СОДЕРЖАНИЕ

ВВЕДЕНИЕ

С развитием технологий и увеличением сложности электронных устройств, точность и эффективность измерений становятся ключевыми факторами в различных областях науки и техники. Одним из важнейших инструментов для таких измерений является осциллограф — прибор, предназначенный для регистрации и анализа электрических сигналов. В частности, цифровые осциллографы, благодаря своей способности обрабатывать данные и отображать результаты с высокой точностью, становятся незаменимыми в широком спектре применений, от научных исследований до разработки и тестирования современных электронных устройств.

С каждым годом требования к таким приборам растут: необходимость анализа сигналов высокой частоты, работы с многоканальными системами, повышение точности измерений и миниатюризация приборов диктуют новые условия для разработки. В то же время, современные тренды в области микроэлектроники, такие как использование микроконтроллеров, позволяют значительно сократить размеры устройств и снизить их стоимость, что делает осциллографы доступными для более широкого круга пользователей, включая студентов, хобби-энтузиастов и мелкие лаборатории.

Актуальность данного проекта обусловлена стремлением создать компактный, цифровой осциллограф на базе микроконтроллера.

1 Теоретическая часть

1.1 Анализ предметной области

Осциллограф – это прибор, предназначенный для регистрации, визуализации и анализа электрических сигналов во времени. Он позволяет исследовать форму сигнала, его частоту, амплитуду, длительность импульсов и другие параметры. Осциллографы широко применяются в радиоэлектронике, схемотехнике, измерительных системах и научных исследованиях, а также при ремонте и отладке электронных устройств.

Существует несколько типов осциллографов, включая аналоговые, цифровые и смешанные модели. Аналоговые осциллографы представляют сигнал в виде отклонения электронного луча на экране ЭЛТ, тогда как цифровые устройства преобразуют сигнал в цифровую форму, обрабатывают его и выводят на дисплей.

Современные осциллографы, особенно портативные и встроенные решения, нередко разрабатываются на основе микроконтроллеров. Это позволяет значительно уменьшить габариты прибора, снизить энергопотребление и обеспечить гибкость в программном управлении. В частности, микроконтроллерные осциллографы могут сохранять сигналы, анализировать их и выводить на экраны с интерфейсом I2C, что делает их удобными для компактных разработки проектов малого размера и доступными в разработке на программном уровне.

Цель разработки – Цифровой осциллограф на базе микроконтроллера. Основная задача устройства – измерение и отображение электрических сигналов в реальном времени на дисплее. В функционал схемы входит сохранение отображенного графика и вывод сохранений из памяти.

Пользовательские аспекты осциллографа требуют отдельного внимания. Эффективность и удобство использования такого устройства во многом зависят от интуитивно понятного интерфейса и доступности инструкций, простоты работы с устройством. Пользователи должны иметь возможность легко задавать необходимые параметры.

1.2 Устройство осциллографа в промышленности

Принцип работы осциллографа заключается в измерении и отображении электрических сигналов (например, напряжений) в виде графика во времени. Осциллограф используется для наблюдения формы сигнала, его амплитуды, частоты.

Функционал современных осциллографов – включает в себя ПЗУ и ОЗУ, масштабирующий модуль, АЦП, контроллер, органы управления и дисплей. Измерение цифровым осциллографом позволяет совершать множество операций, получая разнообразные данные:

1. напряжение постоянного и переменного тока;
2. частоту и период;
3. характеристики и сопротивление напряжения;
4. звук, шум и соотношение шума к сигналу;
5. амплитуду и сдвиг фаз;
6. рабочий цикл;
7. падение напряжения;
8. время подъема и падения.

Рисунок 1.1 – Осциллограф GW Instek MDO-72072EG

Наблюдение и контроль периодических сигналов разных форм (треугольной, прямоугольной и синусоидальной) осуществляется посредством прохождения входного сигнала через масштабирующее устройство, где он усиливается и разделяется в аналогово-цифровой преобразователь, отвечающий за визуализацию. После модификации информация сохраняется в блоке памяти. Далее происходит реконструкция и вывод значений на дисплей.

Широкий диапазон развертки позволяет контролировать даже наносекундные интервалы, наблюдать сигналы в различных точках схемы и измерить время нарастания импульса, что имеет большую важность в работе с цифровой аппаратурой.

Оборудование разных типов помогает осуществлять проверку, настройку и регулировку многообразной радиоэлектроники, электронной техники, ремонт бытовой техники и диагностику ТС. Такие устройства широко применяются в медицине, прикладных, лабораторных и научно-исследовательских сферах. Кроме цифровых осциллографов стоит упомянуть и экземпляры гибридного вида. Выделяя их в группы:

1. электронные – подразделяются в свою очередь на цифровые и аналоговые приборы (по принципу обработки информации);

2. электромеханические – подразделяются на выпрямительные, магнитоэлектрические, электродинамические, электромагнитные, термоэлектрические и электростатические модели.

По количеству лучей и каналов различают однолучевые и многолучевые разновидности (16 и более), а также одноканальные и многоканальные (до 16 каналов). Эти две группы контрольно-измерительных устройств имеют одно существенное отличие. Однолучевые – осциллографы с одним измерительным каналом, показывающие график сигнала только по одному измерительному каналу. Многолучевые – осциллографы с несколькими измерительными каналами. Данный вид наблюдает динамику по нескольким каналам. Осциллограф сложное и комплексное устройство, которое может реализовано

посредством разработки на аналоговых или цифровых логических схемах. В следствие чего осциллографы классифицируют по принципу работы на:

1. аналоговые.
2. аналогово-цифровые.
3. цифровые – делятся на запоминающие (DSO) и люминофорные (DPO) .
4. комбинированные.
5. виртуальные (на базе программного комплекса компьютера).

Принципиальная разница между этими разновидностями заключается в габаритах, возможностях запоминания, а также в методах обработки. Например, аналоговые осциллографы транслируют сигнал в реальном времени, без возможности записи. Аналогово-цифровые модели позволяют увидеть динамику изменения времени или амплитуды.

Полностью цифровые аналоги, соответственно, способны осуществлять цифровую обработку, оцифровывая синусоиду и передавая полученную информацию на дисплей.

1.3 Выбор компонентной базы для реализации проекта

На первом этапе проектирования функциональной схемы необходимо выбрать модуль, который будет отвечать за вывод графической информации и клавиатура управления, для обеспечения функционала взаимодействия с устройством. В данном случае был выбран дисплей 0.96 дюймов с разрешением 128px на 64px для вывода графической информации.

Рисунок 1.2 – Экранный модуль OLED- 12864 с матрицей, I2C

Диагональ дисплея 0.96 дюймов, на плате расположены 4 кнопки для подключения к микроконтроллеру. Все кнопки обеспечены подтягивающим резистором 0805 на 4.7кОм к основному питанию схемы. Выбор данного компонента обусловлен удобством монтажа на плате через отверстия крепления и удобным расположением кнопок управления устройством.

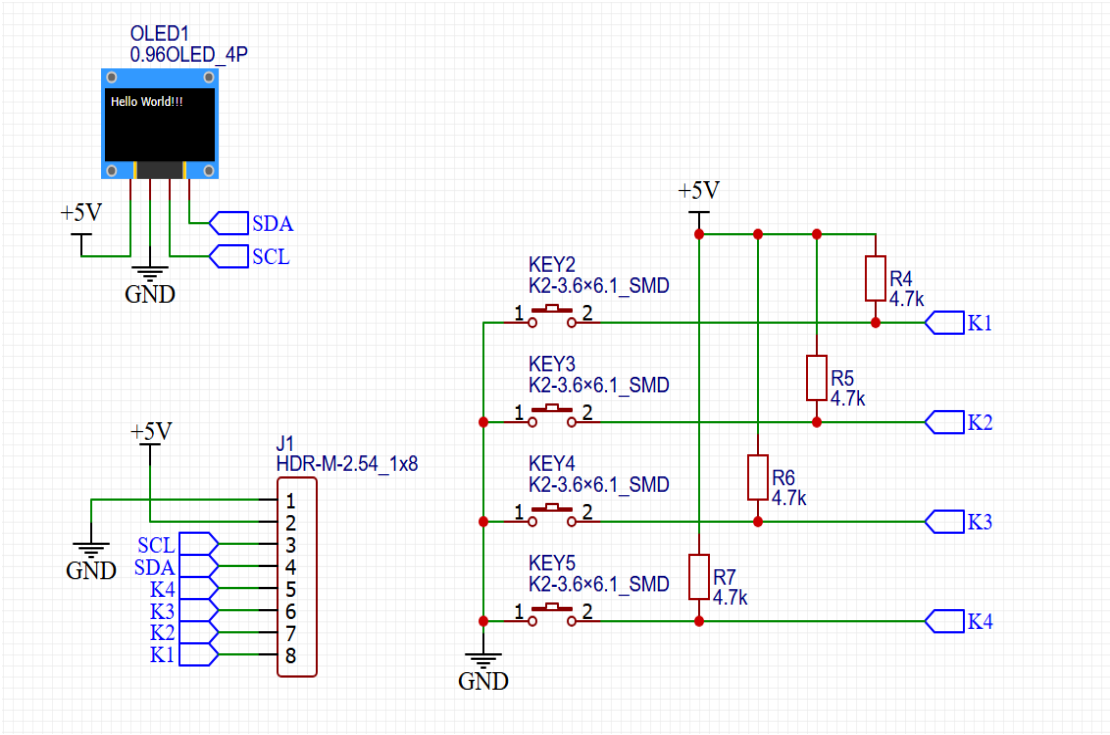


Рисунок 1.3 – Принципиальная схема дисплея со встроенной клавиатурой

На схеме, созданной в среде Easy-EDA указано строение и схема подключения кнопок управления на плате дисплея. В стандартной библиотеке предоставляемой от JLCPCB данный компонент отсутствует, был создан новый компонент с шелкографией и построением принципиальной схемы платы, повторяющей логику подключения и установки используемого образца. Из представленного на Рисунке 1.3 показано, что все выводные контакты имеют подключение к напряжению схемы через резисторы, с указанными ранее номиналами. Таким образом, в состоянии покоя кнопка имеет высокий логический уровень сигнала управления. При нажатии на любую из кнопок, соответствующий ей контакт примет нижний логический уровень. При подключении данного экрана к общей схеме, будет реализовано управление устройством, сохраняя компактность размеров итогового образца.

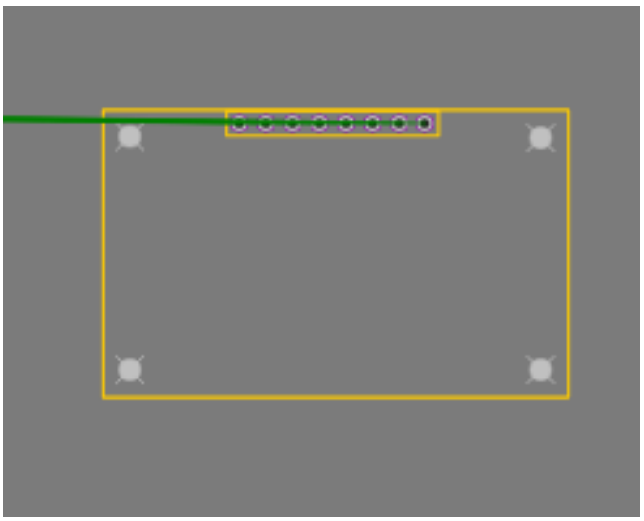


Рисунок 1.4 – Шелкография для расположения на плате

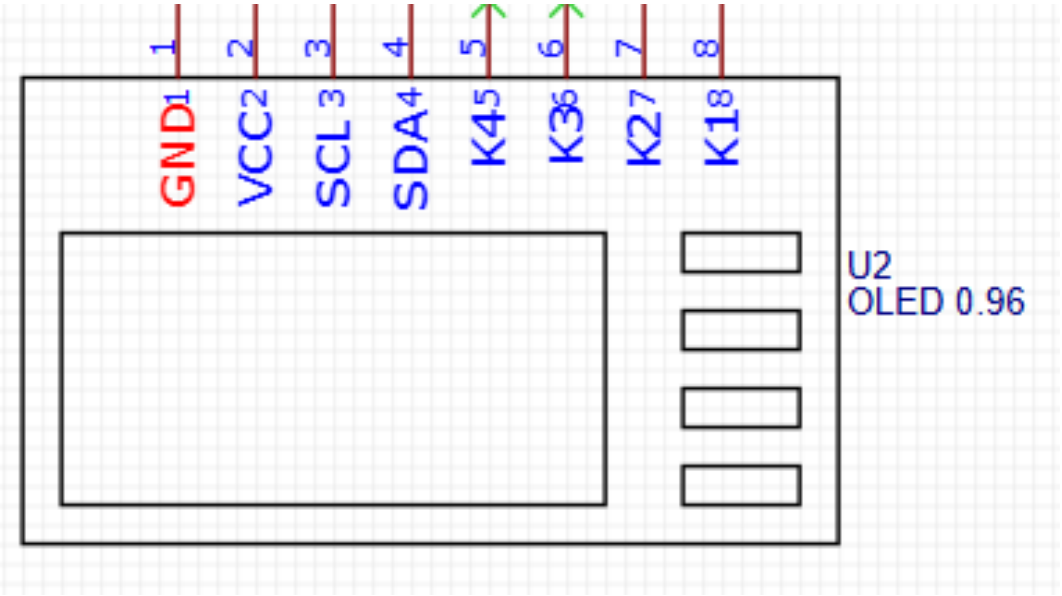


Рисунок 1.5 – Условное обозначение компонента в Easy-EDA

Основной платформой разработки цифрового устройства является цифровой элемент логических и арифметических операций, обеспечивающий функционал для устройства и производящий операции по выводу изображения. Основой платформы является микросхема Attiny 85-20PU. Данный микроконтроллер обеспечивает достаточную вычислительную мощность и функционал подключения интерфейсов, для обеспечения технических требований.

Микроконтроллер имеет 4-х канальный АЦП, обеспечивающий точность измерений в 10 бит, что дает диапазон цифровых значений от 0 до 1023 значений.

Таким образом, беря в учет опорное напряжение 5 вольт, разрешающая способность измерения составляет 4.88 милливольт.

Для вычисления разрешения измерения значений осциллографом приведена формула под номером 1.1.

$$\Delta V = \frac{V_{ref}}{2^N} \quad (1.1)$$

Где ΔV – минимальное изменение напряжения АЦП (разрешение измерения);

V_{ref} – опорное напряжение АЦП;

N – разрядность АЦП;

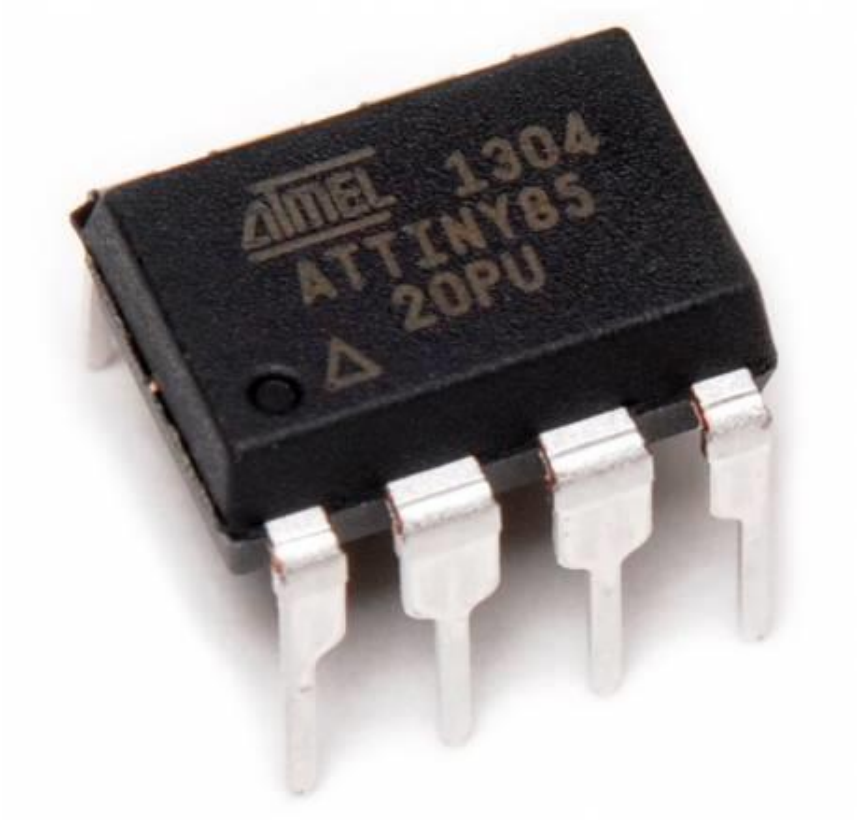


Рисунок 1.4 – Attiny85-20PU

Максимальное пороговое напряжение измерения, без применения схем на резистивных делителях и схем понижения напряжения, составляет – 5 вольт. В стандартных условиях встроенный АЦП способен измерить только однополярное напряжение.

2 Практическая часть

2.1 Проектирование функциональной схемы

Процесс создания функциональной схемы в среде Logisim Evolution представляет собой этап, в котором логические функции и цифровые устройства исполняются через визуальное моделирование. Logisim Evolution предоставляет более современный интерфейс и дополнительные инструменты для проектирования и анализа цифровых схем.

При разработке функциональной схемы необходимо учитывать несколько ключевых этапов. В первую очередь определяются требования к системе, на основе которых выбираются необходимые логические элементы и операторы. Понимание функции, которую должна выполнять схема, является важным моментом, поскольку это поможет оптимально подобрать компоненты.

После того как функциональные требования установлены, следует создать предварительный макет логической схемы. Этот макет может включать как простые логические операции (такие как конъюнкция, дизъюнкция, инверсия), так и более сложные конструкции, включая мультиплексоры и регистры. В САПР Logisim Evolution имеется обширная библиотека различных элементов, что значительно упрощает процесс проектирования. Одной из главных особенностей САПР Logisim Evolution является возможность интерактивного тестирования создаваемой схемы в реальном времени, что позволяет выявлять и устранять ошибки на ранних этапах разработки. Пользователь может подключать логические пробники к схеме и анализировать выходные данные в зависимости от заданных входных параметров.

На первом этапе проектирования устройство нужно определить несколько условий разработки схемы. Логика работы цифрового осциллографа основана на преобразовании аналоговой величины напряжения в цифровое значение, для представления его и проведения вычислительных операций. Не смотря на обширный функционал, Logisim Evolution не предоставляет средств работы с аналоговыми величинами, данная среда основана лишь на операциях с цифровыми и логическими значениями. Поэтому проведем условную

симуляцию работы АЦП. Для проведения всех операций было определено шестнадцатеричное основание системы счета значений. Ниже представлены сегменты схемы, проектируемой в данной вреде. Полная схема устройства представлена в документе КП.1993.09.02.01.2025.Э2.

Для отображения графической информации была взята матрица с разрешением 32 на 16 точек. Данный размер руководствуется сугубо для удобства обработки значений и исходит из ограничений размеров матриц в среде Logisim Evolution. Матрица представлена на Рисунке 2.1.

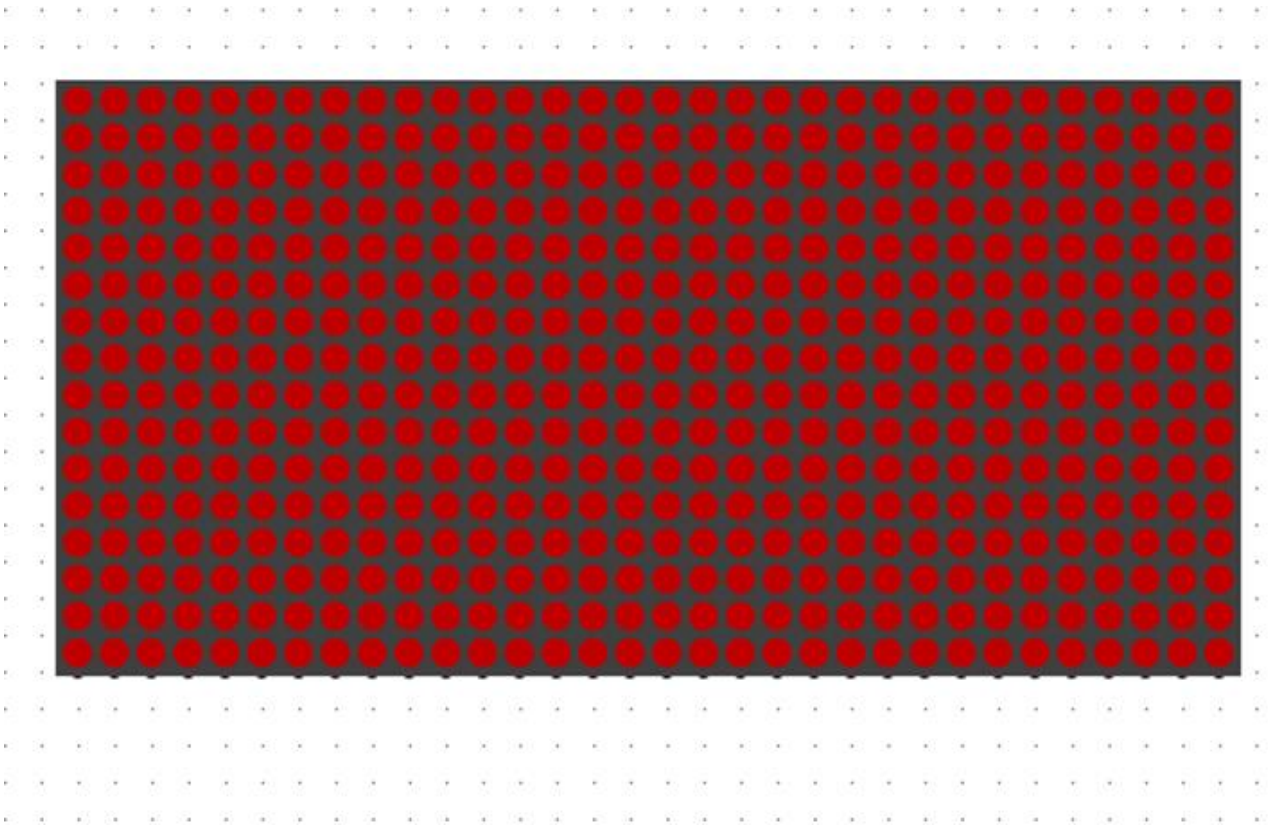


Рисунок 2.1 – Светодиодная матрица

Работа с матрицей происходит посредством ввода ряда чисел с шестнадцатеричным основанием, через шину ввода данных пиксельной матрицы, без применения деления матрицы на страницы рядов. Симуляция работы генератора графика развертки и перевод значений в нужный формат данных реализован через стандартные математические функции и операторы, предоставленные в среде. Схема преобразователя с описанием входных каналов представлена на Рисунке 2.2 и полный вид с генератором на Рисунке 2.3.

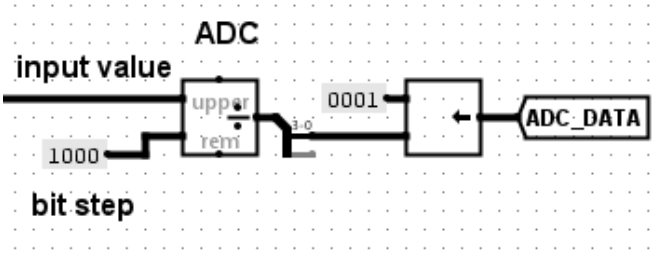


Рисунок 2.2 – Преобразователь значений

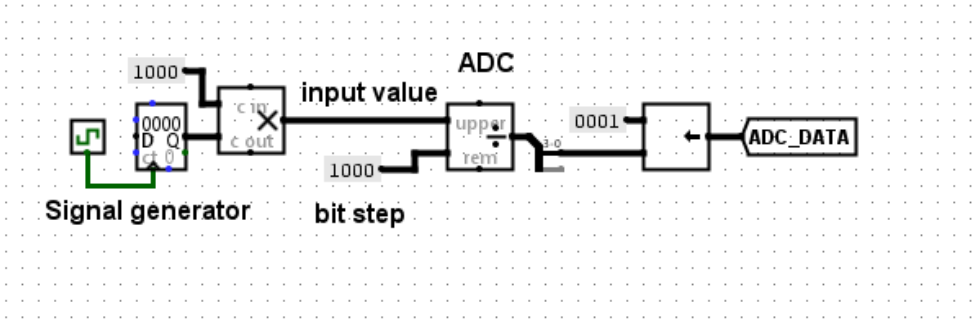


Рисунок 2.3 – Преобразователь с генератором значений

Данная схема выполняет деление числа с основанием 16 на битовую маску шага, результат операции в блок побитового сдвига, для перемещения бита пикселя на вычисленное число шагов по высоте.

На рисунке 2.4 представлена панель управления, где реализован функционал переключения страницы памяти, сохранения графика в память, запрос на чтение из памяти, а также очистка дисплея, смена режима между отображением пробы и данными сохраненными в память. Для упрощения отображения линий применены флаги с функциональными подписями.

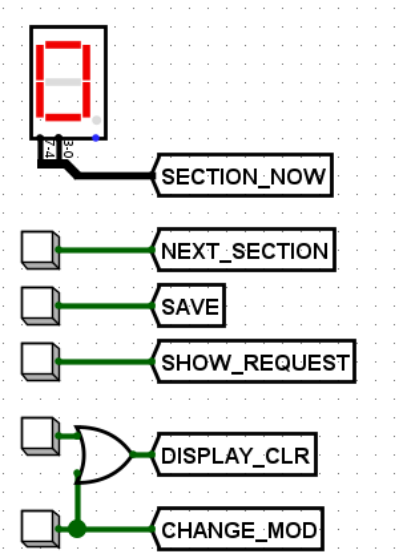


Рисунок 2.4 – Панель управления устройством

На Рисунке 2.5 представлен модуль отображения графика сигнала. Состоящий из сдвигового регистра аккумулятора считываемых значений, сдвигового регистра отображения дисплея, сдвигового регистра сохранения, счетчика для передачи сохраняемого в память массива значений. По запросу данные сохраняются и отправляются на выбранную страницу памяти. Сохраняется уже отображенный на экране график. Для вывода данных применяется счетчик и проверка на наличие данных в регистре через компаратор. Отображаемый поток значений определяет входной мультиплексор, состояние которого управляется режимом устройства. Значения могут идти напрямую из преобразователя значений пробы или из памяти, с выбранной страницы.

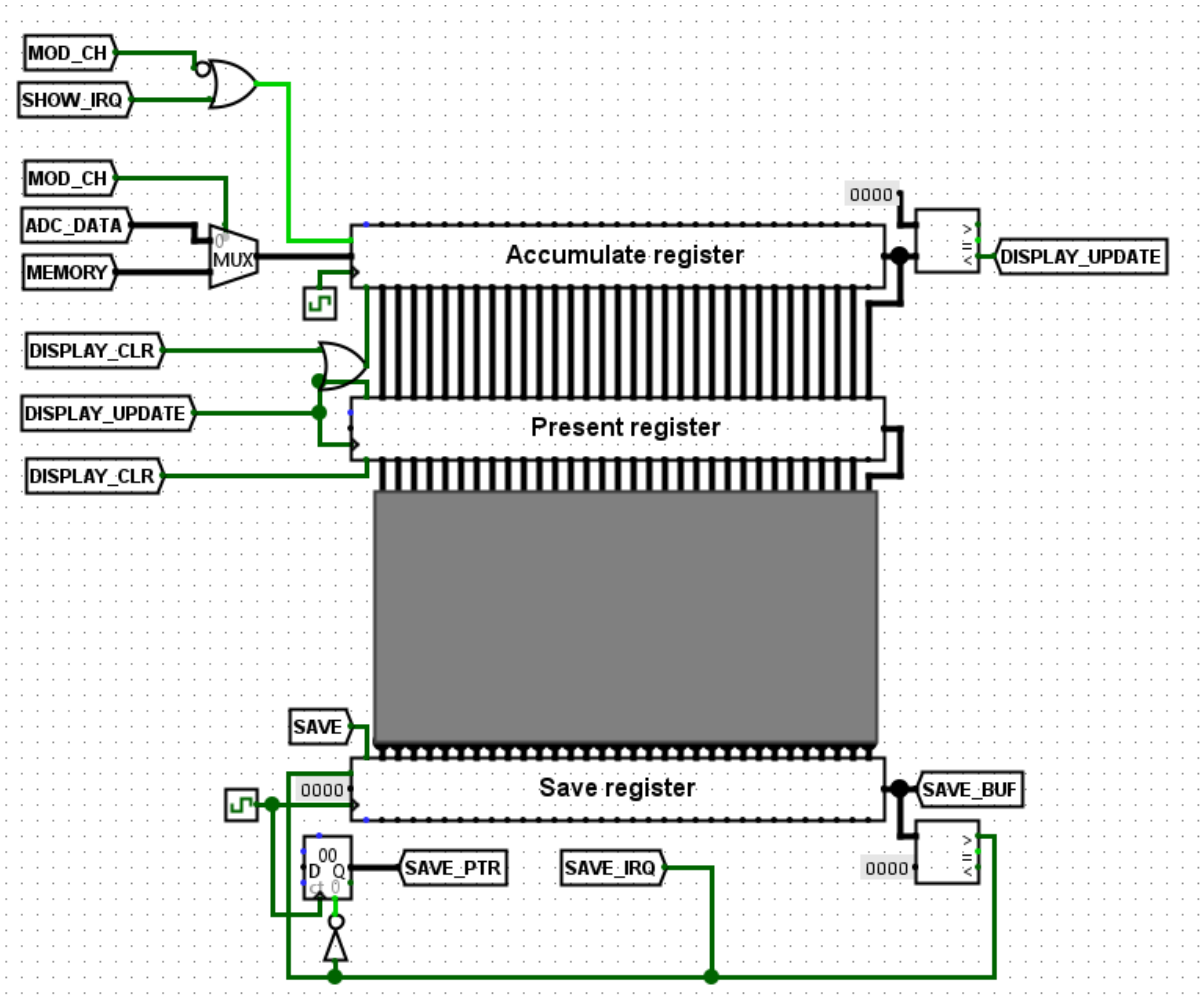


Рисунок 2.5 – Компонент отображения

Работа с памятью реализована через работу с ее адресами, для этого был реализован механизм сдвига указателя на страницу памяти и перемещение

указателя на смещение в странице. Указатель чтения данных и сохранения независимы по своим значениям и работа с ними определяется режимом работы устройства. Блок изменения режима устройства представлен на Рисунке 2.6.

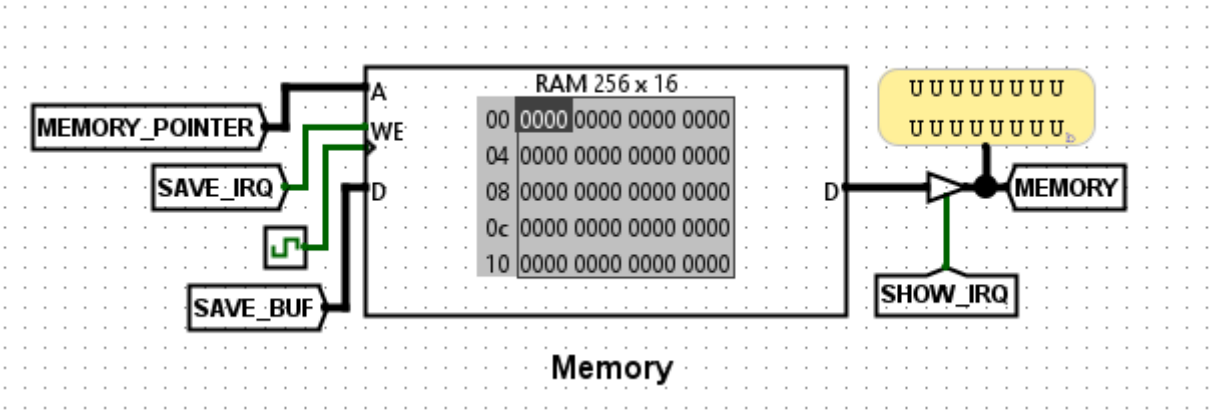


Рисунок 2.6 – Компонент работы с памятью

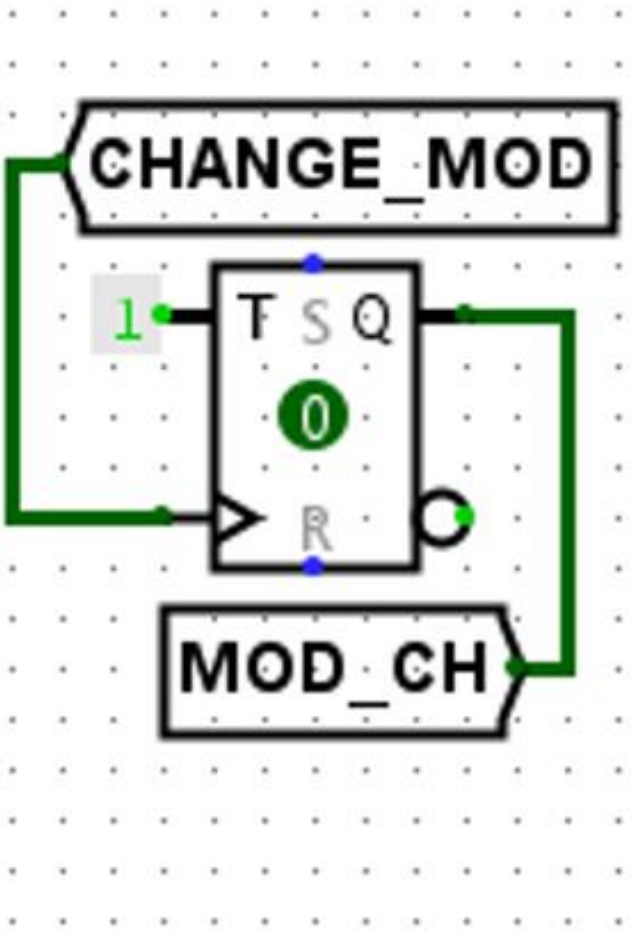


Рисунок 2.7 – Блок смены режима

Триггер режима сохранения и вывода сохраненных значений.

На рисунке 2.8 представлен компонент отображения данных со страницы. При подаче запроса отображения, триггер создает прерывание на отображение данных и счетчик передает на шину адреса читаемых значений. Было применено два счетчика из-за особенностей работы среды с тактовым генератором.

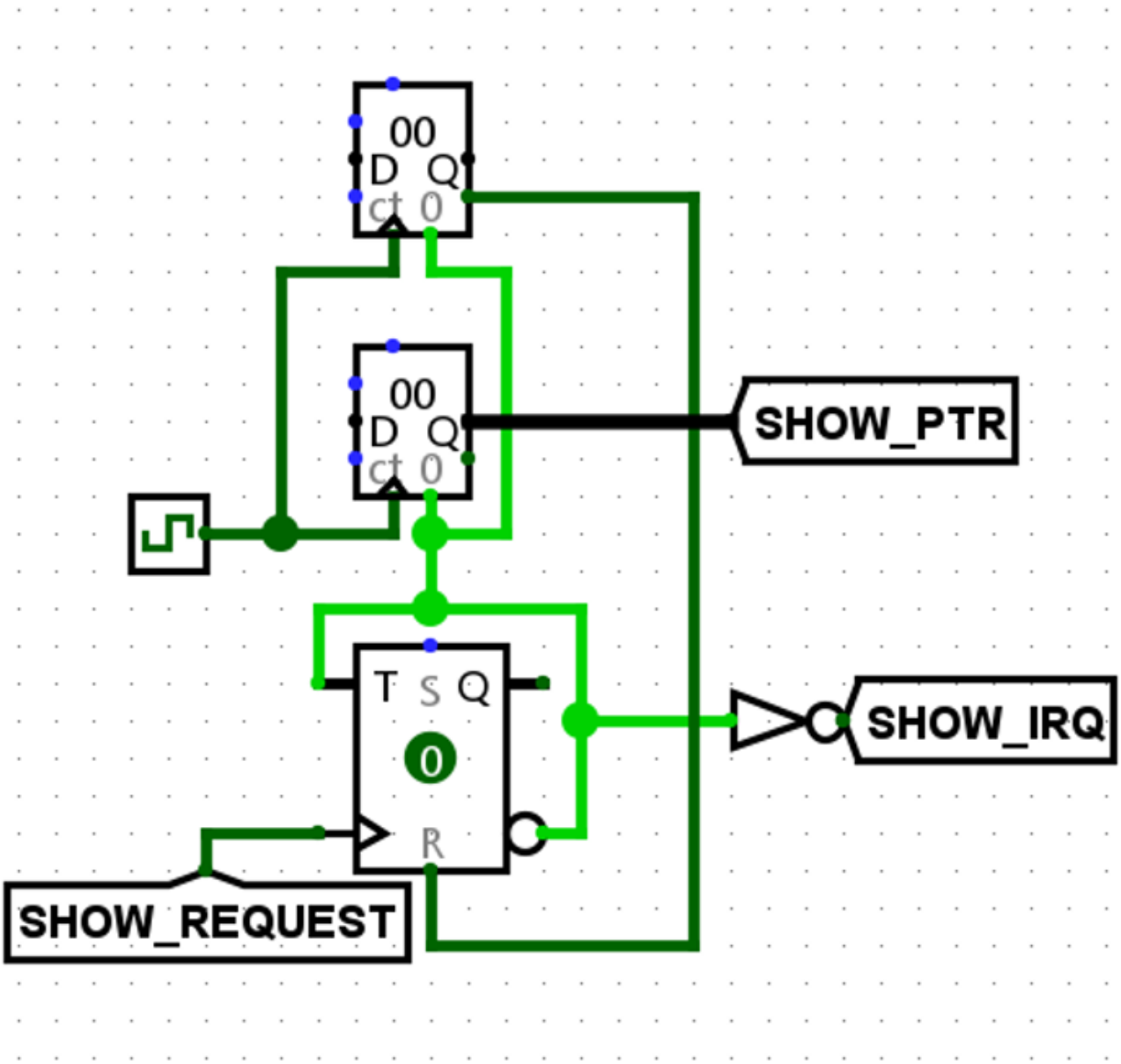


Рисунок 2.8 – Компонент отображения сохранения страницы памяти

2.2 Разработка программного кода в среде эмуляции

Проектирование любого цифрового устройства неотъемлемо требует создания исполняемой программной логики устройства, загружаемой в микроконтроллер и им исполняемой. На рынке присутствует огромное множество реализаций микроконтроллеров разного назначения, в пункте 1.3

была рассмотрена выбранная для проекта платформа. Разработка программного обеспечения для такого вида устройств сопровождается рядом особенностей, характерными для микроконтроллеров, отличимых от бытовых и портативных ЭВМ.

Отсутствие механизмов работы с потоками и самого понятия потоков в рамках работы с контроллерами такого класса, приводит к написанию программного кода, способного работать в последовательном режиме, без асинхронных и параллельных задач вычислений. Для разработки был применен фреймворк “Arduino”, являющейся программным окружением и инструментарием разработки на языке программирования C++. Данный инструментарий предоставляет утилиты компиляции, отладки и разработки под платформу микроконтроллеров. Для обеспечения совместимости с данной средой на нашей платформе, был применен загрузчик “ATTinyCore”. Были применены следующие библиотеки для разработки.

1. GyverButton – удобство работы с внешними кнопками устройства и настройки их логики.

2. Tiny4KOLED – библиотека работы с дисплеями SSD1306 на Attiny.

3. TinyWireM – библиотека работы с устройствами I^2C.

4. EEPROM – работа со встроенной EEPROM памятью.

Листинг программного кода в приложении А.

Для написания программного кода была применена среда эмуляции микроконтроллеров – SimulIDE. Она предоставляет обширный функционал в работе с AVR микроконтроллерами и поддержку компиляции программного кода сторонними средствами. Для написания программного кода была построена схема прототипа устройства на базе Attiny85-20PU, изображённая на Рисунке 2.9. Исходя из размеров дисплея в 128 точек, сохраняемый массив был выбран в 128 значений измерений источника сигнала, для сохранения максимального числа измерений и сохранения пространства оперативной памяти микроконтроллера.

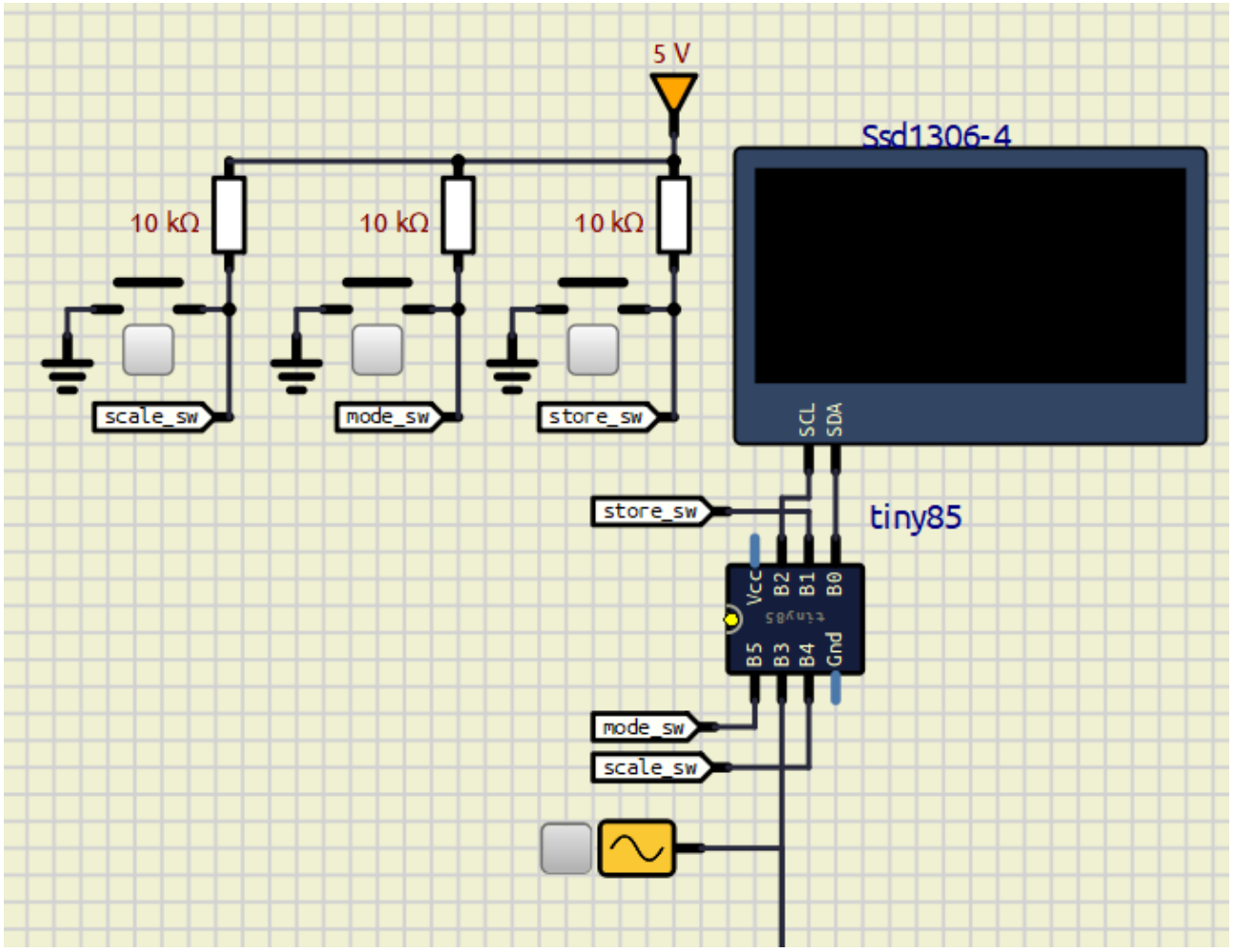


Рисунок 2.9 – SimulIDE схема прототипа

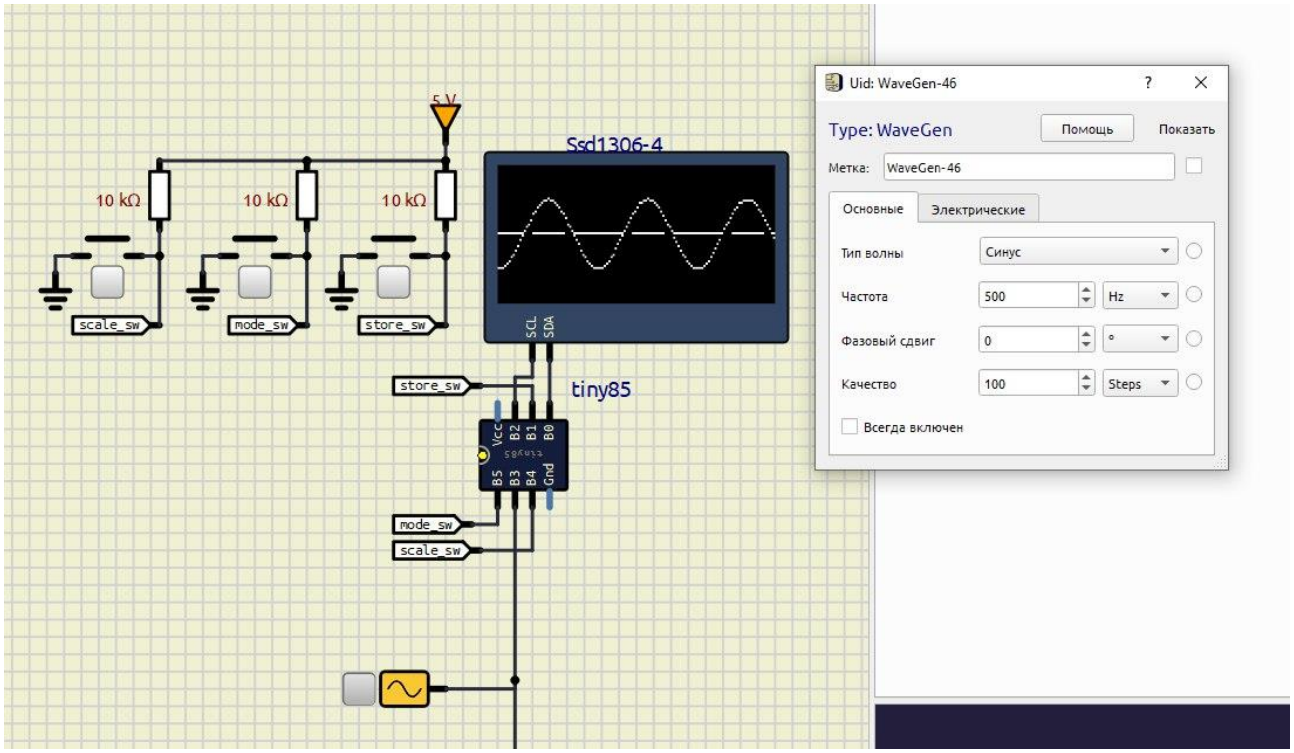


Рисунок 2.10 – Отображение волны 500Hz

Как видно на Рисунке 2.12 и 2.13 было реализовано масштабирование сигнала и сохранение графика значений в память EEPORM. Устройство имеет возможность отображать сохраненные графики и перезаписывать новые в память. Благодаря сохранению в EEPORM, устройство сохраняет графики даже после выключения. В стандартном виде EEPROM отдает значения в виде байта.

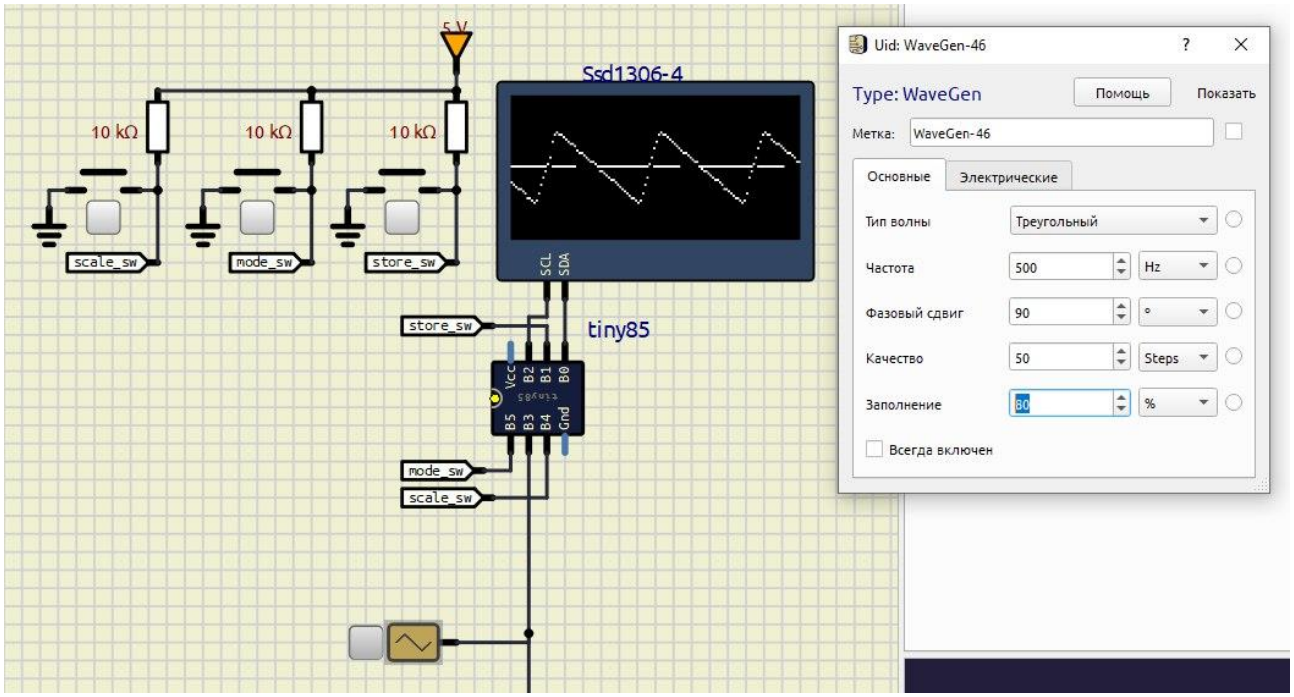


Рисунок 2.11 – Отображение треугольного сигнала с заполнением 80%

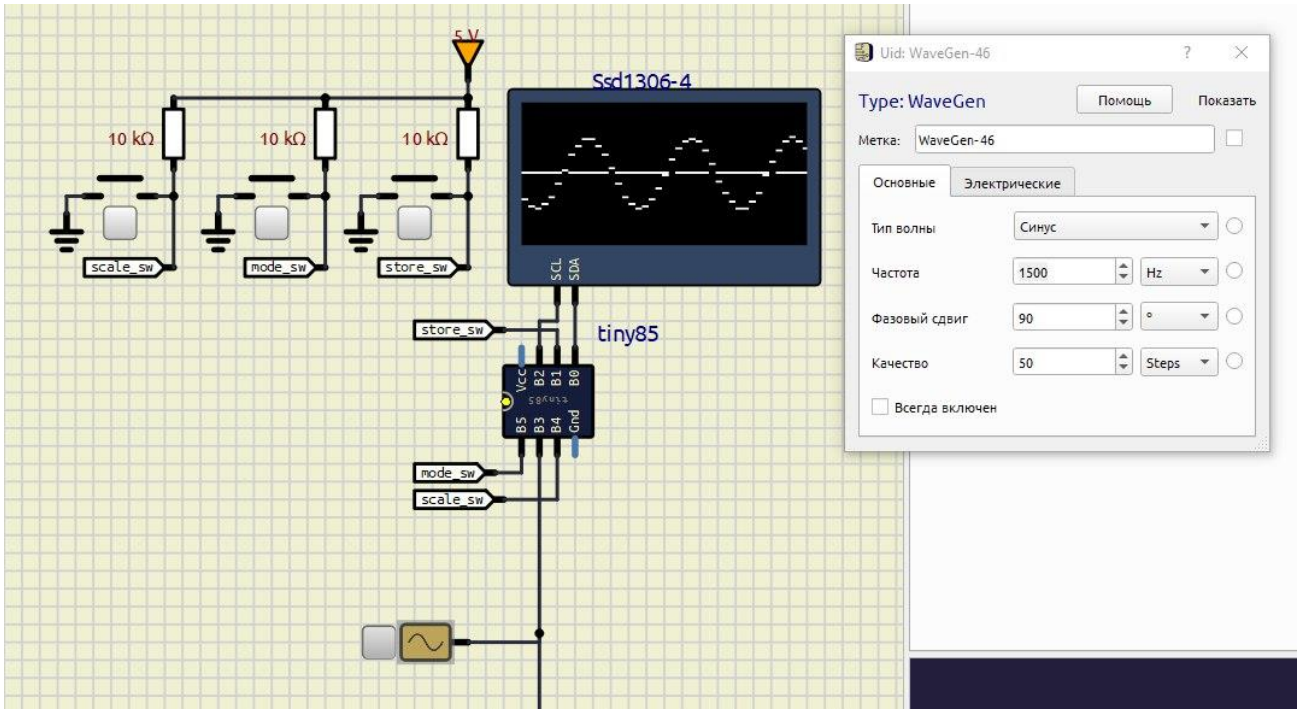
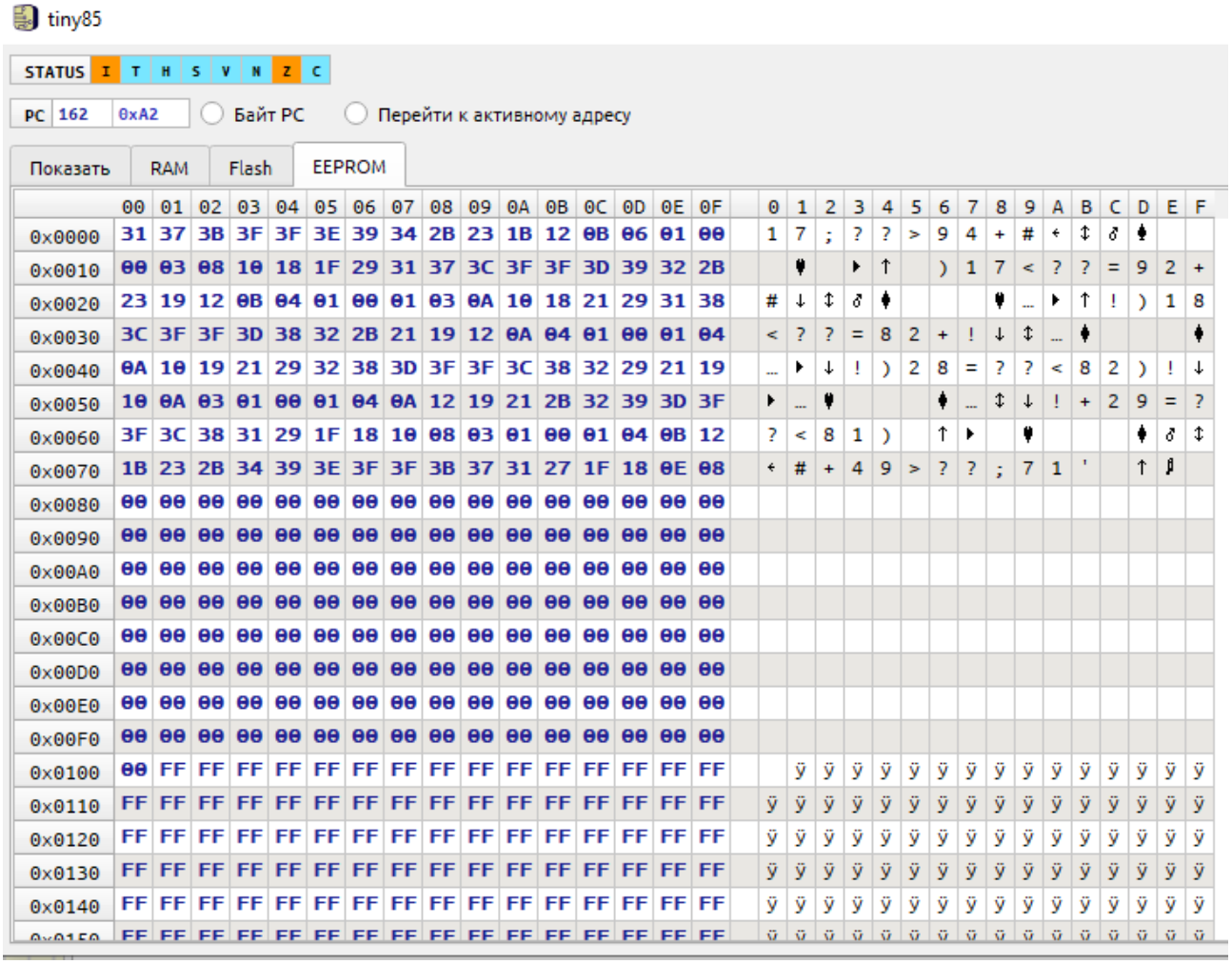


Рисунок 2.12 – Волна 1.5 kHz

При вызове операции чтения данных по указанному адресу или диапазону адресов, в котором может присутствовать вектор числовых значений.



	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0x0000	31	37	3B	3F	3F	3E	39	34	2B	23	1B	12	0B	06	01	00
0x0010	00	03	08	10	18	1F	29	31	37	3C	3F	3F	3D	39	32	2B
0x0020	23	19	12	0B	04	01	00	01	03	0A	10	18	21	29	31	38
0x0030	3C	3F	3F	3D	38	32	2B	21	19	12	0A	04	01	00	01	04
0x0040	0A	10	19	21	29	32	38	3D	3F	3F	3C	38	32	29	21	19
0x0050	10	0A	03	01	00	01	04	0A	12	19	21	2B	32	39	3D	3F
0x0060	3F	3C	38	31	29	1F	18	10	08	03	01	00	01	04	0B	12
0x0070	1B	23	2B	34	39	3E	3F	3F	3B	37	31	27	1F	18	0E	08
0x0080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x00A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x00B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x00C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x00D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x00E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x00F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0100	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0x0110	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0x0120	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0x0130	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0x0140	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Рисунок 2.13 – Сохранение графика в EEPROM в байтах

2.3 Проектирование принципиальной схемы в EasyEDA

В качестве среды проектирования будет использоваться EasyEDA – веб-среда для автоматизированного проектирования электронных устройств предназначенная, как для студентов-энтузиастов, так и профессионалов.

В основе EasyEDA лежит облачный сервис, который производит все вычислительные операции за счет мощных компьютеров, расположенных в Китае. Таким образом, скорость выполнения задач зависит не от характеристик персонального компьютера, а только от скорости интернет-соединения. Также сервис имеет файловый клиент, который немного упрощает и ускоряет работу,

но все операции так же выполняются через облако, и есть возможность локальной обработки изображений схем и плат.

Современные средства автоматизированного проектирования для создания схем электронных устройств работают следующим образом: вначале строится схема электрическая принципиальная, на которой четко видны связи всех компонентов, затем схема проверяется на ошибки визуально и при помощи встроенного компилятора.

EasyEDA предоставляет широкий спектр возможностей, например: редактор схем электрических принципиальных, редактор печатных плат, автотрассировка печатных плат, визуализатор печатной платы в 3D, создание файлов для производства (Gerber) печатной платы, возможность моделирования схем электрических принципиальных, экспорт в BOM (своеобразная спецификация) и многое другое. Исходя из сказанного ранее, в среде была разработана принципиальная схема устройства, представленная на рисунке 2.14.

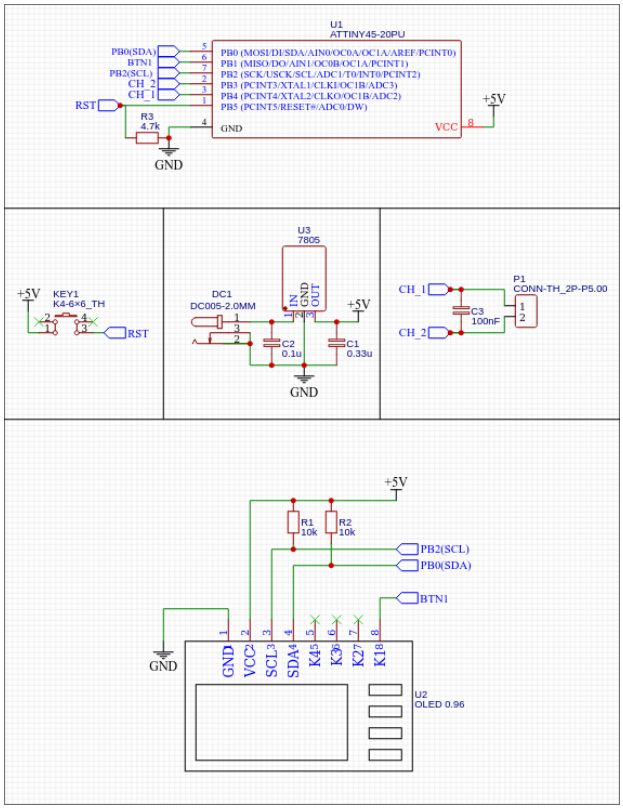


Рисунок 2.14 – Принципиальная схема устройства осциллографа

Предоставленная на рисунке 2.14 схема описывает принципиальное

строение устройства осциллографа. Устройство включает в себя микроконтроллер, выполняющий все вычислительные операции, панель управления кнопок устройства, экран отображения подключенный через шину I2C, кнопку перезагрузки устройства и вывод клеммных зажимных контактов, для подключения измерительного щупа.

Как представлено на рисунке 2.15 – узел питания представлен штекером 2мм DC, стабилизатором питания LM7805 в корпусе TO220 и фильтрующими керамическими конденсаторами на 100nF и 330nF, включенные в схему для подавления внешних шумов, создаваемых источником питания устройства. Результатом преобразования выходит стабильное питание 5 вольт.

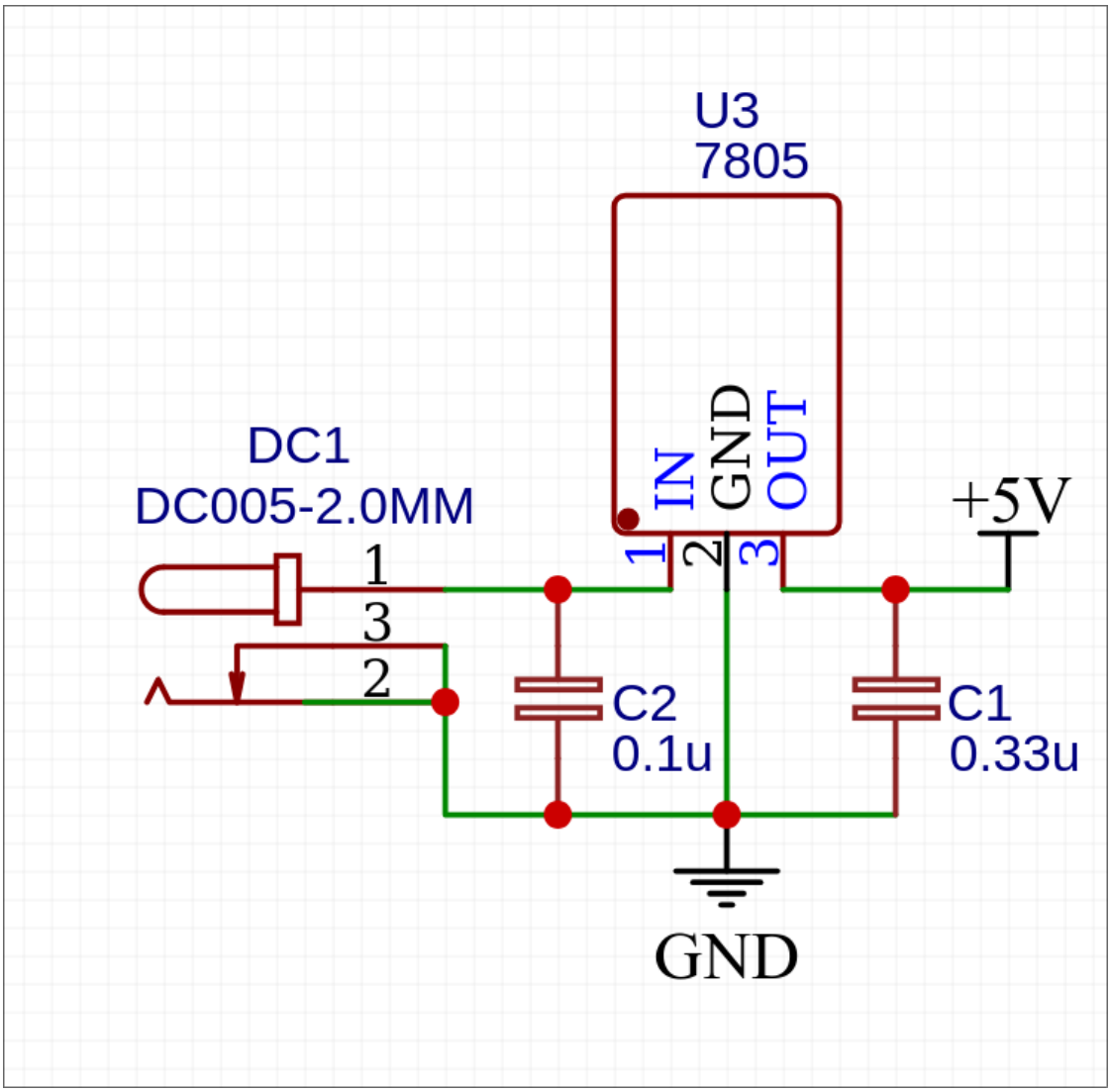


Рисунок 2.15 – Узел питания устройства

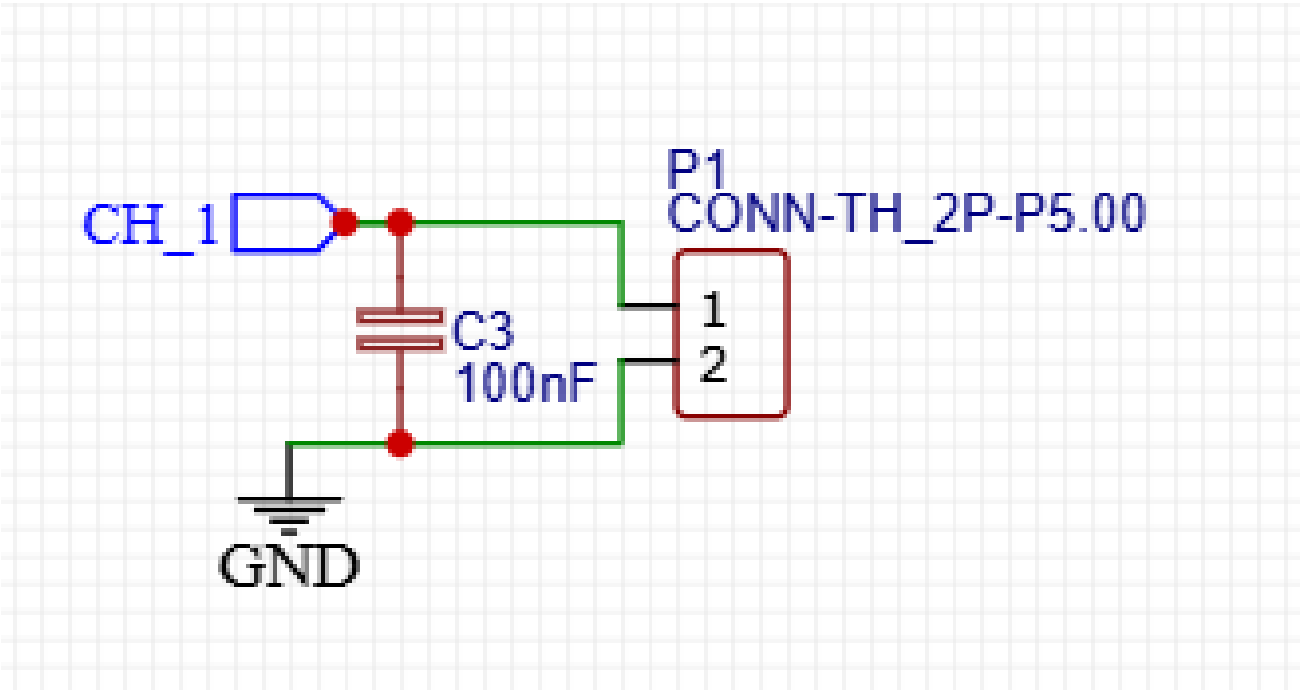


Рисунок 2.16 – Узел крепления измерительного щупа

Для полноценного функционала осциллографа устройство имеет возможность крепления внешних щупов заземления, для сопоставления уровня и канала замера напряжения. Исходя из технических характеристик устройства и вышеописанной схемы, пороговое напряжение измерения – 5 вольт.

2.4 Создание печатной платы

После того как схема была спроектирована, необходимо еще раз проверить правильное подключение всех элементов, после чего в верхнем меню выбрать иконку платы и там пункт «Преобразовать схему в печатную плату», если все в порядке, то среда автоматически откроет редактор печатных плат, отобразит все посадочные места компонентов и связи между ними. После следует разводка печатной платы устройства, первоначально происходит определение размеров устройства и расположение компонентов на ней. Как показано на рисунке 2.17 был определен размер стороны квадрата текстолита, равный 52мм и определены места для микроконтроллера, экрана, коннектора питания платы и коннектора измерительного щупа.

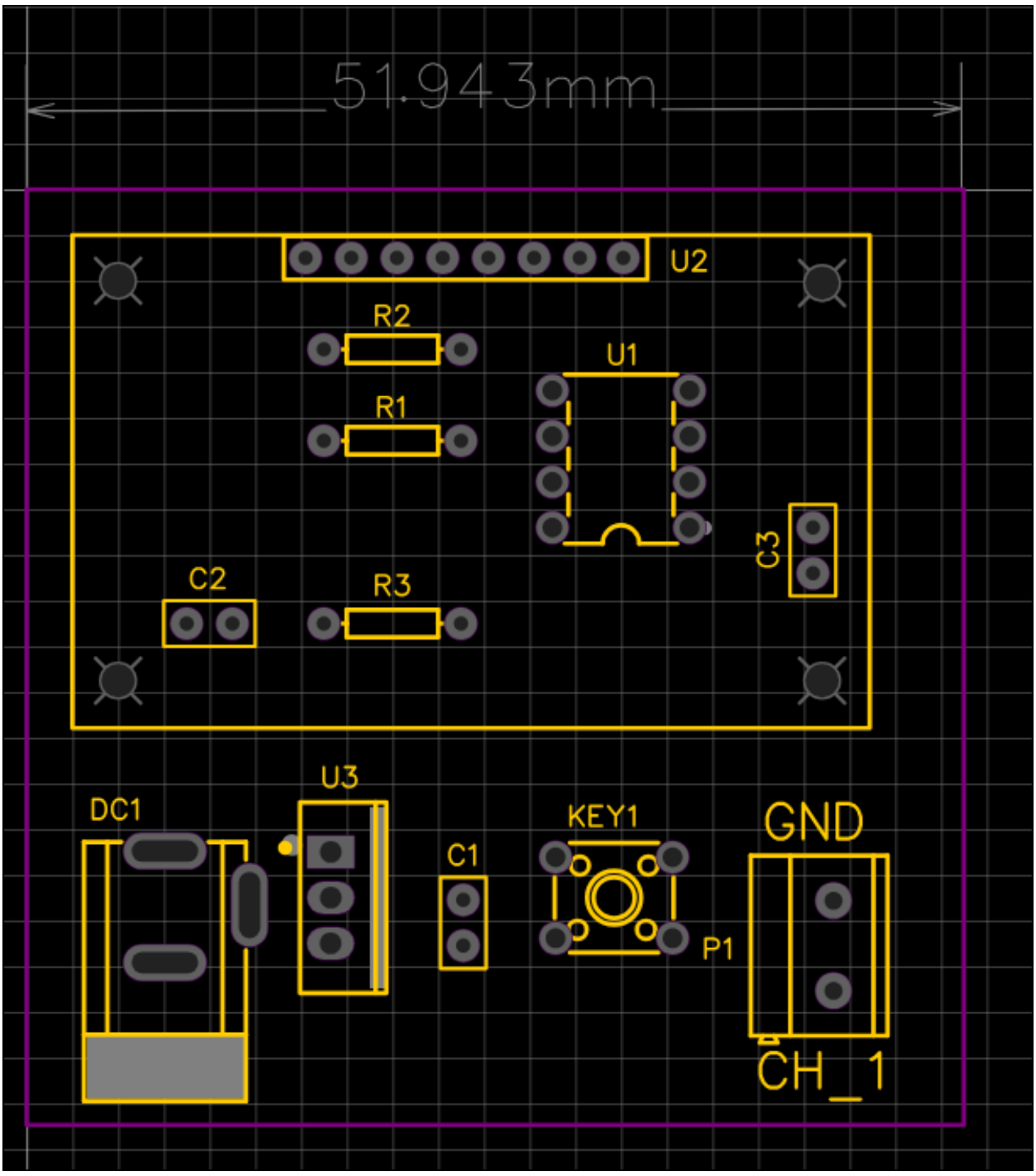


Рисунок 2.17 – Шелкография платы

Подключение коннектора измерительных щупов и разъем внешнего питания устройства вынесены на края текстолита. Для реализации функции аппаратной перезагрузки была вынесена отдельная кнопка, во избежание случайного нажатия и перезагрузки устройства. Подключение щупов реализовано через винтовой разъем, для обеспечения надежного контакта и препятствию непроизвольного извлечения из места фиксации.

На рисунке 2.18 изображена трассировка платы с учетом топологии компонентов, соблюдение толщины дорожек в 1мм.

Для изготовления устройства можно будет применить технологию травления плат ЛУТ методом нанесения трафарета через термотрансферную бумагу на подготовленный текстолит и снятием излишков меди раствором для травления печатных плат, произвести монтаж компонентов согласно разметке и получить готовое устройство, которое можно применить в целях изучения принципов работы осциллографа или модернизации.

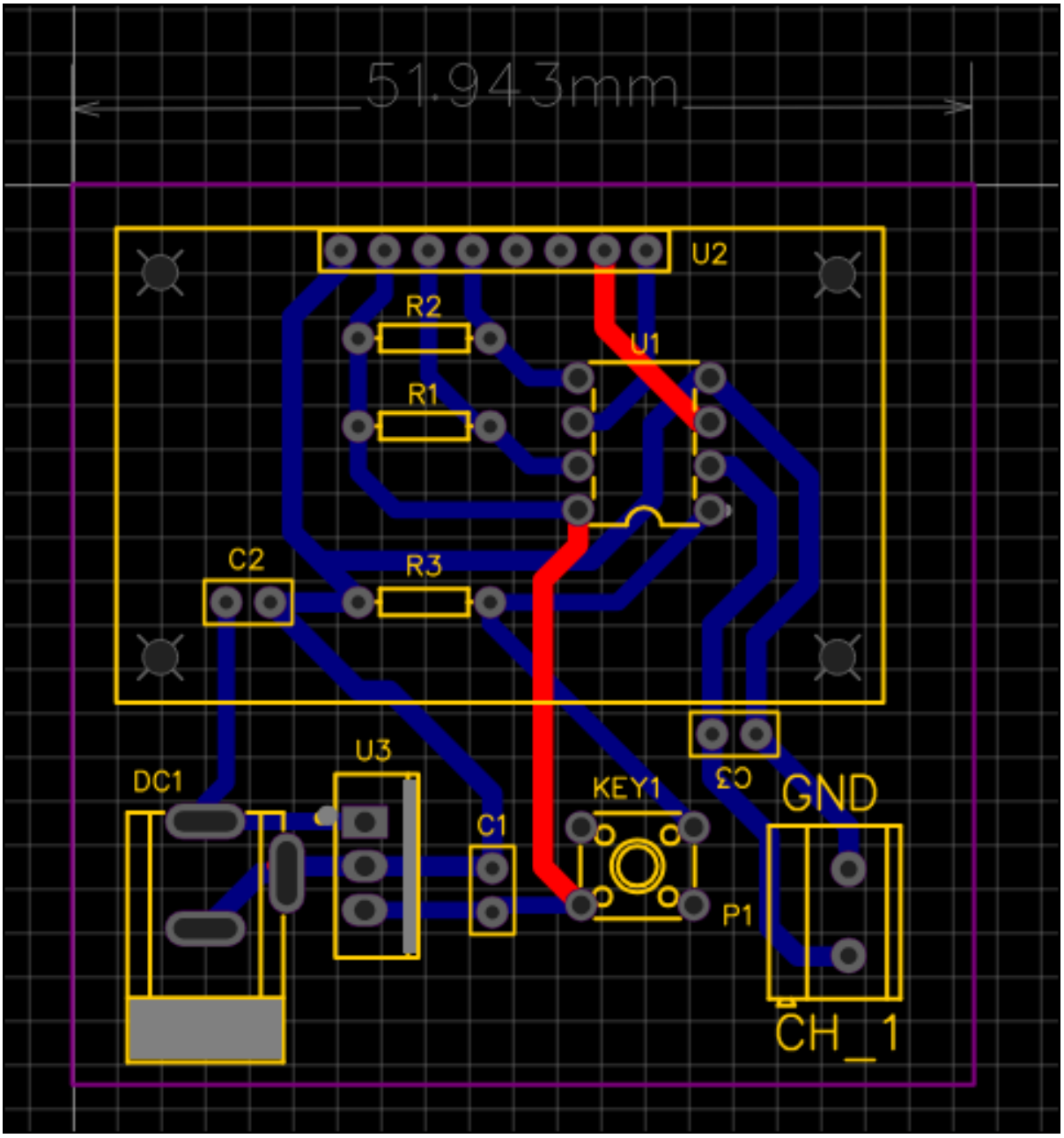


Рисунок 2.18 – Трассировка платы

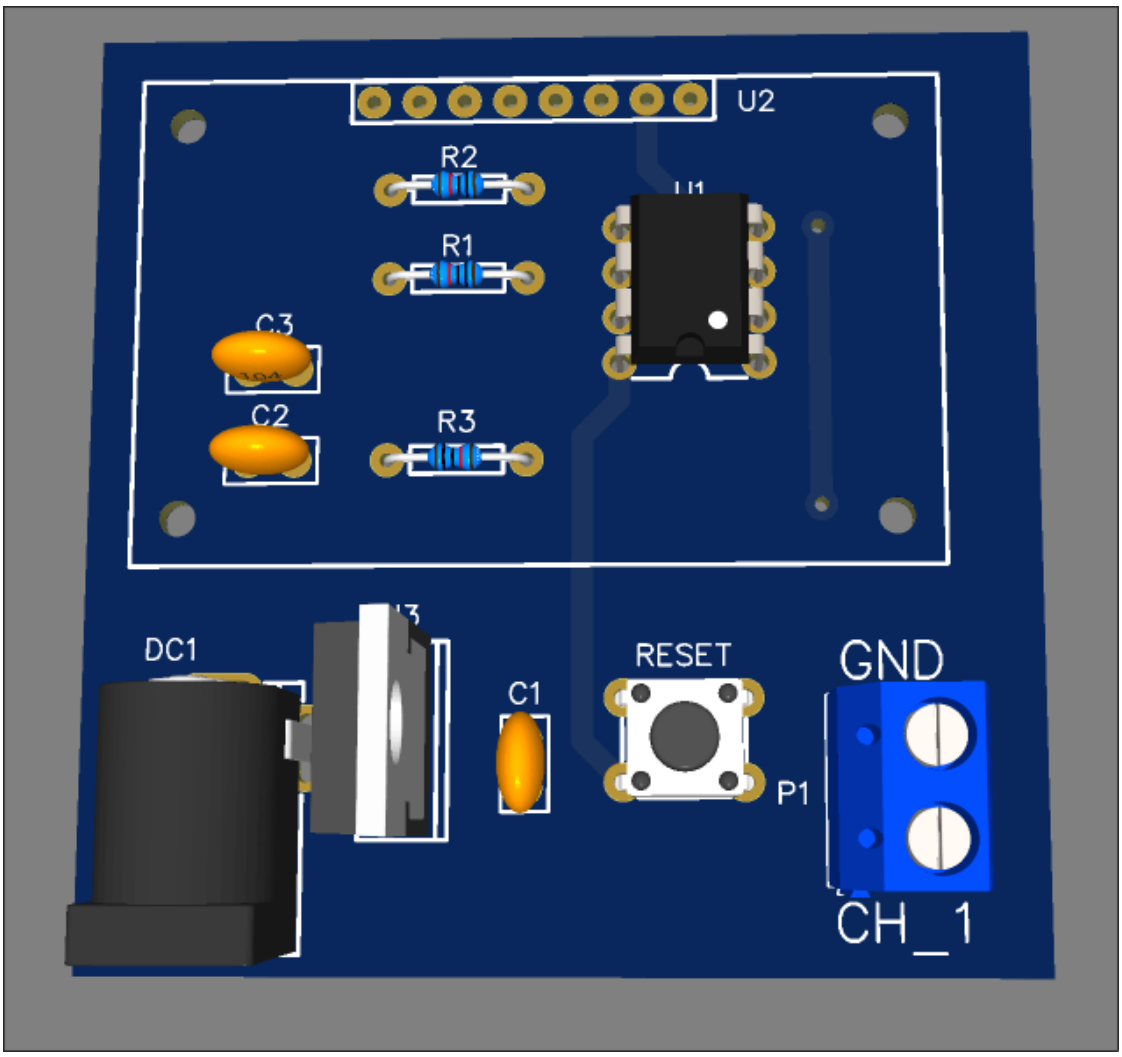


Рисунок 2.19 – 3D модель устройства

2.4 Электрические свойства устройства

В ходе выполнения курсовой работы был создан трафарет печатной платы цифрового осциллографа и приведена принципиальная схема устройства. Технические параметры представлены ниже.

1. Габариты: длина 52мм, ширина 52мм, высота 20мм.
2. Напряжение постоянного источника: от 6 до 10 вольт.
3. Напряжение питания схемы: от 2.5 до 5 вольт.
4. Микроконтроллер: Attiny85-20PU.
5. Максимальный диапазон измерений напряжения: от 0 до 5 вольт.
6. Максимальная частота измеряемого сигнала: 1800 Hz.

ЗАКЛЮЧЕНИЕ

В заключении стоит отметить, что строение осциллографов претерпевало множество изменений и модернизаций, однако сохранялись фундаментальные принципы работы данных устройств. Цифровые осциллографы бесспорно являются одним из важнейших инструментов для промышленности разработки радиокомпонентов, цифровых и аналоговых схем устройств любого назначения.

Все поставленные задачи курсового проекта, связанные с анализом существующих решений, проектированием игрового автомата, а также описанием принципа его работы выполнены. Результаты работы демонстрируют успешное сочетание теоретических знаний и практических навыков, полученных в ходе исследования. На основании проведенного анализа можно сделать вывод о том, что проект имеет перспективы для дальнейшего изучения и реализации, как в образовательной, так и в практической сфере применения, при развитии функционала устройства и его технического усовершенствования.

ПРИЛОЖЕНИЕ А

```
#include <TinyWireM.h>
#include <Tiny4kOLED.h>
#include <GyverButton.h>
#include <EEPROM.h>

// Настройки параметров экрана
#define SCREEN_WIDTH    128 // Ширина экрана в точках
#define SCREEN_HEIGHT   64  // Высота экрана в точках

#define ADC_READ_PIN    PB3 // Пин пробы замеров
#define STORE_BTN_PIN   PB1 // Пин для управление памятью
#define SCALE_BTN_PIN   PB4 // Пин масштабирования изображения
#define MODE_BTN_PIN    PB5 // Пин режима
#define BTN_INTERVAL    500 // Интервал многократных нажатий
#define STORE_SIZE          2     // Размер кольцевого буфера памяти

// Класс для работы с таймером, упрощает логику работы с ним
// сделано для удаления прерываний delay();
class MilTimer {
   private:
      unsigned long tNext = 0, intervalTime;
   public:
      // MilTimer - констуктор класса
      MilTimer(unsigned long time) : intervalTime(time) {}

      // isDone - проверка прохождения таймером интервала
      bool isDone() {
         unsigned long now = millis();
         boolean state = now - tNext >= intervalTime;
         if (state) tNext = now;
         return state;
      }
};
```

```
// Таймер для измерений - 1.5 секунд
MilTimer time(1500);
// Установки парамтеров кнопок управления
GButton modeBtn(MODE_BTN_PIN);      // Кнопка смены режма работы устиройства
GButton storeBtn(STORE_BTN_PIN);       // Кнопка работы с памятью
GButton scaleBtn(SCALE_BTN_PIN);       // Кнопка масштабирования изображения

// Параметры отображения
uint8_t scaleFactor = 1;              // Значение масштабирования графика
uint8_t GraphicBuffer[SCREEN_WIDTH];    // Буфер для экрана в массиве
boolean viewMode = false;   // Режим отображения

// Состояние запроса на сохранение графика.
// По умолчанию - false, при переключении сохраняется буфер в память
boolean saveState = false;

// Индекс сдвига по памяти для работы с ее сегментами
uint8_t memorySaveShift = 0;

// Переключение секции памяти через кольцевой буфер
void incrementSaveShift() {
        memorySaveShift = (memorySaveShift + 1) % STORE_SIZE;
}

void setup() {
   // Инициализируем экран отображения
   oled.begin(
      128, 64, sizeof(tiny4koled_init_128x64br),
      tiny4koled_init_128x64br
   );
   oled.clear(); oled.on(); // Первичная очистка и включение подсветки дисплея
   // Настройки интервалов регистрирования многократных нажатий
   modeBtn.setTimeout(500); storeBtn.setTimeout(500); scaleBtn.setTimeout(500);
}
```

```
void loop() {
  // Обработка кнопок в начале каждого цикла
  modeBtn.tick();
  storeBtn.tick();
  scaleBtn.tick();

  // Масштабируем изображение
  if (scaleBtn.isSingle()) { scaleFactor = scaleFactor + 1; }

  // Сброс масштаба на экране
  if (scaleBtn.isHold()) { scaleFactor = 1; }

  // Сброс EEPROM
   if (scaleBtn.isTriple()) {
      for (uint8_t i = 0; i < 128 * STORE_SIZE; i++) EEPROM.write(i, 0);
   }

  // Смена режима отображения из памяти или чтения пробы
  if (modeBtn.isHold()) { viewMode = !viewMode; return; }
  if (modeBtn.isSingle()) { incrementSaveShift(); return; }
   // Сохраняем уже отображаемый фрейм с графиком в буфер, если стоит режим чтения канала
  // Задается состояние переменной статуса сохранения, сохранение проихсодит на следующем цикле
  // Сохраняется уже отображенный фрейм графика, не новый
  if (storeBtn.isHold() && !viewMode) { saveState = true; return; }

  // Проверка на режим чтения памяти, если да, то делаем отображение по запросу
  if (storeBtn.isSingle() && viewMode) {
    readADCfromMEM(GraphicBuffer);
    oledDisplay(GraphicBuffer, scaleFactor);
  }
```

```
// Стандартное отображение графика
    if (!viewMode && time.isDone()) {
        if (saveState) {
            saveState = false;
            saveADCtoMEM(GraphicBuffer);
        } else {
            readADC(GraphicBuffer);
            oledDisplay(GraphicBuffer, scaleFactor);
        }
    }
}

// readSavedADC - читает из памяти со сдвигом в буфер для экрана
void readADCfromMEM(uint8_t *arr) {
    for (uint8_t x = 0; x < SCREEN_WIDTH; x++) {
        arr[x] = EEPROM.read(memorySaveShift * 128 + x);
    }
}

// saveADCtoMEM - сохраняет из буффера в память
void saveADCtoMEM(uint8_t *arr) {
    for (uint8_t x = 0; x < SCREEN_WIDTH; x++) {
        EEPROM.write(memorySaveShift * 128 + x, arr[x]);
    }
}

// readADC - формирует массив значений ADC
void readADC(uint8_t *arr) {
    for (uint8_t x = 0; x < SCREEN_WIDTH; x++) {
        uint16_t value = analogRead(ADC_READ_PIN);
        arr[x] = uint8_t(value >> 4); // Сдвиг битов вправо на 4
    }
}
```

```
// oledDisplay - отобрпажаем массив в память дисплея
void oledDisplay(uint8_t* valueArr, uint8_t scale) {
    oled.clear();
    for (uint8_t x = 0; x < SCREEN_WIDTH; x++) {
        uint8_t y = valueArr[x / scale];
        setOledDot(x, 31); // Отрисовка центральной линии
        setOledDot(x, y);
    }
}

// setOledDot - установки точки на нужной координате
void setOledDot(uint8_t x, uint8_t y) {
    // Ограничиваем значение по высоте отрисовки
    y = (y > 63) ? 63 : y;

    // Определяем страницу (каждая страница 8 пикселей)
    uint8_t page = y / 8;
    // Определяем бит в странице
    uint8_t shiftBit = 1 << (y % 8);

    // Устанавливаем страницу пространства экрана
    oled.setCursor(x, page);

    // Устанавливаем бит в позицию
    oled.startData();
    oled.sendData(shiftBit);
    oled.endData();
}
```

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Осциллографы Основные принципы измерений: Учебное пособие – с.Tektronix: 60 с.

2 Современная осциллография и осциллографы – Дьяконов В.П.: 322с.

3 Сайт форума для радиолюбителей. Форма доступа: https://cxem.net/izmer/izmer77.php.

4 Карманный осциллограф на микроконтроллере STC8051. Форум разработчиков и IT технологий Хабр. Форма доступа: https://habr.com/ru/companies/ruvds/articles/831634.