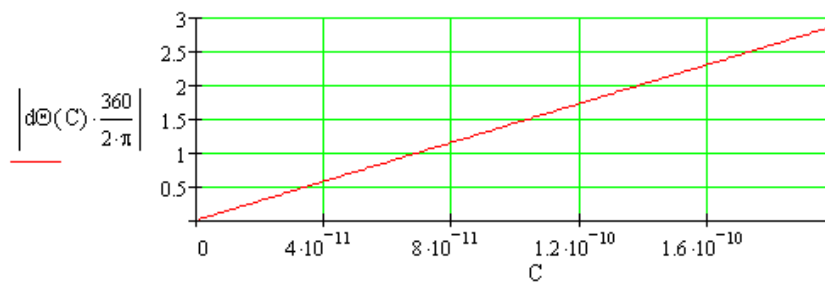


Рисунок 2 - график зависимости смещения фазы с датчика от номинала ёмкости конденсатора фильтра.

$$d_Cdat := 0.8 \cdot 10^{-4}$$

$$C := 10^{-10}$$

Оценим влияние резистора:

$$f := 400 \quad \text{Гц - опорная частота датчика;}$$

$$dR := 0.0025 \quad \text{- задаём максимальное изменение сопротивления}$$

Приложение В - продолжение

$$Y := \frac{\frac{1}{i \cdot 2 \cdot \pi \cdot f \cdot C}}{\frac{1}{i \cdot 2 \cdot \pi \cdot f \cdot C} + R\phi \cdot (1 + dR)} \quad \text{передаточная функция при изменении сопротивления}$$

$$Y1 := \frac{\frac{1}{i \cdot 2 \cdot \pi \cdot f \cdot C}}{\frac{1}{i \cdot 2 \cdot \pi \cdot f \cdot C} + R\phi} \quad \begin{array}{l} \text{передаточная функция при номинальном} \\ \text{сопротивлении} \end{array}$$

$$d\Theta(C) := \arg(Y1(C))$$

$$\Theta R := \left| \frac{(\arg(Y) - \arg(Y1))}{2 \cdot \pi} \right|$$

$$\Theta R = 9.994 \times 10^{-6} \quad \text{относительное отклонение ФЧХ}$$

Итого получаем, что точность датчика в результате старения не может измениться более, чем на

$$d_dat := dP + d_Cdat + \Theta R$$

$$d_dat = 1.059 \times 10^{-4}$$

Теперь проверим температурную стабильность датчика. По документации на конденсаторы, уход ёмкости от начального значения в диапазоне от -60 до +125 градусов не превышает 2%. Это означает, что температурная нестабильность равна уходу ёмкости в результате старения и равна 0,0001. Дрейфом операционного усилителя, как показано выше, можно пренебречь.

Найдём максимальное отклонение показаний датчика в результате старения и температурной нестабильности:

$$D := d_dat + d_Cdat \quad D = 1.859 \times 10^{-4}$$

Оценим целесообразность установки такого датчика. Найдём частоту среза RC - фильтра.

$$F_{ср} := \frac{0.16}{R\phi \cdot C} \quad F_{ср} = 1.6 \times 10^4 \text{ Гц.}$$

Следовательно, установка такого фильтра целесообразна, так как частота единичного усиления значительно выше (1,5 МГц по документации).

Вывод :

1. Установка фильтра целесообразна.
2. Параметры элементов фильтра :

$$R\phi = 1 \times 10^5 \quad \text{Ом ;}$$

$$C = 1 \times 10^{-10} \quad \text{Ф .}$$

3. Отклонение показаний датчика, вызванные температурной и временной нестабильностью, не хуже

$$D = 1.859 \times 10^{-4} \quad .$$

Приложение Г – программа контроллера приёмника

```
// DfilterFW
// written by Sidorenko D.A. KB SO STEL
// includes
#include <msp430x14x.h>
#include <stdio.h>

// C O N S T A N T S :
//----- define port function -----
// P5OUT
#define P5flag 0X01
#define P5_MOSFET_ON1 0X02
#define P5_MOSFET_ON2 0X04
//P6OUT:
#define FADC_in 0X01
#define KPeakDetADC_in 0X02

//----- define port state -----
#define OUT_DIRECTION 0xff
#define IN_DIRECTION 0x00

//----- receive and transmit constants-----

//----- the program use functions: -----
void controller_init(void);           //initialisation of the
controller
void filter_init();

unsigned ufunc(unsigned x);           //to define voltage with adc
unsigned ufuncp(unsigned x);         //to define voltage with adc-
precision
unsigned ADC(unsigned port_pin);
void delay(unsigned t); //t-в мкс

// ADCTF[ADC]=16*lg(xc)
unsigned int ADC32[32] =
{0,382,430,458,478,494,507,517,527,535,542,549,555,560,565,570,575,579,583,58
7,590,594,597,600,603,606,608,611,614,616,618,621};
unsigned t,o1,o2,o3,KU1_ON,KU2_ON,ADCM,dB,counter,dBsum;
//----- - MAIN PROGRAM - -----
void main(void)
{ register unsigned int temp1;
  INT16U ADCR,ADCD;           // ADC result, ADC in decibels
  INT16U temp2;
  controller_init();
  filter_init();
  // ADC init
  ADC12CTL0 = ADC12ON;
  ADC12MCTL0 = /*port_pin*/0 + 0xB0;           // reference select,
channel sel.
  // Timer init
  WDTCTL = WDTPW +WDTHOLD;           // Stop WDT
  TACTL = TASSEL1 + TACLR;           // SMCLK, Clear Tar
  CCR0 = 1000; /*512-1; */           // PWM Period
  CCTL1 = OUTMOD_7;           // CCR1 reset/set
  CCR1 =100/*384*/;           // CCR1 PWM duty cycle
  CCTL2 = OUTMOD_7;           // CCR2 reset/set
  CCR2 = 128;           // CCR2 PWM duty cycle
  P1DIR |= 0x0C;           // P1.2 and P1.3 output
  P1SEL |= 0x0C;           // P1.2 and P1.3 TA1/2 otions
  TACTL |= MC0;           // Start Timer_A in up mode
```

Приложение Г - продолжение

```

//основной цикл
temp1=0;
KU1_ON=0; // + 18 dB
KU2_ON=0; // - 18 dB
counter = 0;
dBsum = 0;
for (;;)
{
    ADCR = ADC(1);          // get ADC value from qasypeak
    ADCR = 4095 - ADCR;     // invert
    ADCD=0;
    ADCM=ADCR;
    P5OUT |= P5flag;
    dB = 0;
    if (ADCR > 31)
    {
        while (ADCR > 31)
        {
            ADCR = ADCR >> 1;
            dB += 48; // - 3 dB
        }
    }
    dB += ADC32[ADCR];
    if(KU1_ON) dB -= 259; // - 18 dB
    if(KU2_ON) dB += 237; // + 18 dB
    // PWM out
    dB -= 160;
    dBsum += dB;
    counter++;
    if (counter > 63)
    {
        CCR1= dBsum >> 6; //
        counter = 0;
        dBsum = 0;
    }
    // anouth amplification?
    if (KU2_ON) // if 1/100 is not anouth,
    {
        if (ADCM<30) //
        {
            KU2_ON=0; // * 1
            P5OUT &= ~P5_MOSFET_ON2;
            for (temp1=0;temp1<50000;temp1++);
        }
    }
    else
    {
        if (KU1_ON) // if *100 is to much,
        {
            if (ADCM>2400)
            {
                P5OUT &= ~P5_MOSFET_ON1; // *1
                KU1_ON=0;
                for (temp1=0;temp1<50000;temp1++);
            }
        }
        else
        {
            if (ADCM>2200) // if to much
            {
                KU2_ON=1;
                P5OUT |= P5_MOSFET_ON2; // *1/100
                for (temp1=0;temp1<50000;temp1++);
            }
            if (ADCM<30) // not anouth -
            {
                KU1_ON=1;
                P5OUT |= P5_MOSFET_ON1; // *100
                for (temp1=0;temp1<50000;temp1++);
            }
        }
    }
}

```

Приложение Г - продолжение

```

        for (templ=0;templ<50000;templ++);
        for (ol=0;ol<50000;ol++);

    }

}
//main

// interrupt[TIMERA0_VECTOR] void Timer_A (void)
#pragma vector=TIMERA0_VECTOR//VECTOR
__interrupt void my_f3(void)
{ TACTL &= ~(MC0+MC1);//stop timer
}
INT16U ADC(INT8U port_pin)
{ unsigned int ADCo1;
  //P5OUT = P5flag;

  //ADC_SEL = 0xff; //select alternative port
function-ADC
  ADC12CTL0 = ADC12ON; //+REF2_5V+REFON;//-internal
ref
  //ADC12MCTL0 = 0x10;// internal ref
  //----- setup ADC -----
-
  ADC12MCTL0 = port_pin + 0xB0; // reference select, channel
sel.
  //delay(20);

  //PSM_CTL_POWER(PSM_OFF);

  ADC12CTL0 |= ADC12SC + ENC; // Sampling open
  ADCo1 = 0;
  ADCo1 = 1;
  //delay(1);
  ADC12CTL0 &= ~ADC12SC; // Sampling closed, start
conversion
  //PSM_CTL_POWER(PSM_ON);
  while ((ADC12CTL1 & ADC12BUSY) == 1); // ADC12BUSY?
  ADC12CTL0 &= ~ENC;
  //P5OUT &= ~P5flag;
  return(ADCo1); //return ADC value
}

void delay(unsigned t)//t-B MKC
{ //TIMER_A setup
  TACTL = TASSEL1 + TACLK; // SMCLK, clear TAR
  TACTL |= ID1+ID0; // DIVIDER=8
  CCTL0 = CCIE; // CCR0 interrupt enabled
  CCR0 = t;

  TACTL |= MC0; // Start Timer_a in upmode
  while(TACTL & MC0);
}

//----- initialisation function -----
/*choosing high frequency oscillator XT2
wait for stabilisation of oscillator
setup UART 19.200 kBod,8 bit length
stop WDT

```

Приложение Г - продолжение

```
*/
void controller_init(void)
{

//Oscillator
BCSCTL1|=XTS;
BCSCTL1 &= ~XT2OFF;
//BCSCTL1=0;
#ifdef release
for (i=0;i<5000;i++);
#endif
IFG1 &= ~OFIFG;
BCSCTL2 |= SELM1+SELS;
UCTL0 &= ~ SWRST;//SWRST=0
WDTCTL=WDTPW+WDTHOLD;                               //WDTCTL=0X5A80;//Stop watchdog
timer
    _EINT();

}

void filter_init()
{
//----- define port direction and state -----
//P5
P5DIR = P5flag+P5_MOSFET_ON1+P5_MOSFET_ON2;
P5OUT = P5flag;
//P6
P6DIR = 0;//OUT_D;//IN-INPUTS
P6OUT = 0;//OUT_D;
P6SEL = FADC_in + KPeakDetADC_in;//
}
```

Приложение Д – пример отладочной программы на языке TurboPascal 7.0

```
{AIK debug software}
uses dos,crt;
var ignoreerror,bolka:boolean;
    s,b1,b2,b3,kal,w1,l,n,i,k,flag,f2,f3,f4:longint;
    c:char;
    r,j,w:real;
{
-----procedure COMInit-speed :115200/6 =19600
}
procedure COMInit;
begin
clrscr;
port[$3fb]:=128;
port[$3f8]:=6;
port[$3f9]:=0;
port[$3fb]:=3;
delay(60);
writeln('      *      port initialization complete  * ');
delay(60);
writeln('      *      press Esc to Esc      * ');
writeln(' use buttons:up,down,right,left' );
writeln(' 1-adc horisontal' );
writeln(' 2-adc vertical' );
writeln(' 3-azimut (horisontal)' );
writeln(' 4-ugol (vertical)' );
writeln(' 5-adc 1' );
writeln(' K-kalibrovka ' );
writeln(' P-proverka ');
writeln(' I-get info:adch,adcv,azimut,ugol');
writeln(' S-scan (r) right (n) steps(not work yet)');
writeln(' move right c steps R C' );
writeln(' left L C' );
writeln(' up U C' );
writeln(' down D C' );
{writeln(' kalibrovka ' );}
end;
{-----procedure sendbyte(COMport:word;b:byte);
}
procedure sendbyte(COMport:word;b:byte);
var status:byte;
begin
while (port[$3fd-(COMport-1)*$100] and $20 = 0)and (not ignoreerror) do;
port[$3f8-(COMport-1)*$100]:=b;
end;
{
function getbyte3(COMport:word):byte;
var status:byte;
begin
```

```

while (port[$3fd-(COMport-1)*$100] and $01 = 0) and (not ignoreerror) do;
getbyte3:=port[$3f8-(COMport-1)*$100];
end;
}
function getbyte2(COMport:word):byte;
var status:byte;
begin
    f2:=0;

    bolka:=false;
    repeat f2:=f2+1;
if port[$3fd-(COMport-1)*$100] and $01 = 1 then bolka:=true;
until (f2 = 500000) or bolka;
    {writeln(bolka); {}
getbyte2:=port[$3f8-(COMport-1)*$100];
end;

function getbyte(COMport:word):byte;
var status:byte;
begin
    f2:=0;

    bolka:=false;
    repeat f2:=f2+1;
if port[$3fd-(COMport-1)*$100] and $01 = 1 then bolka:=true;
until (f2 = 500000) or bolka;
    {writeln(bolka); {}
getbyte:=port[$3f8-(COMport-1)*$100];
end;

begin
    i:=0; j :=0; k:=0;

COMInit; {}
{com proove}
port[$3f8]:=8;
delay(1000);
if port[$3f8]<>13 then
begin
port[$3f8]:=21;
delay(1000);
if port[$3f8]<>10 then
begin
writeln('communication is not established');
writeln(' PRESS ESC AND TURN ON THE RED BUTTON');
sound(880);
delay(900);
nosound;
sound(440);
delay(900);

```

```

nosound;
sound(220);
delay(900);
nosound;

```

```

end;

```

```

end;
sendbyte(1,13);
write('proverka');
l:=getbyte2(1);
if l=15 then writeln('- azimuth and ugol working') else
begin
  writeln('-azimuth and ugol NOT working',l);
  sound(1975);
  delay(300);
  nosound;
  delay(300);
  sound(1975);
  delay(300);
  nosound;
  delay(300);
  sound(1760);
  delay(300);
  nosound;

```

```

end;
sendbyte(1,18);
kal:=getbyte2(1);
sendbyte(1,1);
kal:=kal+getbyte2(1)*256;
s:=0;
writeLN(' kalibrovka finished ');
repeat {*****ОСНОВНОЙ ЦИКЛ}
c:=readkey;
n:=Ord(C);
writeln(' ',n);
if c=#0 then
begin
  c:=readkey;
  i:=Ord(C);
  if i=77 then begin j:=port[$3f8];sendbyte(1,21);j:=getbyte2(1);write('r');end;
  if i=75 then begin j:=port[$3f8];sendbyte(1,22);j:=getbyte2(1);write('l');end;
  if i=72 then begin j:=port[$3f8];sendbyte(1,23);j:=getbyte2(1);write('up ');end;
  if i=80 then begin j:=port[$3f8];sendbyte(1,24);j:=getbyte2(1);write('down');end;
end;
if n=105 then {data,i}
begin
  s:=s+1;
  j:=port[$3f8];

```

```

sendbyte(1,20);
write('adc0(hor pol)=');
j:=getbyte(1);
{write(j,' ');}
w1:=port[$3f8];{}
sendbyte(1,1);
k:=getbyte(1);
{write(k,' ');}
w:=4*(j+256*k);
w:=w/1000;
write(w:6:3,'V');
write(' azimuth=');
w1:=port[$3f8];{}
sendbyte(1,1);
j:=getbyte(1);
{write(j,' ');}
w1:=port[$3f8];{}
sendbyte(1,1);
k:=getbyte(1);
{write(k,' ');}
j:=j+256*k;
j:=(j/kal)*360;
write(j:6:2,'DEG');
write(' ugol=');
w1:=port[$3f8];{}
sendbyte(1,1);
j:=getbyte(1);
{write(j,' ');}
w1:=port[$3f8];{}
sendbyte(1,1);
k:=getbyte(1);
{write(k,' ');}
j:=j+256*k;
j:=(j/kal)*360;
write(j:6:2,'DEG');
write(' adc0(ver_pol)=');
w1:=port[$3f8];{}
sendbyte(1,1);
j:=getbyte(1);
{write(j,' ');}
w1:=port[$3f8];{}
sendbyte(1,10);
k:=getbyte(1);{write(k,' ');}
w:=4*(j+256*k)/1000;{}
writeln(w:5:3,'V');
end;
if n=49 then{1-adch}
begin
j:=port[$3f8];
sendbyte(1,9);
write('adc0(hor pol)=');
j:=getbyte(1);

```

```

{write(j, ' '); {}
w1:=port[$3f8]; {}
sendbyte(1,10);
k:=getbyte(1);
{write(k, ' '); {}
w:=4*(j+256*k)/1000;
write(w:6:3,'V');
end;
if n=50 then {2-adcv}
begin
j:=port[$3f8];
sendbyte(1,10);
write('adc0(vert pol)=');
j:=getbyte(1);
{write(j, ' '); {}
w1:=port[$3f8]; {}
sendbyte(1,10);
k:=getbyte(1);
{write(k, ' '); {}
w:=4*(j+256*k)/1000;
write(w:6:3,'V');
end;
if n=51 then {-az}
begin
j:=port[$3f8];
write('azimut=');
sendbyte(1,11);
j:=getbyte(1);
{write(j, ' '); {}
w1:=port[$3f8]; {}
sendbyte(1,1);
k:=getbyte(1);
{write(k, ' '); {}
w:=j+256*k;
write(w:6:2);
if kal>0 then
begin
r:=(w/kal)*360;
writeln(' ',r:6:3,'degrees');
end;
end;
if n=52 then {4-ugol}
begin
j:=port[$3f8];
write('ugol=');
sendbyte(1,12);
j:=getbyte(1);
{write(j, ' '); {}
w1:=port[$3f8]; {}
sendbyte(1,1);
k:=getbyte(1);
{write(k, ' '); {}

```

```

w:=j+256*k;
write(w:5:0);
if kal>0 then
begin
r:=(w/kal)*360;
writeln(' ',r:5:3,'degrees');
end;

end;
if n=53 then {1-adc1}
begin
j:=port[$3f8];
sendbyte(1,9);
write('adc1=');
j:=getbyte(1);
w1:=port[$3f8]; {}
sendbyte(1,10);
k:=getbyte(1);
write(w1,' '); {}
w:=4*(j+256*k)/1000;
write(w:6:2,'V');
end;

if n=107 then {kalibrovka}
begin
j:=port[$3f8];
write('   360 degrees=');
sendbyte(1,18);
j:=getbyte(1);
{write(j,' '); {}
w1:=port[$3f8]; {}
sendbyte(1,10);
w1:=w1+getbyte(1)*256;
writeln(w1);
kal:=w1;
end;
if n=112 then {}
begin
j:=port[$3f8];
sendbyte(1,13);
write('proverka');
j:=getbyte2(1);
write(j:5:0); {}
if j=15 then writeln('working') else writeln('not working')
end;
if n=114 then
begin
write('move right ');
sendbyte(1,14);
j:=getbyte(1);
read(b1);
b2:=b1 mod 256;

```

```

b1:=b1 div 256;
sendbyte(1,b2);
j:=getbyte(1);
sendbyte(1,b1);
{for i:=1 to 1000 do
begin
delay(3);
write(port[$3f8]);
end; } i:=0;
repeat if port[$3fd] and 1 =1 then i:=i+1;
until port[$3f8]=20;
write(i);
writeln('steps ');
end;
if n=108 then
begin
sendbyte(1,15);
write('move left ');
j:=getbyte(1);
readln(b1);
b2:=b1 mod 256;
b1:=b1 div 256;
sendbyte(1,b2);
j:=getbyte(1);
sendbyte(1,b1);
i:=0;
repeat if port[$3fd] and 1 =1 then i:=i+1;
until port[$3f8]=20;
write(i);
writeln('steps ');
end;
if n=117 then
begin
sendbyte(1,16);
write('move up ');
j:=getbyte(1);
read(b1);
writeln('steps ');
b2:=b1 mod 256;
b1:=b1 div 256;
sendbyte(1,b2);
j:=getbyte(1);
sendbyte(1,b1);
i:=0;
repeat if port[$3fd] and 1 =1 then i:=i+1;
until port[$3f8]=20;
write(i);
writeln('steps ');
end;
if n=100 then
begin
sendbyte(1,17);

```

```

write('move down ');
j:=getbyte2(1);
read(b1);
b2:=b1 mod 256;
b1:=b1 div 256;
sendbyte(1,b2);
j:=getbyte2(1);
sendbyte(1,b1);
i:=0;
repeat if (port[$3fd] and 1 =1) or (port[$3fd] and 1 =1) or (port[$3fd] and 1 =1) or
(port[$3fd] and 1 =1) then i:=i+1;
until port[$3f8]=20;
writeln(i);
writeln('steps ');

end;
if n=115 then
begin
write('scan');
c:=readkey;
if c='r' then
begin
sendbyte(1,25);
getbyte(1);
read(b3);
b2:=b3 mod 256;
b1:=b3 div 256;
sendbyte(1,b2);
j:=getbyte(1);
sendbyte(1,b1);
j:=getbyte(1);
for l:=1 to b3 do
begin
write('adc0(hor pol)=');
j:=getbyte(1);
{write(j,' ');}
w1:=port[$3f8]; {}
sendbyte(1,1);
k:=getbyte(1);
{write(k,' ');}
w:=4*(j+256*k);
w:=w/1000;
write(w:6:3,'V');
write(' adc0(ver_pol)=');
w1:=port[$3f8]; {}
sendbyte(1,1);
j:=getbyte(1);
{write(j,' '); {
w1:=port[$3f8]; {}
sendbyte(1,1);
k:=getbyte(1); {write(k,' ');}
w:=4*(j+256*k)/1000; {}

```

```
writeln(w:5:3,'V');
```

```
write(' azimuth=');  
w1:=port[$3f8]; {}  
sendbyte(1,1);  
j:=getbyte(1);  
{write(j,' '); { }  
w1:=port[$3f8]; {}  
sendbyte(1,1);  
k:=getbyte(1);  
{write(k,' '); { }  
j:=j+256*k;  
j:=(j/kal)*360;  
write(j:6:2,'DEG');  
w1:=port[$3f8]; {}  
sendbyte(1,1);  
end;  
end;
```

```
end;  
until n=27;  
writeln('*****---FINISH---*****');  
delay(150);  
sound(147);  
delay(500);  
nosound;  
delay(40);  
end.
```

Приложение Е – схема модуля контроллера в программе OrCad 9.1

Приложение Ж - перечень элементов