



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

по дисциплине «Микропроцессорные системы»

НА ТЕМУ:

Микропроцессорная система контроля
попадания волана в заданную зону

Студент

ИУ6-72Б

(Группа)

(Подпись, дата)

А.А. Морозова

(И.О. Фамилия)

Руководитель

(Подпись, дата)

А.А. Сотников

(И.О. Фамилия)

2021 г.

РЕФЕРАТ

РПЗ 42с., 23 рис., 5 табл., 21 источник, 2 прил.

VELOSTAT, ARDUINO, ДАТЧИК ДАВЛЕНИЯ, БАДМИНТОН

Объектом разработки является микропроцессорная система контроля попадания волана в заданную зону.

Цель работы – разработка и сборка макетного образца микропроцессорной системы контроля попадания волана в заданную зону.

При проектировании решены следующие задачи: анализ объекта разработки на функциональном уровне, разработка функциональной схемы модуля, выбор элементной базы для реализации объекта, разработка принципиальной схемы модуля, расчет электрических параметров.

Результатом проектирования является комплект документации, включающий в себя:

1. Схему электрическую функциональную;
2. Схему электрическую принципиальную;
3. Спецификацию;
4. Листинги кода.

Устройство обладает следующими техническими характеристиками:

Использование материалов	медь, Velostat
Тактовая частота	16 МГц
Мощность потребления	не более 1.5 Вт

Условия эксплуатации:

- Температура окружающей среды 17-20 °C
- Относительная влажность воздуха 40-60%

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ.....	5
ВВЕДЕНИЕ.....	6
1 Проектирование.....	7
1.1 Принцип работы устройства.....	7
1.2 Разработка электрической функциональной схемы устройства.....	9
1.3 Выбор элементной базы.....	10
1.4 Разработка электрической принципиальной схемы устройства.....	17
1.5 Расчет потребляемой мощности.....	20
3 Программирование макетного образца.....	22
4 Работа макетного образца.....	29
ЗАКЛЮЧЕНИЕ.....	32
ПРИЛОЖЕНИЕ А. Спецификация.....	35
ПРИЛОЖЕНИЕ Б. Программный код.....	37

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

АЦП — аналого-цифровой преобразователь.

ЦАП — цифро-аналоговый преобразователь.

COM-порт (communications port) — последовательный порт.

I2C (Inter-Integrated Circuit) — последовательная асимметричная шина.

PWM (pulse-width modulation) — ШИМ, широтно-импульсная модуляция.

SPI (Serial Peripheral Interface) — последовательный периферийный интерфейс.

UART (Universal Asynchronous Receiver/Transmitter) — универсальный асинхронный приёмопередатчик.

USB (Universal Serial Bus) — универсальная последовательная шина.

USART (Universal Synchronous/Asynchronous Receiver/Transmitter) — универсальный синхронный/асинхронный приемник/передатчик.

ВВЕДЕНИЕ

В данной работе изложены основные этапы разработки микропроцессорной системы контроля попадания волана в заданную зону.

В настоящее время достижения науки плотно вошли в повседневную жизнь человека, упрощая её, позволяя осуществлять то, что раньше казалось невозможным. «Умный дом» уже не вызывает удивления, а различные технологии позволили разнообразить досуг. Особенно отчетливо это можно увидеть в спорте. Уже сложно представить, чтобы человек вышел на пробежку без приложения, отслеживающего пройденную дистанцию. «Умные часы» помогают считывать пульс и давление, отслеживать состояние, собирать статистику.

Современные технологии используются в различных видах спорта, позволяя не только улучшить качество подготовки, но и разнообразить тренировочный процесс. Актуальность разработанного устройства обуславливается тем, что наличие чувствительного к давлению мата позволяет помочь в тренировках в тех видах спорта, где важна точность, в частности, в бадминтоне.

Разработанное устройство обладает следующими ключевыми характеристиками: реагирует на нажатия и, в соответствие с местом, куда было приложено давление, включает соответствующий светодиод — зеленый или красный.

1 Проектирование

1.1 Принцип работы устройства

В микропроцессорной системе контроля попадания волана в заданную зону можно выделить две основные части — устройство, состоящее из микроконтроллера и других микросхем, и датчик давления, созданный на основе материала Velostat и меди.

Velostat представляет собой непрозрачную объемно-проводящую пленку, наполненную полиолефином. Активное сопротивление материала уменьшается при приложении давления. Если материал расположить между двумя проводящими слоями, можно использовать его в качестве датчика давления.

В данном устройстве в качестве проводящих слоёв была использована медная лента. Сам мат представляет из себя следующее: лист материала Velostat располагается между рядами лент из меди. К ним припаиваются провода. К одному слою поступает питание, провода со второго слоя поступают на аналоговые входы собранной схемы.

Общая схема мата выглядит примерно так, как показано на рисунке 1.

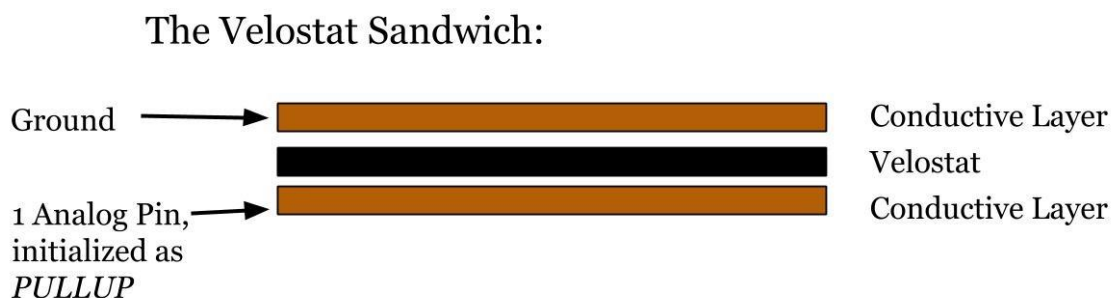


Рисунок 1 — Основа мата

Принцип работы заключается в следующем. С платы микроконтроллера с помощью подачи напряжения на цифровые выходы производится выборка текущей горизонтальной линии, которая будет подключена к питанию. Затем производится последовательная выборка 16-ти вертикальных линий. Если к мату приложено давление, в одной из этих точек напряжение будет ненулевым.

Данная выборка производится для всех горизонтальных линий, на которые последовательно подается напряжение. Если изображать схематически, весь процесс выглядит так, как представлено на рисунке 2.

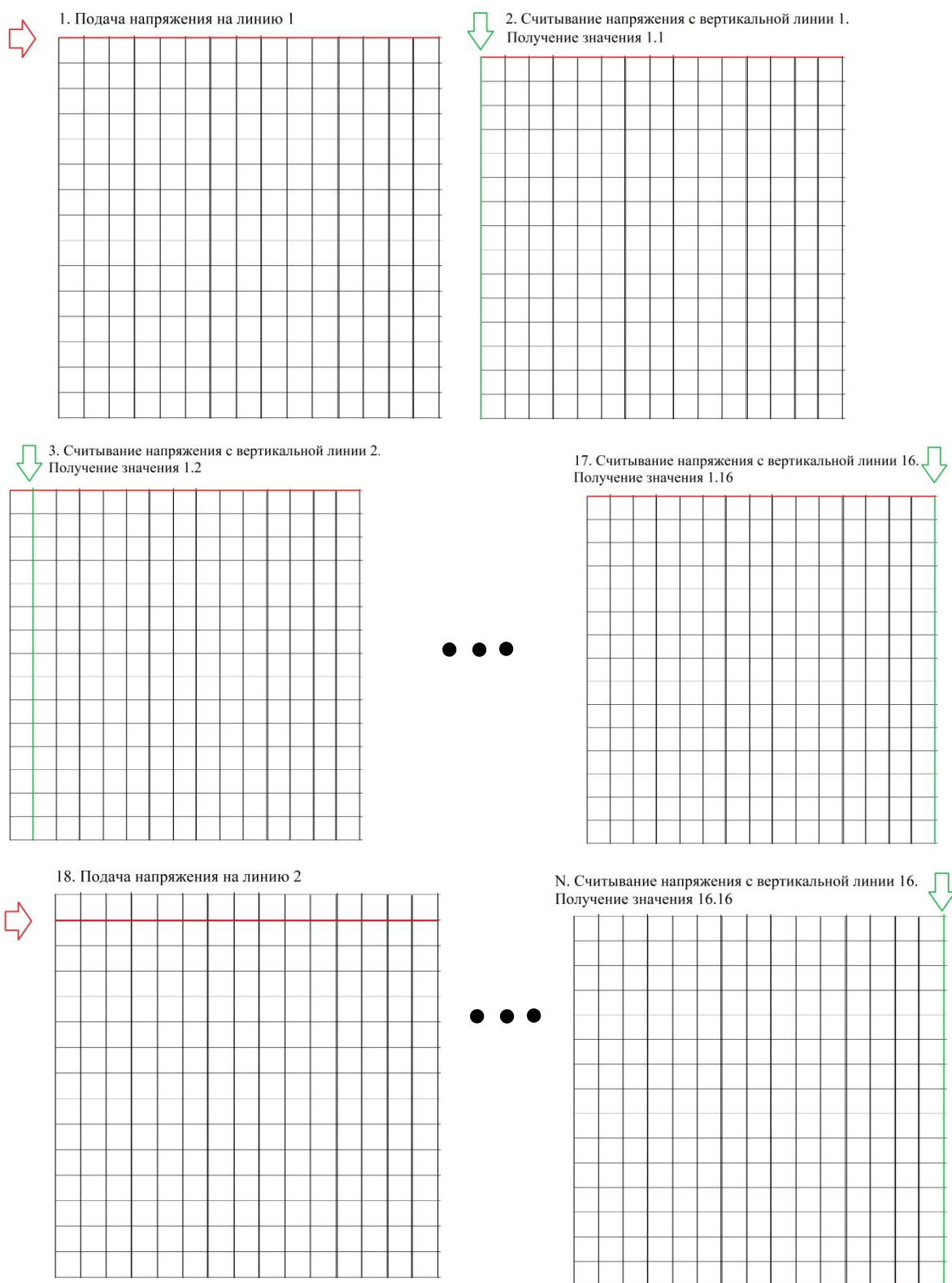


Рисунок 2 — Выборка точек

1.2 Разработка электрической функциональной схемы устройства

Чтобы определять линии, на которые подается питание в определенный момент времени, было принято решение использовать демultipлексор (для подачи питания на выбранную линию) и мультиплексор (для считывания напряжения с выбранной линии). Также использование этих двух устройств позволило использовать более компактные (и более дешевые) модели программируемых плат. Обычно эти модели обладают меньшим количеством цифровых и аналоговых входов и выходов и расширяются при помощи дополнительных модулей, в том числе мультиплексоров и демultipлексоров.

Программирование микроконтроллера производилось через преобразователь USB-UART, макетный образец подключался к компьютеру/ноутбуку. После загрузки программы питание осуществлялось от батарейки.

На основании приведенных выше данных были определены следующие блоки:

1. Источник питания — подача напряжения на макетный образец.
2. Микроконтроллер — управляющее устройство.
3. Мультиплексор и демultipлексор — блоки выборки линий мата.
4. Чувствительный к давлению мат — собранный из материала Velostat, линий медной ленты и непроводящего защищающего материала (в самой простой версии используется бумага) датчик.
5. Светодиоды — блок вывода результатов для пользователя.

Разработанная функциональная схема макетного образца представлена на рисунке 3.

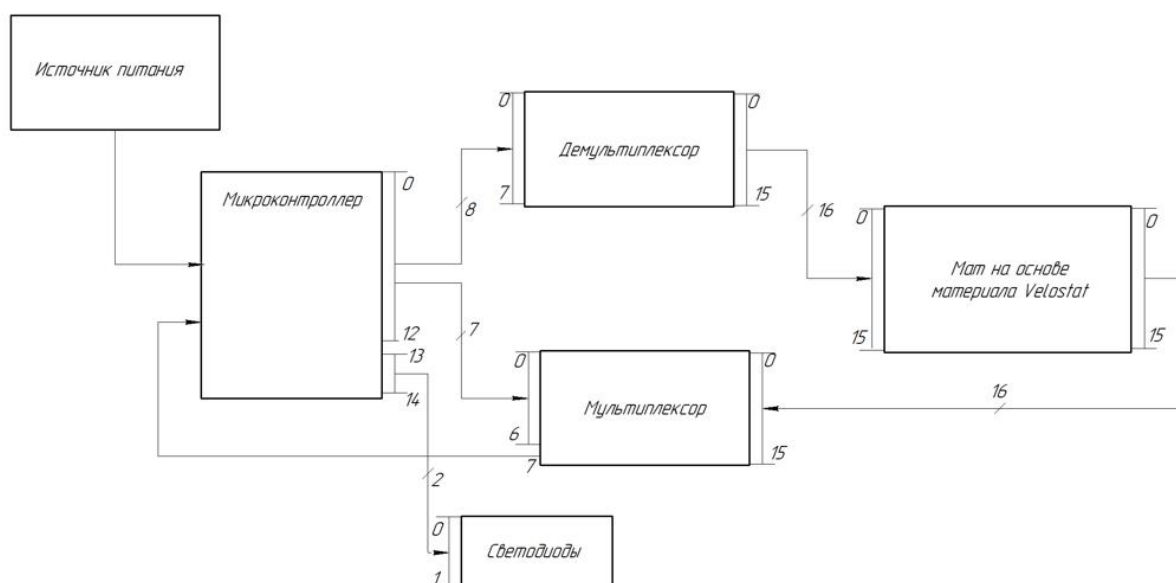


Рисунок 3 — Функциональная схема

Источник питания подает напряжение на схему. У микроконтроллера используется 15 выходов — 13 для мультиплексора и демультиплексора и 2 для светодиодов. Для демультиплексора используется 6 своих выходов (S0...S3, SIG, EN) и 2 общих с мультиплексором (VCC и GND). Для мультиплексора — выходы S0...S3, EN (свои), VCC и GND (общие). SIG от мультиплексора поступает на вход микроконтроллера.

Для блока светодиодов используется два выхода. Определенный светодиод включается в зависимости от того, к какому выходу подали напряжение.

1.3 Выбор элементной базы

Выбирая элементную базу, учли потребляемую мощность микросхем, а также их стоимость.

Для платы с микроконтроллером выбор производился среди линейки Arduino — это одна из наиболее доступных фирм для макетирования систем в домашних условиях.

Линейка Arduino представлена широким спектром моделей, начиная с Arduino Mega 2560, у которой 54 цифровых выводов, заканчивая Arduino Nano, у которой всего 14 цифровых пинов. В классической линейке устройств в

основном применяются микроконтроллеры семейства Atmel AVR — ATmega2560, ATmega32U4, ATmega328, ATtiny85, ATmega168.

Для данной микропроцессорной системы было принято решение использовать модель Arduino Pro Mini с микроконтроллером ATmega328. Среди остальной линейки серии эта плата отличается маленьким размером. Также Arduino Pro Mini обладает удовлетворительным соотношением цены к техническим требованиям для разработанного мата.

Данная модель обладает характеристиками, представленными в таблице 1.

Таблица 1 — Характеристики платформы Arduino Pro Mini

Характеристика	Значение
Рабочее напряжение	5 В
Входное напряжение	5-12 В
Цифровые входы/выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	8
Постоянный ток через вход/выход	40 мА
Флеш-память	32 Кб (2 используются для загрузчика)
ОЗУ	1 Кб
EEPROM	512 байт
Тактовая частота	16 МГц
Рабочая температура	-40..85°C

Структурная схема микроконтроллера ATmega328 представлена на рисунке 4.

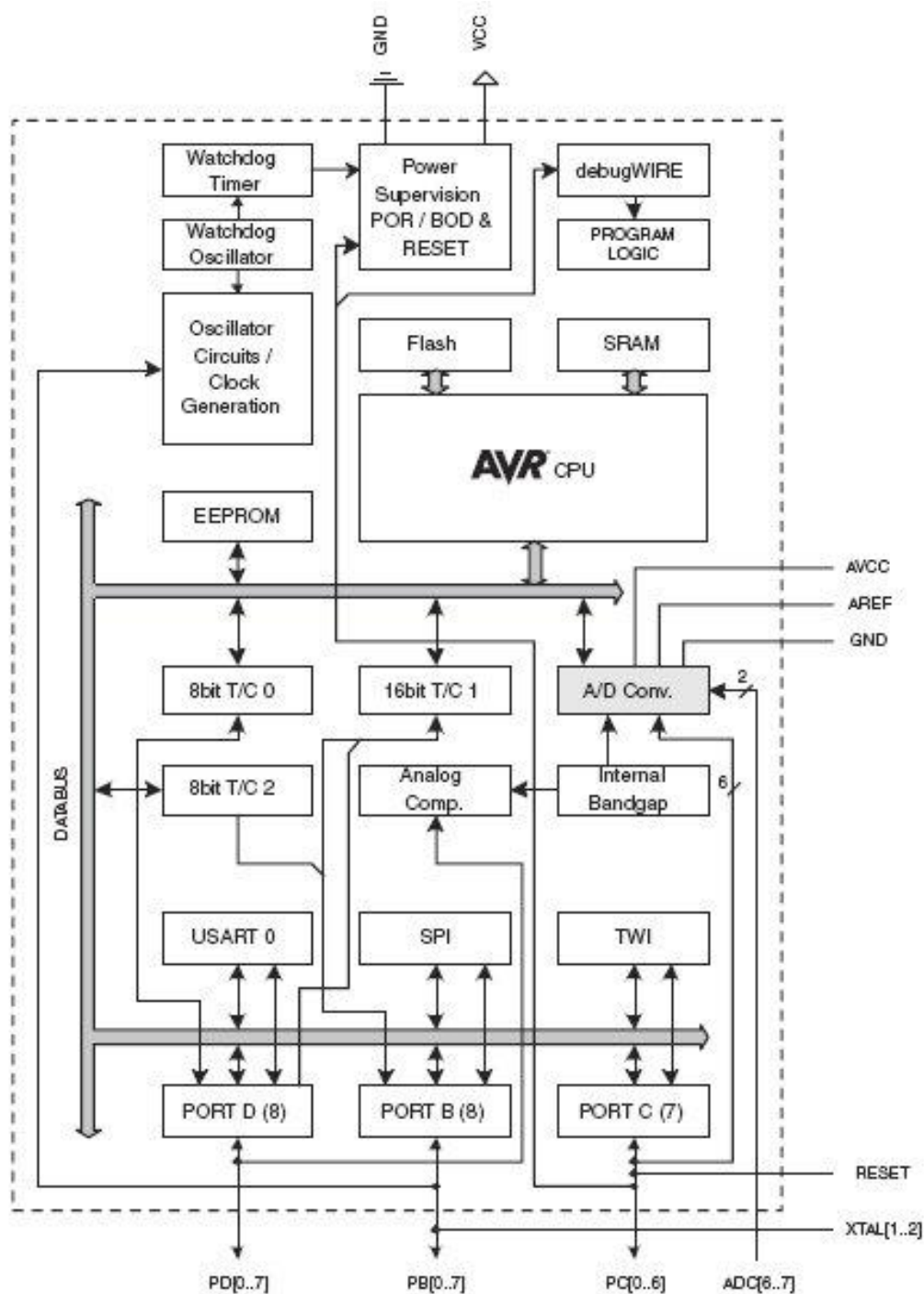


Рисунок 4 — Структурная схема микроконтроллера ATmega328

Микроконтроллер обладает следующими периферийными устройствами:

1. Два 8-битных таймера/счетчика с модулем сравнения и делителями частоты.
2. 16-битным таймером/счетчиком с модулем сравнения и делителем частоты, а также с режимом записи.
3. Счетчиком реального времени с отдельным генератором.
4. Шестью каналами PWM (аналог ЦАП).

5. 6-канальным ЦАП со встроенным датчиком температуры.
6. Программируемым последовательным портом USART.
7. Последовательным интерфейсом SPI.
8. Интерфейсом I2C.
9. Программируемым сторожевым таймером с отдельным внутренним генератором.
10. Внутренней схемой сравнения напряжений.
11. Блоком обработки прерываний и пробуждения при изменении напряжений на выводах микроконтроллера.

Также микроконтроллер обладает специальными функциями:

1. Сбросом при включении питания и программным распознаванием.
2. Внутренним калибруемым генератором тактовых импульсов.
3. Обработкой внутренних и внешних прерываний.
4. 6 режимами сна (пониженным энергопотреблением и снижением шумов для более точного преобразования АЦП).

Плату Arduino Pro Mini можно питать несколькими способом — подавать стабилизированное напряжение 5 В (через USB и преобразователь USB-UART или через вывод VCC) или подавать нестабилизированное напряжение от источника питания — батарейки, зарядного устройства.

Внешний вид платы представлен на рисунке 5.

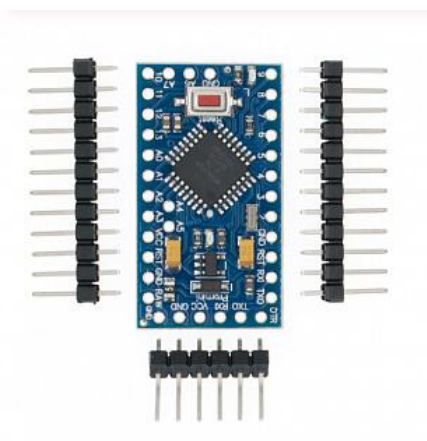


Рисунок 5 — Внешний вид платы Arduino Pro Mini

Каждый из 14 цифровых выводов может работать как в режиме входа, так и в режиме выхода. Максимальный ток через один вывод составляет 40 мА. Все

выводы сопряжены с внутренними подтягивающими резисторами номиналом 20-50 кОм. Также некоторые выводы Arduino могут выполнять дополнительные функции:

1. Последовательный интерфейс: выводы 0 (RX) и 1 (TX). Используются для получения (RX) и передачи (TX) данных по последовательному интерфейсу. Эти выводы соединены с линиями TX-0 и RX-1 шестиконтактного разъема.

2. Внешние прерывания: выводы 2 и 3. Данные выводы могут служить источниками прерываний, возникающих при различных условиях: при низком уровне сигнала, по фронту, по спаду или при изменении сигнала на этих выводах.

3. ШИМ: выводы 3, 5, 6, 9, 10 и 11.

4. Интерфейс SPI: выводы 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Данные выводы позволяют осуществлять связь по интерфейсу SPI.

5. Светодиод: 13. Встроенный светодиод, подсоединенный к цифровому выводу 13. При отправке значения HIGH светодиод включается, при отправке LOW - выключается.

В Arduino Pro Mini есть 6 аналоговых входов, каждый из которых может представить аналоговое напряжение в виде 10-битного числа (1024 различных значения). Четыре из них выведены на контакты возле края платы; два остальных (входы 4 и 5) выведены на распаечные площадки внутри платы. Измерение напряжения осуществляется относительно диапазона от 0 до VCC. Помимо этого, некоторые из аналоговых входов имеют дополнительные функции:

1. I2C: вывод A4 (SDA) и вывод A5 (SCL). С использованием библиотеки Wire позволяют осуществлять связь по интерфейсу I2C.

2. Reset. Формирование низкого уровня (LOW) на этом выводе приведет к перезагрузке микроконтроллера.

Как было указано выше, Arduino Pro Mini обладает 6 аналоговыми выводами. В случае разработанного макета происходит подача питания и считывание информации с 32 линий (по 16 на слой). Даже у самой большой

платы линейки Arduino, Arduino Mega, есть только 16 аналоговых входов. Для увеличения их количества было принято решение использовать мультиплексор и демультимплексор.

Наиболее часто используемой в системах Arduino микросхемой является 16-канальный мультиплексор/демультимплексор CD74HC4067. Он также поставляется на плате, что облегчает его подключение к микроконтроллеру Arduino. Внешний вид CD74HC4067 представлен на рисунке 6.

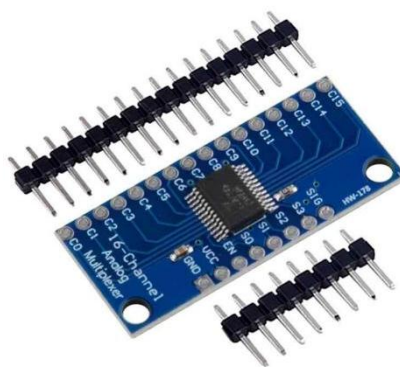


Рисунок 6 — 16-канальный мультиплексор/демультимплексор CD74HC4067

Технические параметры устройства представлены в таблице 2.

Таблица 2 — Характеристики микросхемы CD74HC4067

Характеристика	Значение
Напряжение питания	2-6 В
Внутреннее сопротивление ключей	70 Ом при напряжении 4.5 В
Скорость срабатывания	6 нс при напряжении 4.5 В
Диапазон рабочих температур	-55..125 °С

Технические параметры устройства подходят под условия работы разработанного макетного образца.

В схеме были использованы светодиоды GNL-3012HD с рабочей температурой -40..80 °С с постоянным прямым током 20 мА и постоянным напряжением 2.1 В (сопротивление 105 Ом). Эта часть цепи питается от 5 В. Для понижения силы тока по формуле 1 рассчитано сопротивление.

$$R = \frac{U}{I} = \frac{(5 - 2.1) \text{ В}}{20 \text{ мА}} \approx 145 \text{ Ом} \quad (1)$$

В данном случае было допустимо использовать резисторы с сопротивлением 150 Ом. Для макета были выбраны резисторы металлопленочные MF-25 (C2-23) с рабочей температурой -55...155 °С и точностью 1%.

В отсутствие приложенного давления с АЦП получаются значения приблизительно в 800 единиц. Тогда можно рассчитать примерное напряжение на входе по формуле 2.

$$V_{\text{IN}} = \frac{\text{ADC} \cdot V_{\text{REF}}}{1024}, \quad (2)$$

ADC — значение с АЦП, V_{REF} — напряжение на выводе, 5 В. Тогда $V_{\text{IN}} = 3.85 \text{ В}$.

Помимо того, чтобы просто определять присутствие приложенного давления, полезно было бы уметь отличать силу более сильного нажатия от более слабого. Однако в данном диапазоне (от 800 до 1023) это не представлялось возможным.

Для того, чтобы убрать из цепи часть напряжения, было принято добавить резисторы в 1 кОм на выводах демultipлексора. Были использованы резисторы углеродистые CF-25 (C1-4) с сопротивлением 1 кОм и погрешностью 5%, с рабочей температурой -55...155 °С. Таким образом, в состоянии покоя получились значения, близкие к нулевым — 10-20 единиц, то есть около 68 мВ. При приложенном к Velostat давлении были получены значения, близкие к 800 единицам, то есть около 3.85 В.

Итоговая спецификация представлена в приложении А.

1.4 Разработка электрической принципиальной схемы устройства

Часть принципиальной схемы с микроконтроллером, его обвязкой и блоком светодиодов представлена на рисунке 7.

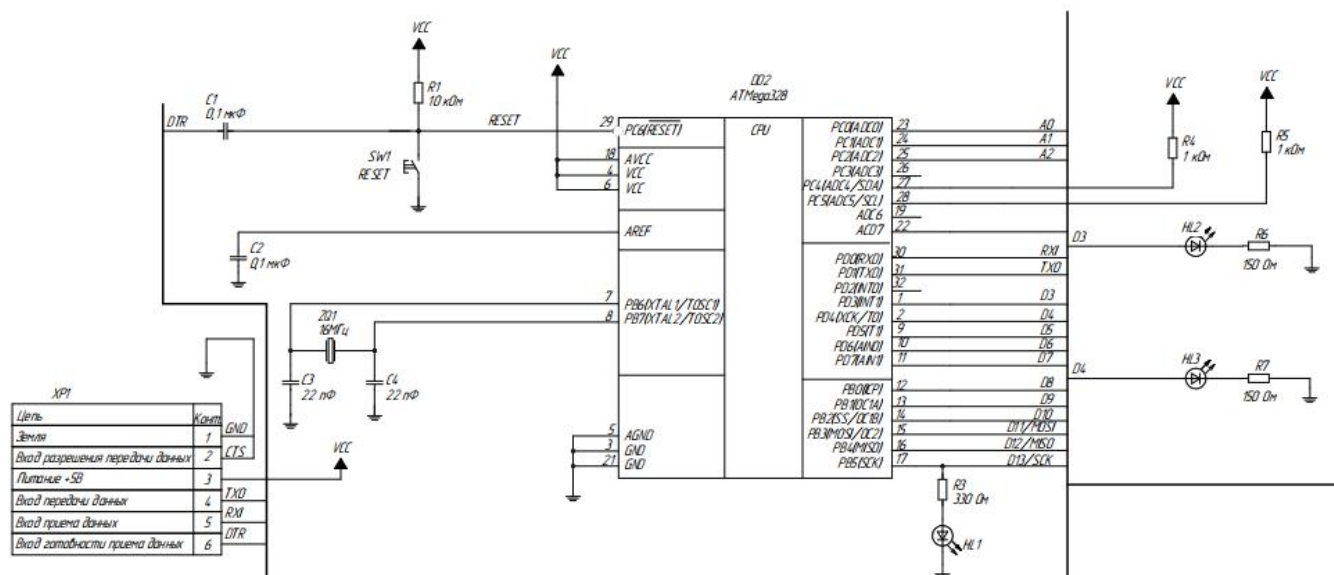


Рисунок 7 — ATmega328 с обвязкой и блоком светодиодов

На данном рисунке можно увидеть элементы, которые принадлежат самой плате Arduino Pro Mini и которые легко можно на ней найти, к примеру, кнопку сброса SW1, светодиод HL1.

Также на рисунке 7 изображен разъем XP1, предназначенный для вывода на преобразователь USB-UART FTDI Basic. Выводы TXO и RXI предназначены для обмена данными, с помощью VCC и GND подключено питание (стабилизированное, 5 В), также присутствуют выводы для сигналов DTR и CTS.

На рисунке 8 представлена часть схемы, содержащая стабилизатор напряжения.

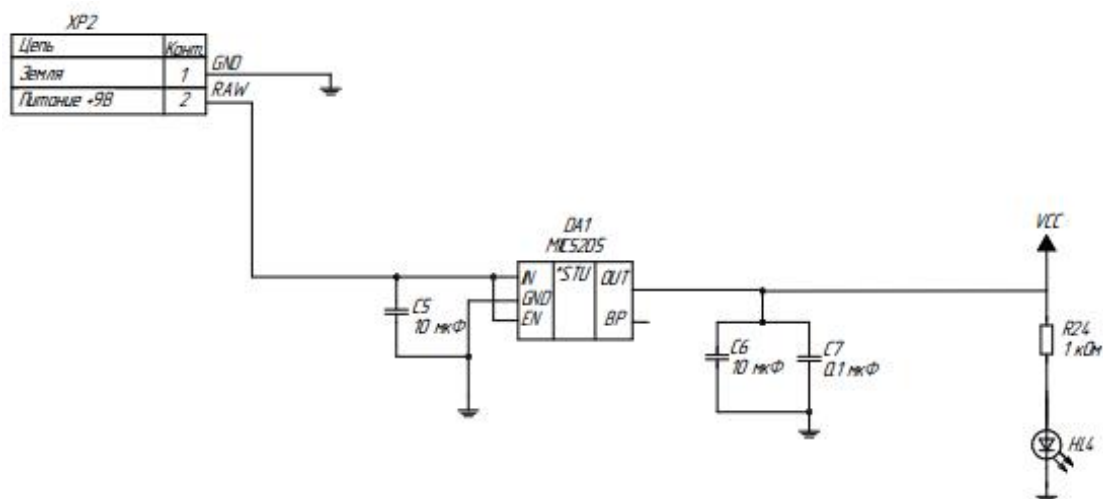


Рисунок 8 — Стабилизатор напряжения

Разъем XP2 используется для подключения нестабилизированного напряжения 9 В, к примеру, батарейки «Крона». С помощью MIC5205 производится понижение напряжения до 5 В, светодиод HL4 присутствует для индикации наличия питания на плате. Обвязка из конденсаторов позволяет погасить лишние шумы.

Блок выборки с демультиплексором представлен на рисунке 9.

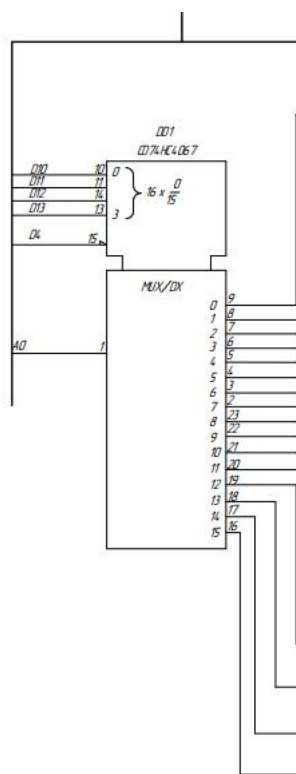


Рисунок 9 — CD74HC4067 в режиме демультиплексора

Элемент CD74HC4067 может работать как в режиме демультиплексора, так и в режиме мультиплексора. К демультиплексору с микроконтроллера поступают сигналы D10, D11, D12, D13, с помощью которых производится выборка нужного канала (0...15), а также A0, который передается на выбранный канал. Сигнал D4 запрещает или разрешает работу элемента.

Часть схемы с датчиком давления представлена на рисунке 10.

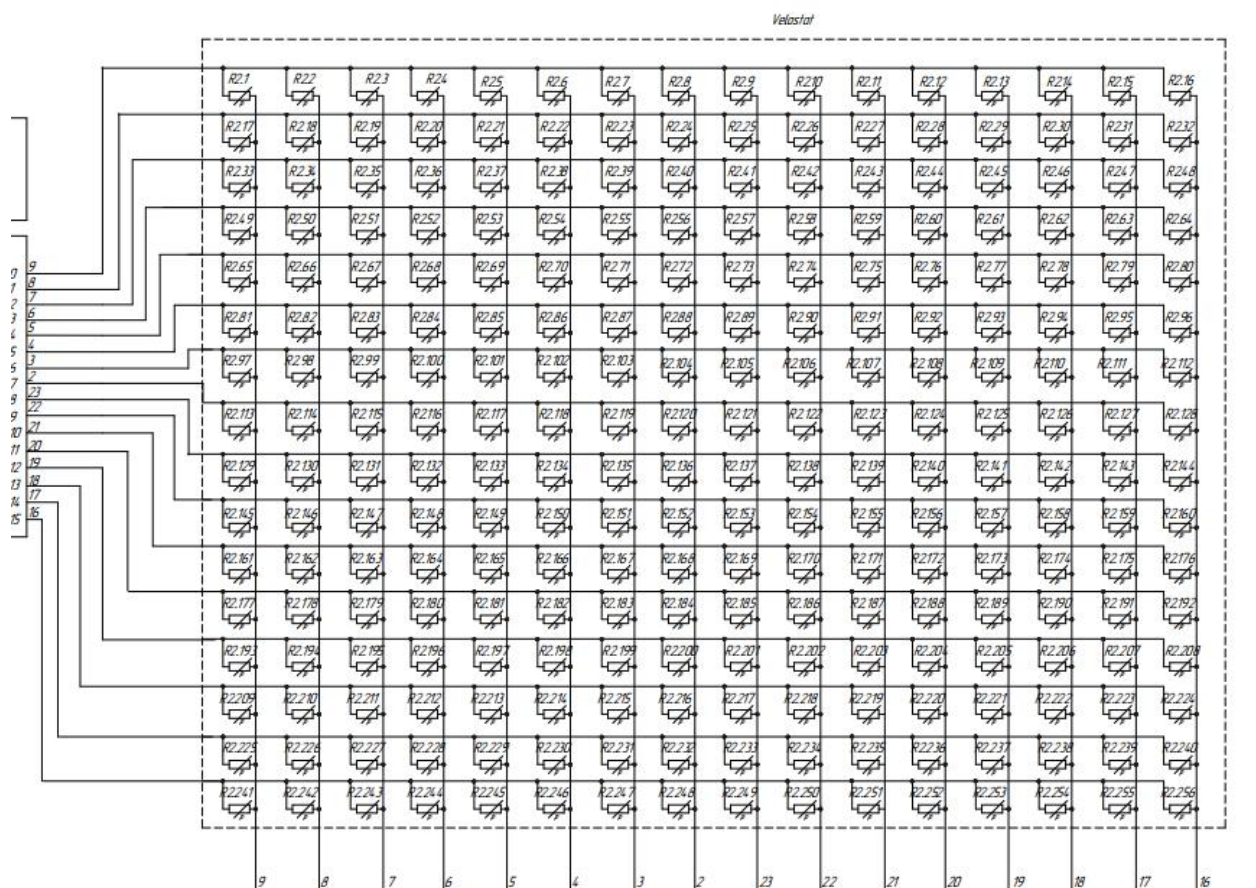


Рисунок 10 — Чувствительный к давлению мат

Было принято решение обозначить материал Velostat как матрицу тензорезисторов, так как его сопротивление меняется в зависимости от приложенного давления.

Так, для того, чтобы получить значение напряжения с резистора R2.19, необходимо подать питание на вывод 1 (сигнал 8) демультиплексора и считать сигнал 7.

Блок выборки с демультиплексором представлен на рисунке 11.

$$P = \frac{U^2}{R} \quad (3)$$

Таблица 3 — Мощности резисторов

Элемент	Количество	Значение мощности, Вт	Σ , Вт
MF-25	2	0.167	0.334
CF-25	16	0.025	0.4

Потребляемая мощность схемы рассчитывается по формуле 4.

$$P_{\Sigma} = \Sigma P_S + \Sigma P_D \quad (4)$$

Статическую мощность можно рассчитать по формуле 5, динамическую — по формуле 6.

$$P_S = U_{cc} * I_{cc} \quad (5)$$

$$P_D = C_{PD} * U_{cc}^2 * F_{IN} \quad (6)$$

Значения для CD74HC4067 вынесем в таблицу 4.

Таблица 4 — Подсчет мощности микросхемы CD74HC4067

U_{cc} , В	I_{cc} , μA	P_S , мВт	C_{PD} , пФ	F_{IN} , МГц	P_D , мВт	N	P_{Σ} , Вт
5	160	0.8	29	16	11.6	2	0.025

Arduino Pro Mini при питании от 9 В потребляет 0.162 Вт, при питании от 5 В — 0.085 Вт. Среди линейки Arduino данная плата обладает наименьшим энергопотреблением.

Мощности для светодиодов были рассчитаны из вывода формулы 3.

Общая сводка мощностей собрана в таблице 5.

Таблица 5 — Подсчет мощности электрической схемы

Элемент	Мощность, Вт
Arduino Pro Mini	0.162
CD74HC4067	0.025
MF-25	0.334
CF-25	0.4
GNL-3012HD	0.084
Σ	1.005 Вт

3 Программирование макетного образца

Язык программирования Arduino называется Arduino C и представляет собой язык C++ с фреймворком Wiring. Он имеет некоторые отличия по части написания кода, который компилируется и собирается с помощью avr-gcc, с особенностями, облегчающими написание работающей программы — имеется набор библиотек, включающий в себя функции и объекты.

Программирование микроконтроллера ATmega328 было произведено с помощью среды разработки Arduino IDE.

Все микроконтроллеры от Arduino имеют специальный загрузчик, созданный на основе Atmel AVR Application Note AN109. Этот загрузчик может работать через интерфейс USB, с его помощью производилось программирование.

В алгоритме работы можно выделить два блока — считывание значений и анализ полученных результатов.

В первом блоке в цикле для каждой линии необходимо было совершить следующие действия:

1. Подать напряжение на линию питания.

2. Считать 16 значений напряжения с 16 точек, заполнить логическую матрицу — матрицу из 256 значений, в которой «1» означает значение выше некоторого минимального значения напряжения, «0» же означает значение ниже этого.

Дополнительно для отладки сохраняется матрица значений напряжения, которые мат считывает с аналогового входа — при подключении через USB-UART и связи через COM-порт эти значения передавались на компьютер.

В блоке анализа производится оценка логической матрицы. Изначально в коде задана линия, при нажатии до которой макетный образец должен подавать напряжение на зеленый светодиод, на остальные же — на красный. Сравнив единицы из матрицы с необходимыми точками, система выдает ответный сигнал.

Ниже на рисунках 12-18 представлены схемы алгоритмов функций.

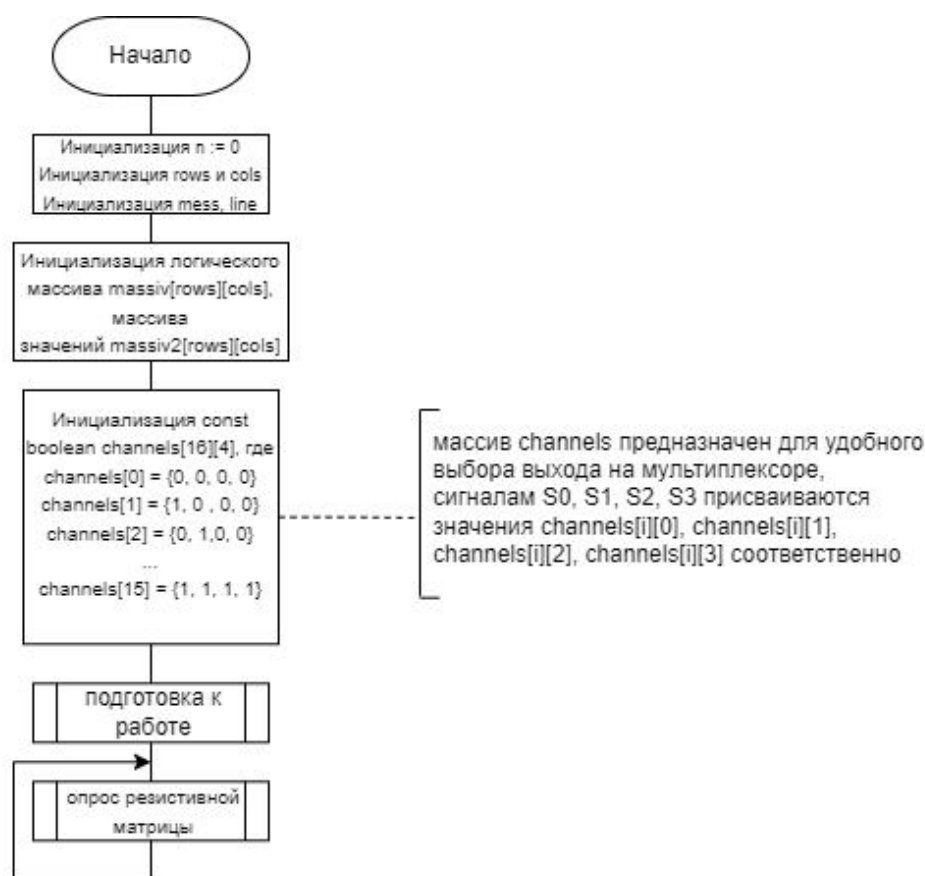


Рисунок 12 — Схема алгоритма верхнего уровня

В первую очередь необходимо задать некоторые глобальные переменные, к которым имеют доступ все функции программы. Это n — максимальное значение с аналоговых входов в отсутствие приложенного давления («шум»), неизменяемые $cols$ и $rows$ — количество вертикальных и горизонтальных линий медной ленты, массив значений с аналоговых выходов, обновляющийся при каждом проходе по матрице мата и сохраняющий значения для отладки, $massiv2$. Также сохраняется логический массив $massiv$, в котором единица означает попадание в точку на мате, ноль — отсутствие попадания. При этом попаданием считается такое значение на аналоговом входе, которое больше значения $n + mess$, где $mess$ — определенное в ходе экспериментов значение, при котором датчик давления корректно реагирует на касания. $Line$ — линия, до которой считается игровое «поле», зажигается зеленый светодиод. После нее начинается «аут», то есть включается красный светодиод.

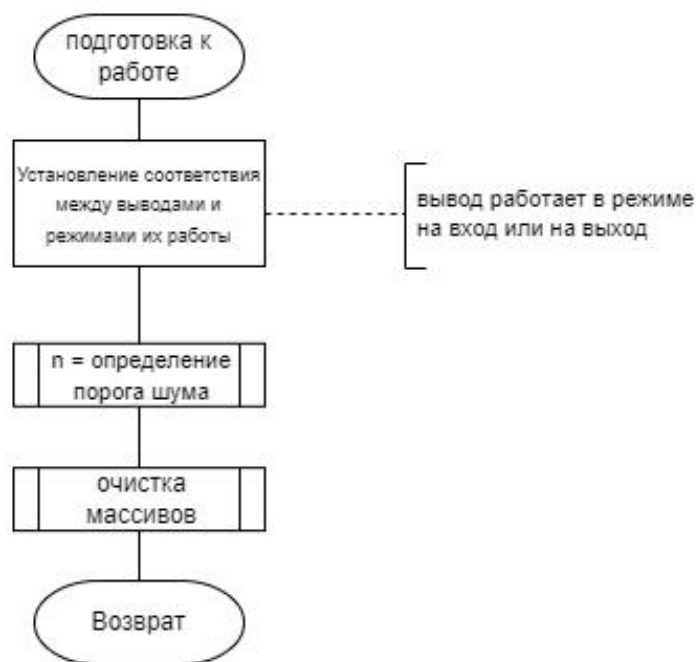


Рисунок 13 — Схема алгоритма процедуры подготовки к работе

В первую очередь необходимо установить соответствие между выводами и режимами работы — на вход или на выход, то есть сигнал либо читается, либо выводится. Далее происходит поиск «шума». Затем происходит обнуление заданных массивов.

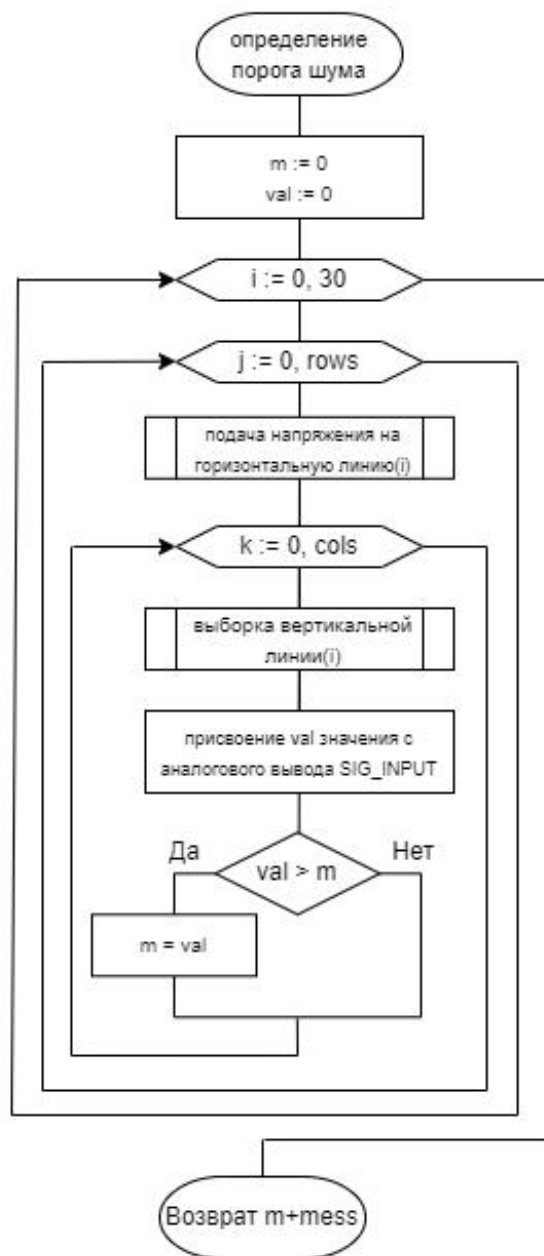


Рисунок 14 — Схема алгоритма нахождения порога «шума»

«Шум» находится следующим образом — 30 раз происходит проход по матрице мата, считываются значения напряжения. Из них выбирается максимальное. Сумма этого максимума и некоторой константы и считается «шумом».

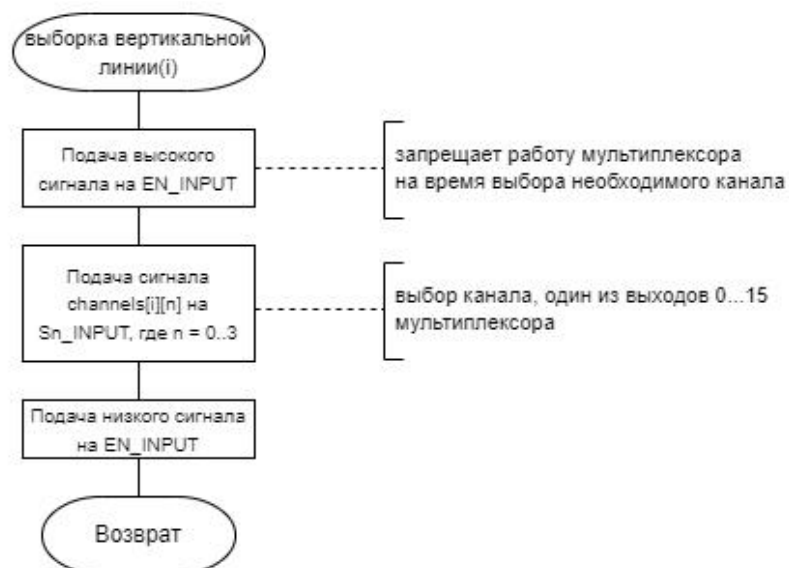


Рисунок 15 — Схема алгоритма выбора вертикальной линии

Сигнал EN на мультиплексоре или демультиплексоре запрещает его работу. Затем происходит установка нужного из C0...C15 значения на выводах S0...S3. Наконец, подается низкий сигнал на EN — мультиплексор начинает работу, сигнал с выбранного вывода подается на SIG.



Рисунок 16 — Схема алгоритма подачи напряжения на горизонтальную линию

Работа схожа со схемой алгоритма выше, но здесь уже сигнал SIG передается на один из выбранных из C0...C15 выводов.

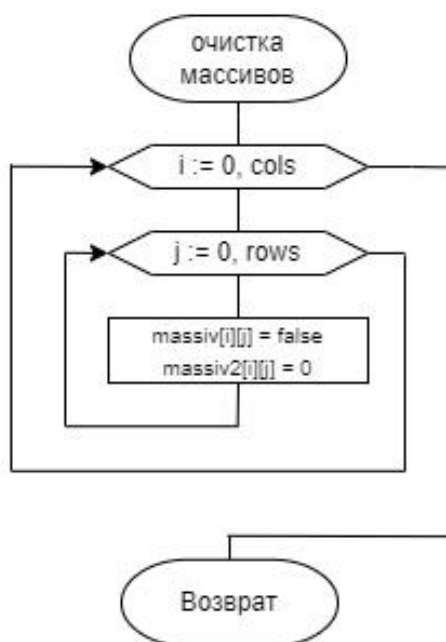


Рисунок 17 — Схема алгоритма очистки массивов

Для логического массива нулевым значением считается отсутствие попадания, значение false, для массива значений — 0.

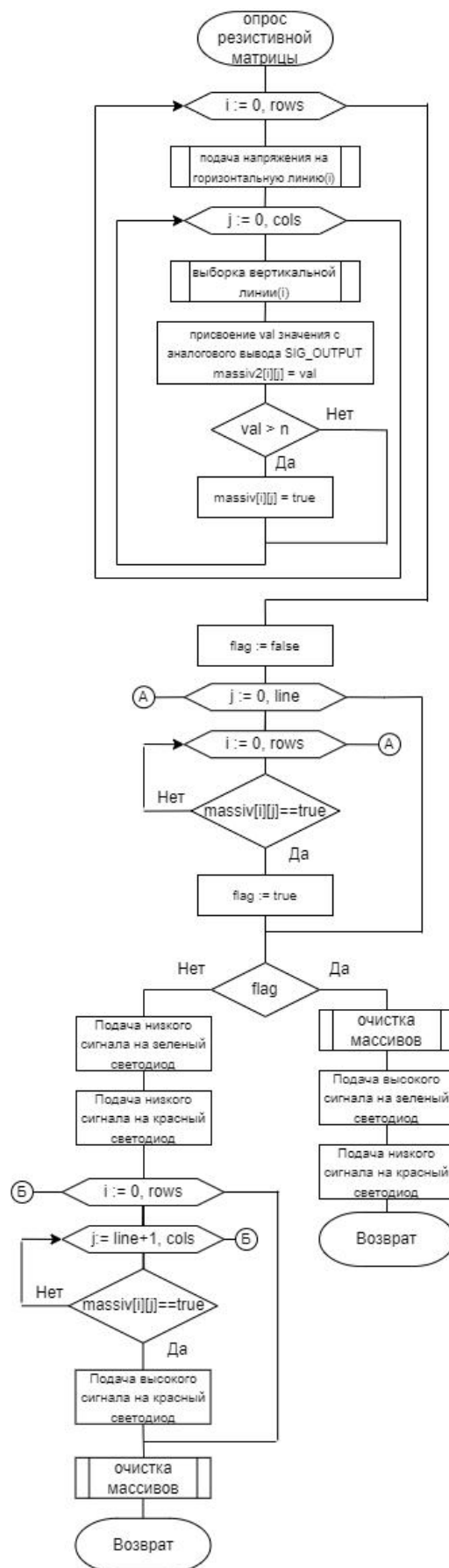


Рисунок 18 — Схема алгоритма опроса резистивной матрицы

Основной цикл выглядит следующим образом. При первом проходе по матрице значений происходит заполнение матрицы значений и логической матрицы. Затем происходит проход по части логической матрицы, до значения line. Если зафиксировано приложенное давление, происходит выход из цикла, напряжение подается на зеленый светодиод. Иначе начинается обработка второй части матрицы. Если там зафиксировано приложенное давление, начинается обработка подачи напряжения на красный светодиод.

Вне зависимости от того, был ли включен какой-либо светодиод, в конце основного цикла происходит очистка матриц.

Код программы приведен в приложении Б. При подключении устройства через переходник USB-UART к компьютеру можно отладить систему, раскомментировав строки.

4 Работа макетного образца

Схема подключения к мату на основе материала Velostat представлена на рисунке 19.

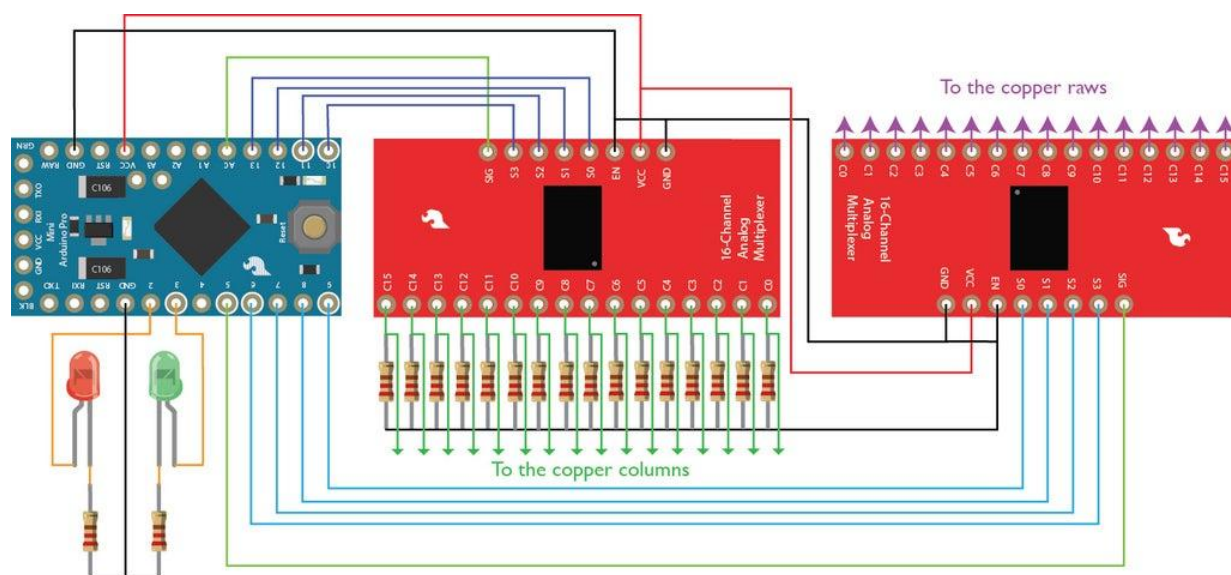


Рисунок 19 — Схема подключения

Внешний вид макетного образца представлен на рисунке 20.

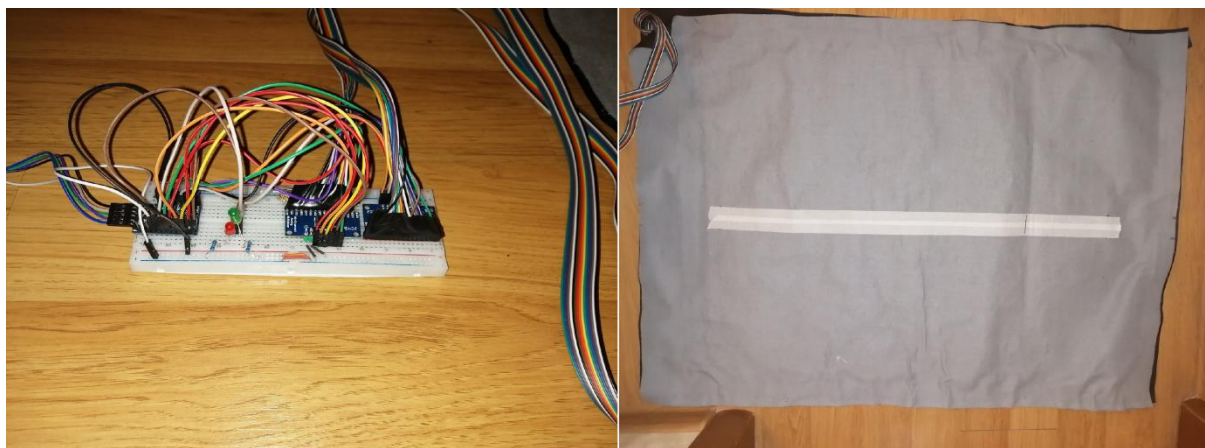


Рисунок 20 — Макетный образец

На рисунке 21 показано строение самого мата.



Рисунок 21 — Внутреннее содержимое мата — линии медной ленты и Velostat

На рисунке 22 представлен результат работы макетного образца при нажатии на участки, на которых загорается зеленый светодиод; на рисунке 23 — на которых загорается красный светодиод.

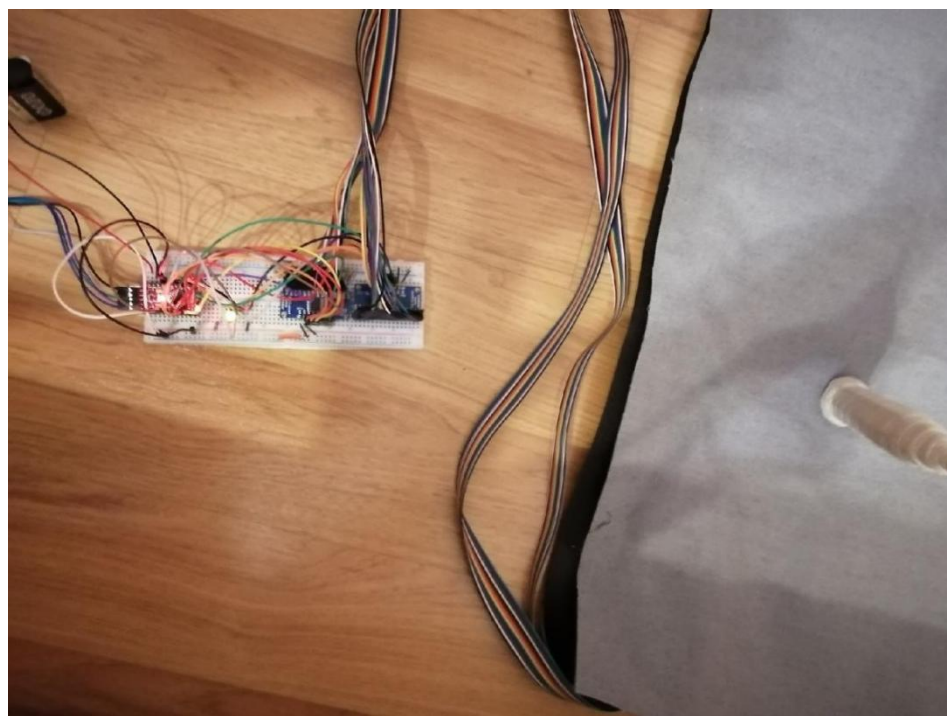


Рисунок 22 — Результат работы при нажатии и включении зеленого светодиода

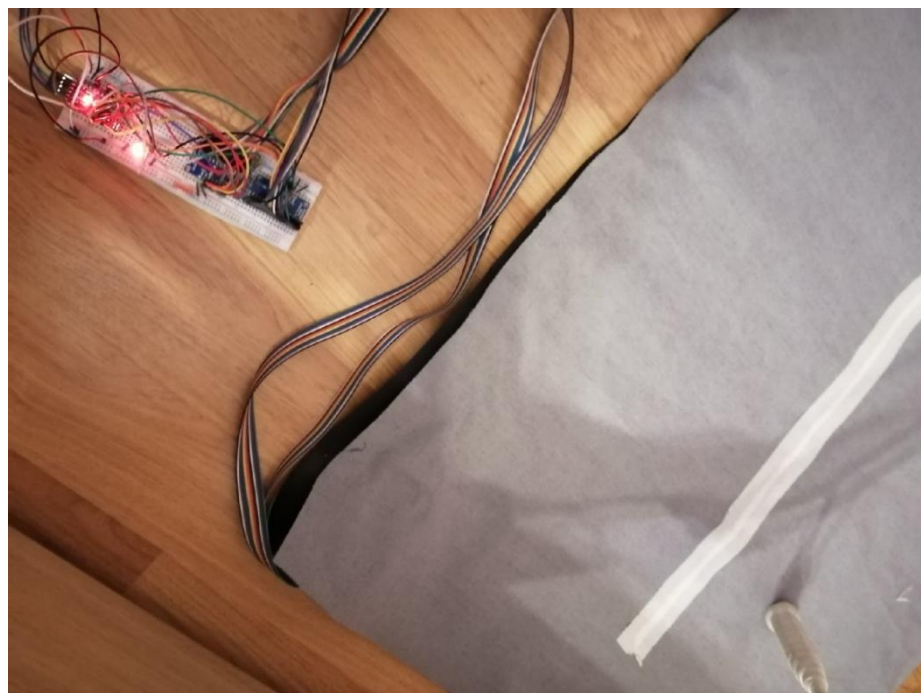


Рисунок 23 — Результат работы при нажатии и включении красного светодиода

ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы получен комплект конструкторской документации для микропроцессорной системы контроля попадания волана в заданную зону, а также собран макетный образец данной системы. Устройство отвечает заданным характеристикам.

Также в ходе работы были выполнено следующее: проанализирован объект разработки на функциональном уровне, разработана функциональная схема модуля, произведен выбор элементной базы для реализации объекта, разработана принципиальная схема модуля, рассчитаны электрические параметры.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 2.730-73. Единая система конструкторской документации. Обозначения условные графические в схемах. Приборы полупроводниковые.
2. ГОСТ 2.728-74. Единая система конструкторской документации. Обозначения условные графические в схемах. Резисторы, конденсаторы.
3. ГОСТ 2.710-81. Единая система конструкторской документации. Обозначения буквенно-цифровые в электрических схемах.
4. ГОСТ 2.755-87. Единая система конструкторской документации. Обозначения условные графические в схемах. Устройства коммутационные и контактные соединения.
5. ГОСТ 2.702-2011. Единая система конструкторской документации. Правила выполнения электрических схем.
6. ГОСТ 7.32-2017. Отчет о научно-исследовательской работе. Структура и правила оформления.
7. Петин В.А. Проекты с использованием контроллера Arduino. 2-е изд. СПб.: БХВ-Петербург, 2015. 464 с
8. Справочник по пайке / Ред. Петрунин И. Е. - 2-е изд., перераб. и дополн. - М. : Машиностроение, 1984. - 398 с.
9. Хартов, В. Я. Методические указания к выполнению курсовой работы по дисциплине «Микропроцессорные системы» : учебно-методическое пособие / В. Я. Хартов. — Москва : Издательство МГТУ им. Н. Э. Баумана, 2021. — 31, [1] с. : ил.
10. Хартов В.Я. Микроконтроллеры AVR. Практикум для начинающих. 2-е изд. М.: Из-во МГТУ им. Баумана, 2012.
11. Хартов В.Я. Микропроцессорные системы: учеб. пособие для студ. Учреждений высш. образования / В.Я.Хартов. – 2-е изд., испр. и доп. – М.: Издательский центр «Академия», 2014. – 368 с.
12. Документация на Arduino Pro Mini [Электронный ресурс]. – URL: https://www.delta-n.ru/Info/Arduino_Pro_Mini.pdf (Дата обращения 2021-14-09)

13. Портал магистров ДонНТУ. Реферат по теме выпускной работы Куксина И.Ю. [Электронный ресурс]. — URL: <https://masters.donntu.org/2013/etf/kuksin/diss/index.htm> (дата обращения 2021-09-02)
14. «ЧИП и ДИП» [Электронный ресурс]. — URL: <https://www.chipdip.ru/> (дата обращения 2021-09-02)
15. Alterozoom. Velostat [Электронный ресурс]. — URL: <https://alterozoom.com/ru/documents/43822.html> (дата обращения 2021-09-02)
16. Arduino [Электронный ресурс]. — URL: <https://www.arduino.cc/> (дата обращения 2021-09-12)
17. Arduino Master [Электронный ресурс]. — URL: <https://arduinomaster.ru> (дата обращения 2021-09-12)
18. Arduino.ru [Электронный ресурс]. — URL: <http://arduino.ru/> (дата обращения 2021-09-12)
19. Arduino.ua [Электронный ресурс]. — URL: <https://doc.arduino.ua/> (дата обращения 2021-09-16)
20. E-Portfolio for Robotics 243 Workshop Spring 2015 [Электронный ресурс]. — URL: <https://www.sites.google.com/a/mtholyoke.edu/cs-243-spring-15-barkh22g/home> (дата обращения 2021-09-02)
21. Robolive.ru. Современная робототехника [Электронный ресурс]. — URL: <https://robolive.ru/> (дата обращения 2021-10-09)

ПРИЛОЖЕНИЕ А

**Спецификация
на 1 листе**

ПРИЛОЖЕНИЕ Б

**Программный код
на 5 листах**

Листинг 1 — Программный код

```
#define SIG_OUTPUT A1
#define S3_OUTPUT 13
#define S2_OUTPUT 12
#define S1_OUTPUT 11
#define S0_OUTPUT 10
#define EN_OUTPUT 4

#define SIG_INPUT A0
#define S3_INPUT 9
#define S2_INPUT 8
#define S1_INPUT 7
#define S0_INPUT 6
#define EN_INPUT 5

#define DIODE_GREEN 3
#define DIODE_RED 2

const int cols = 15;
const int rows = 16;
const int mess = 50;
int line = 6;

int n = 0;
bool massiv[rows][cols];
int massiv2[rows][cols];

void clear_massiv() {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            massiv[i][j] = false;
            massiv2[i][j] = 0;
        }
    }
}

const boolean channels [16][4] {
    {0, 0, 0, 0}, //channel 0
    {1, 0, 0, 0}, //channel 1
    {0, 1, 0, 0}, //channel 2
    {1, 1, 0, 0}, //channel 3
```

Продолжение листинга 1

```
{0, 0, 1, 0}, //channel 4
{1, 0, 1, 0}, //channel 5
{0, 1, 1, 0}, //channel 6
{1, 1, 1, 0}, //channel 7
{0, 0, 0, 1}, //channel 8
{1, 0, 0, 1}, //channel 9
{0, 1, 0, 1}, //channel 10
{1, 1, 0, 1}, //channel 11
{0, 0, 1, 1}, //channel 12
{1, 0, 1, 1}, //channel 13
{0, 1, 1, 1}, //channel 14
{1, 1, 1, 1} //channel 15
};

void setup() {
    Serial.begin(9600);
    pinMode(S0_OUTPUT, OUTPUT);
    pinMode(S1_OUTPUT, OUTPUT);
    pinMode(S2_OUTPUT, OUTPUT);
    pinMode(S3_OUTPUT, OUTPUT);

    pinMode(S0_INPUT, OUTPUT);
    pinMode(S1_INPUT, OUTPUT);
    pinMode(S2_INPUT, OUTPUT);
    pinMode(S3_INPUT, OUTPUT);

    pinMode(EN_OUTPUT, OUTPUT);
    pinMode(EN_INPUT, OUTPUT);

    pinMode(SIG_OUTPUT, OUTPUT);
    pinMode(SIG_INPUT, INPUT);

    pinMode(DIODE_RED, OUTPUT);
    pinMode(DIODE_GREEN, OUTPUT);

    n = noise();
    clear_massiv();
}
```

Продолжение листинга 1

```
void loop() {
    //необходимо пройти по всем портам S0..S3 выхода и подать на C0..C15 платы
питания высокий сигнал
    for (int i = 0; i < rows; i++) {
        connect_line_rows(i);
        for (int j = 0; j < cols; j++) {
            connect_line_cols(j);
            int val = analogRead(SIG_INPUT);
            massiv2[i][j] = val;
            if (val > n) {
                massiv[i][j] = true;
            }
        }
    }
    bool flag = false;
    for (int j = 0; j <= line; j++) {
        for (int i = 0; i < rows; i++) {
            if (massiv[i][j] == true) {
                //Serial.print("in ");
                //Serial.print(i);
                //Serial.print(" ");
                //Serial.print(j);
                //Serial.print(" ");
                //Serial.println(massiv2[i][j]);
                flag = true;
                break;
            }
        }
    }

    if (flag) {
        clear_massiv();
        digitalWrite(DIODE_RED, LOW);
        digitalWrite(DIODE_GREEN, HIGH);
        return;
    } else {
        digitalWrite(DIODE_GREEN, LOW);
        digitalWrite(DIODE_RED, LOW);
    }

    for (int i = 0; i < rows; i++) {
```

Продолжение листинга 1

```
for (int j = line+1; j < cols; j++) {
    if (massiv[i][j] == true) {
        digitalWrite(DIODE_RED, HIGH);
        //Serial.print("out ");
        //Serial.print(i);
        //Serial.print(" ");
        //Serial.print(j);
        //Serial.print(" ");
        //Serial.println(massiv2[i][j]);

    }
}
clear_massiv();
}

int noise() {
    int m = 0;
    int val = 0;
    //Serial.println("Calibrate...");
    for (int i = 0; i < 30; i++) {
        for (int j = 0; j < rows; j++) {
            connect_line_rows(j);
            for (int k = 0; k < cols; k++) {
                connect_line_cols(k);
                val = analogRead(SIG_INPUT);
                if (val > m) {
                    m = val;
                }
            }
        }
    }
    //Serial.println(m+mess);
    //Serial.println("Done");
    return m + mess;
}

void connect_line_rows(int i) {
    //запрещаем работу
```


Продолжение листинга 1

```
digitalWrite(EN_OUTPUT, HIGH);

digitalWrite(S0_OUTPUT, channels[i][0]);
digitalWrite(S1_OUTPUT, channels[i][1]);
digitalWrite(S2_OUTPUT, channels[i][2]);
digitalWrite(S3_OUTPUT, channels[i][3]);

digitalWrite(EN_OUTPUT, LOW);
digitalWrite(SIG_OUTPUT, HIGH);
}

void connect_line_cols(int i) {
    //запрещаем работы
    digitalWrite(EN_INPUT, HIGH);

    digitalWrite(S0_INPUT, channels[i][0]);
    digitalWrite(S1_INPUT, channels[i][1]);
    digitalWrite(S2_INPUT, channels[i][2]);
    digitalWrite(S3_INPUT, channels[i][3]);
    digitalWrite(EN_INPUT, LOW);
}
```