

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский Томский политехнический университет»

Школа / филиал	ИШНКБ
Обеспечивающее подразделение	Отделение электронной инженерии
Направление подготовки / специальность	12.03.04 Биотехнические системы и технологии
Образовательная программа (направленность (профиль))	Биотехнические системы технологии

ОТЧЕТ
ПО УЧЕБНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

Тема	Разработка автоматического усилителя модуля регистрации ЭМГ
------	---

Выполнил обучающийся	Нгуен.Н.Ф
Группа	1Д01

(Подпись обучающегося)

Дата сдачи работы «__» _____ 2024 г.

Проверил науч. руководитель _____
(Степень, звание, должность) (Ф.И.О.)

Дата проверки _____ 2024 г.

Баллы (из 40) _____

Допустить/не допустить к защите,

Подпись _____

Отчет принял¹ _____
(Степень, звание, должность) (Ф.И.О.)

Дата защиты _____ 2024г.

Баллы (из 60) _____

Подпись _____

Итоговое решение: зачет/незачет, итоговые баллы (из 100) _____

Томск 2024

¹ Председатель комиссии по защите УИРС – руководитель ООП

УТВЕРЖДАЮ

Руководитель ООП

_____ Е.Ю. Дикман

«___» _____ 2024 г.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

На учебно-исследовательскую работу

Тема учебно-исследовательской работы:

Разработка автоматического усилителя модуля регистрации ЭМГ

Перечень работ (заданий), подлежащих выполнению:

1. Разработка структурной схемы устройства
2. Разработка принципиальной схемы платы модуля регистрации ЭМГ
3. Разработка программы для микроконтроллера
4. Экспериментальные результаты

Научный руководитель

(Должность)

(подпись)

(Ф. И. О.)

Задание принял к исполнению _____

(Подпись)

(Ф. И. О. обучающегося)

«___» _____ 2024

Оглавление

Введение.....	4
Разработка структурной схемы устройства.....	5
Разработка принципиальной схемы плата модуля регистрации ЭМГ	7
Инструментальный усилитель AD8221	7
Системный фильтр	7
Преобразователи напряжения	8
Аналого-цифровое преобразование	9
Контактные выходы	9
Разработка моделей печатных плат модуля регистрации ЭМГ	10
Разработка программы для микроконтроллер	11
Разработка библиотеки кода и I2C для MCP3421.....	11
Настройка для HC-05	15
Функция автоматического усиления	16
Экспериментальные результаты	17
Список литературы	19
Приложение А	20
Приложение Б	21
Приложение В.....	22
Приложение Г	23
Приложение Д	24
Приложение Е.....	25

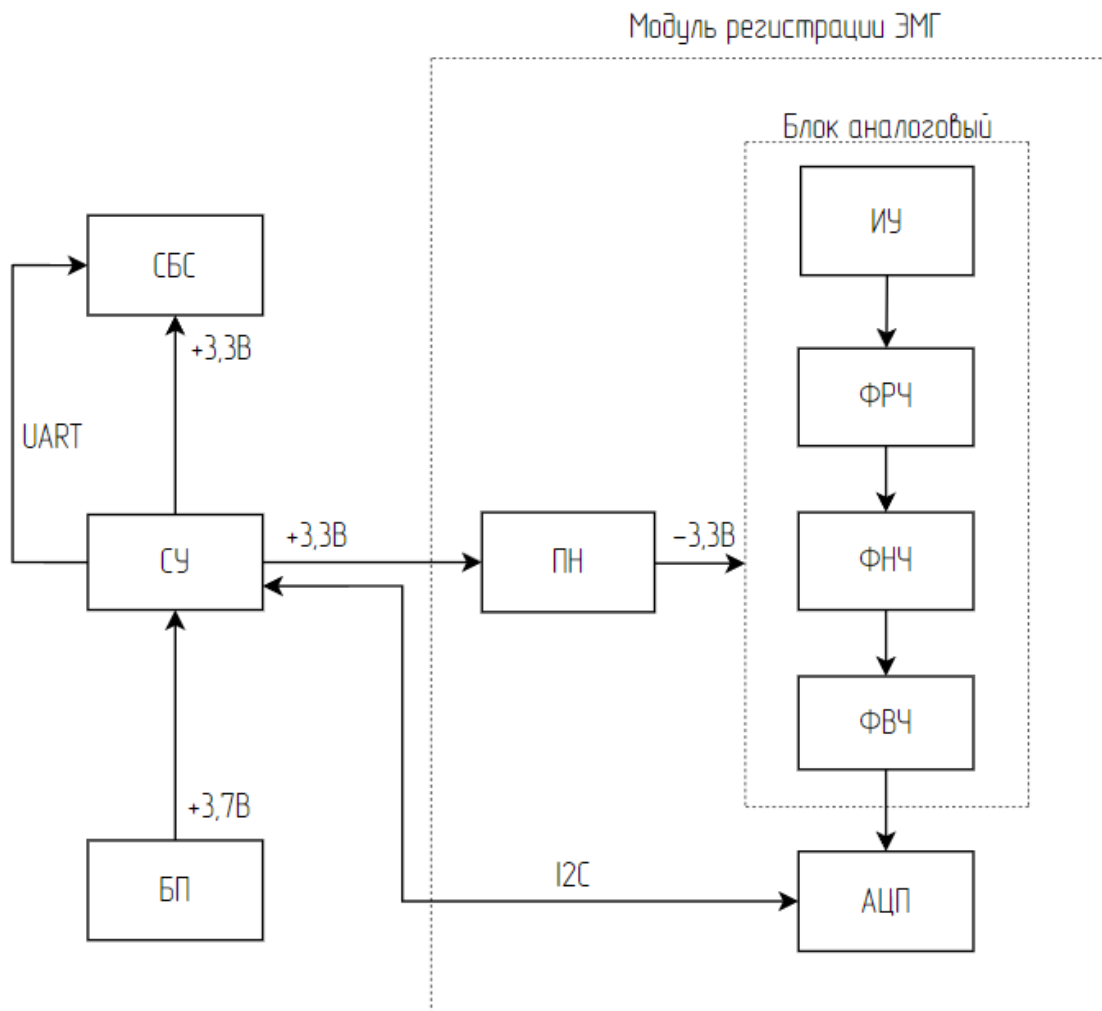
Введение

Разрабатываемое устройство главным образом направлено на использование в сборе данных ЭМГ верхней конечности с целью использования его для управления протезом руки.

Верхняя конечность включает в себя различные группы мышц, каждая из которых проявляет различный диапазон амплитуды в сигналах электромиографии (ЭМГ). Вместо использования ручной калибровки усиления усилителя или фиксированного уровня усиления автоматическая система регулировки усиления предоставляет более точное и практичное решение для захвата различных диапазонов ЭМГ, характерных для различных групп мышц. Этот автоматизированный подход не только повышает точность мониторинга активности мышц, но также обладает значительной универсальностью в многочисленных областях применения, включая управление протезами. Возможность динамически адаптировать уровень усиления сенсора для соответствия конкретным характеристикам ЭМГ различных мышечных групп не только обеспечивает более точный сбор данных, но также открывает перспективы для передовых приложений в области протезирования, где точное и отзывчивое управление играет ключевую роль в улучшении функциональности и интеграции протезированных конечностей.

Разработка структурной схемы устройства

На рисунке 1 представлена структурная схема устройства автоматического усилителя модуля регистрации электромиографии (ЭМГ).



БП – блок питания

СБС – система беспроводной связи

СУ – система управления

АЦП – аналого-цифровое преобразование

ПН – преобразователи напряжения

ИУ – инструментальный усилитель

ФРЧ – фильтр режекторного частот

ФНЧ – фильтр нижних частот

ФВЧ – фильтр верхних частот

Рисунок 1 – Структурная схема

Блок питания (БП) обеспечивает требуемым напряжением каждый элемент схемы. В качестве источника будут выступать Li-ion аккумулятор. Для стабилизации и получения необходимого напряжения используется преобразователя постоянного напряжения на 3.3 В.

Система управления (СУ) – обеспечивает взаимодействие и управление всех блоков, устройства. Для реализации данной системы будет использоваться МК STM8S 103F3P6. Поскольку он оснащен необходимыми интерфейсами: UART, I2C; достаточным количеством памяти и частоты. Задача СУ будет заключаться в приеме и обработке данных от модуля регистрации ЭМГ, последующей отправки этих данных через СБС.

Система беспроводной связи (СБС) – беспроводная связь будет обеспечивать контакт устройства с принимающей стороной (протезом, ноутбуком...) для передачи данных. Будет использоваться модуль Bluetooth HC-05, который прост в использовании, предназначенный для прозрачной настройки беспроводного соединения. Из СУ будут передаваться собранные данные от АЦП в СБС и отправляться принимающей стороне.

Регистрация модуля ЭМГ включает: ПН - Преобразователи напряжение будет подавать отрицательное напряжение на усилители. ИН - Инструментальный усилитель будет принимать дифференциальный сигнал от электрода и эффективно подавлять общий сигнал. Система фильтров включает в себя ФРЧ - фильтр режекторного частот, ФНЧ - фильтр нижних частот, ФВЧ - фильтр верхних частот. Эти фильтры хорошо удаляют почти весь шум и помехи. Кстати, через все фильтры сигнал будет усилен настолько, что его можно будет прочесть. АЦП - аналого-цифровое преобразование используется для преобразования аналогового сигнала в цифровой сигнал, и с помощью интерфейса I2C МК может считывать этот цифровой сигнал.

Разработка принципиальной схемы плата модуля регистрации ЭМГ

Инструментальный усилитель AD8221

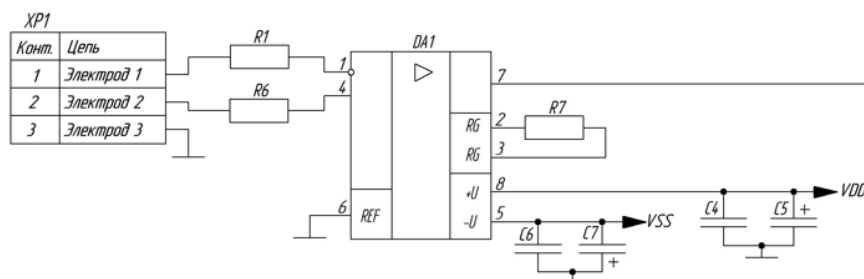


Рисунок 2 – Схема инструментального усилителя

Данный инструментальный усилитель питается он напряжения ± 3.3 В, для сглаживания более высокочастотных помех применяются конденсаторы более низкой емкости C4, C5, C6, C7. Конденсатор керамический емкостью 0,1 мкФ (C4, C6) следует разместить рядом с каждым контактом питания, танталовый конденсатор емкостью 10 мкФ (C5, C7) можно использовать дальше от детали[1].

Резистор R1, R6 используется для ограничения тока при наличии напряжения ESD. Следовательно, все компоненты будут защищены от высотких тока.

Системый фильтр

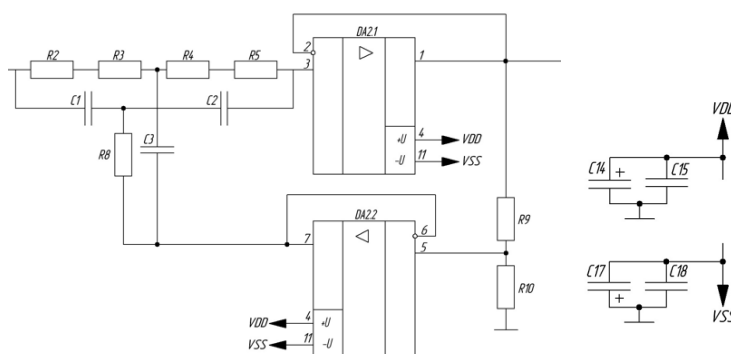


Рисунок 3 – Фильтр режекторного частот

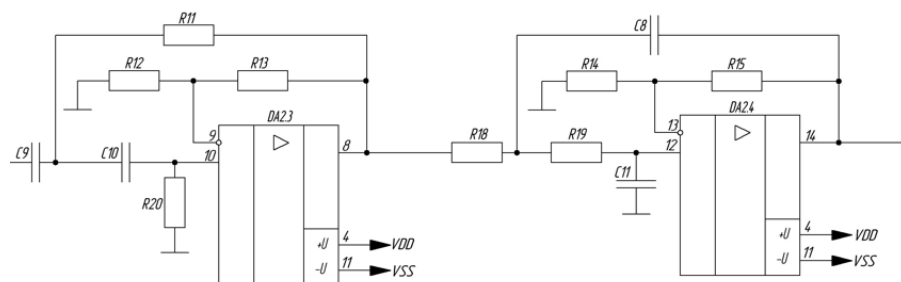


Рисунок 4 – Фильтр нижник и верхн частот

Данные операционные усилители питаются он напряжения ± 3.3 В, для сглаживания более высокочастотных помех применяются конденсаторы более низкой емкости C14, C15, C17, C18. Конденсатор керамический емкостью 0,1 мкФ (C15, C18) следует разместить рядом с каждым контактом питания, танталовый конденсатор емкостью 2,2 мкФ (C14, C77) можно использовать дальше от детали.

Преобразователи напряжения

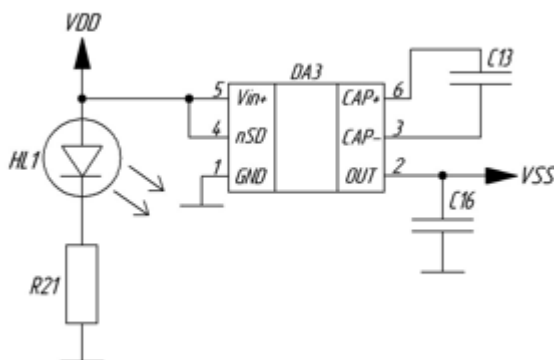


Рисунок 5 – Схема преобразователя напряжения

Чтобы максимизировать эффективность, уменьшить падение выходного напряжения и пульсации напряжения[2], был использованы два конденсатора Murata 10 мкФ с низким ESR.

Для индикации статуса получаемых напряжения используется светодиод HL1 с ток ограничительным резистором (R21).

Аналого-цифровое преобразование

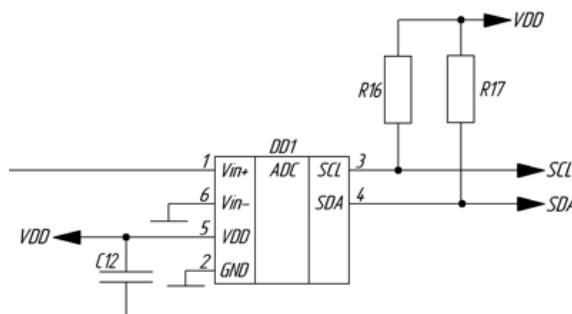


Рисунок 6 – Схема АЦП

Данный АЦП питается от напряжения 3.3 В, для сглаживания более высокочастотных помех применяются керамический конденсатор более низкой емкости C12. Конденсатор следует размещать как можно ближе к выводу VDD (в пределах одного дюйма). Если схема приложения имеет отдельные цифровые и аналоговые источники питания, VDD и VSS устройства MCP3421 должны находиться в аналоговой плоскости [3].

Выходы SCL и SDA MCP3421 имеют конфигурации с открытым стоком. Для этих контактов требуется подтягивающий резистор, как показано на рисунке 6. Подтягивающий резистор обычно выбирается в диапазоне от 5 кОм до 10 кОм для стандартного и быстрого режимов [3].

Контактные выходы

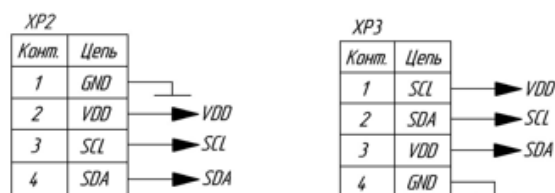


Рисунок 7 – Схема выходы

Выход предназначен для подключения к MCU через шину I2C, а также доступен в случае использования нескольких модулей.

Разработка моделей печатных плат модуля регистрации ЭМГ

На рисунках 8-9 приводятся результаты разработки платы модуля регистрации ЭМГ. Плата обеспечена монтажными отверстиями для крепления её внутри корпуса. На плате имеются два отверстия диаметром 4 мм для пайки электродами. Расположение элементов было спроектировано с целью минимизировать шум и помехи. Для достижения максимальной эффективности, входной тракт инструментального усилителя выполнен синхронным. Высокочастотный шум может исходить от цифровой части. Поэтому аналоговая часть отделена от цифровой. Линия вывода сигнала изолирована небольшими отверстиями, что гарантирует отсутствие помех.

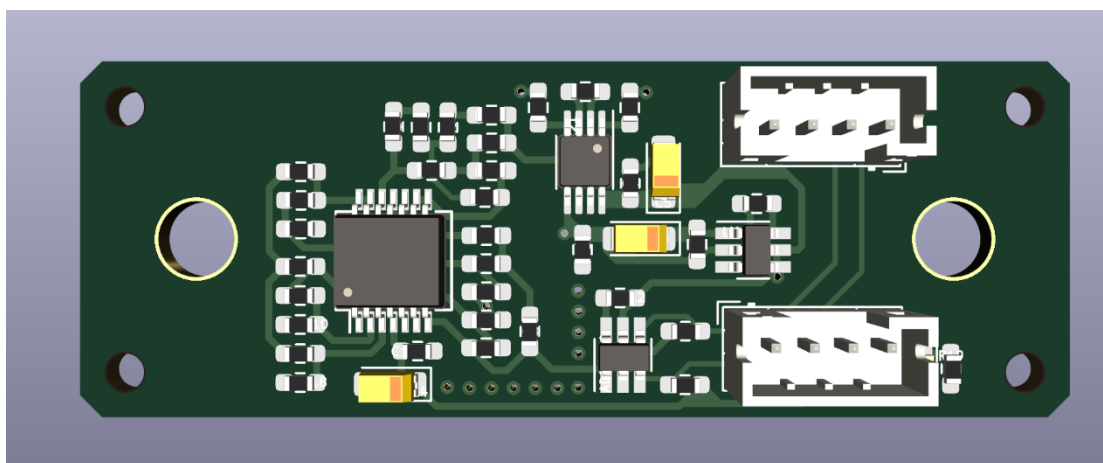


Рисунок 8 – Вид на верхнюю сторону платы

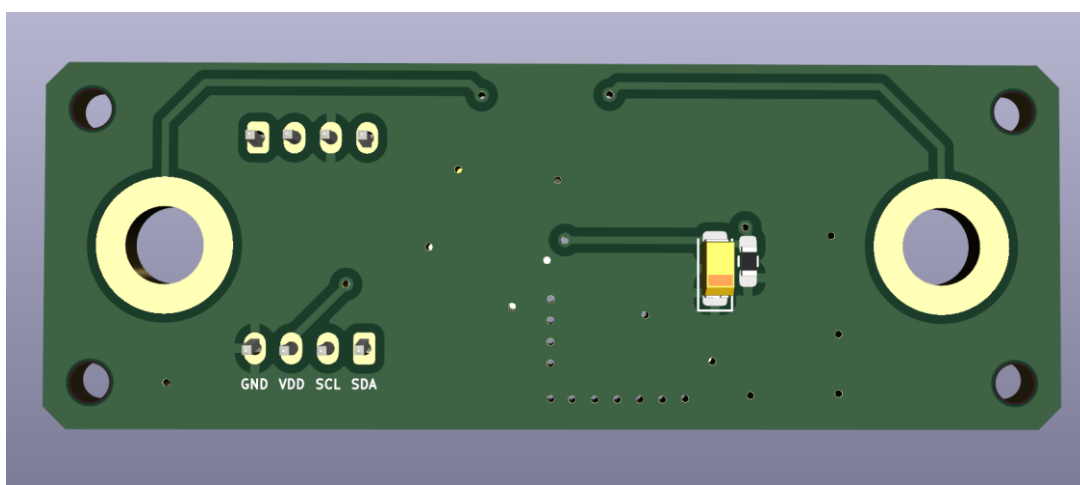


Рисунок 9 – Вид на нижнюю сторону платы

Разработка программы для микроконтроллер

Разработка библиотеки кода и I2C для MCP3421.

Конфигурация АЦП MCP3421 представлена на рисунке 10. Библиотечный код для MCP3421 был написан на основе его конфигурации.

R/W-1	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
$\overline{\text{RDY}}$	C1	C0	$\overline{\text{O/C}}$	S1	S0	G1	G0
1 *	0 *	0 *	1 *	0 *	0 *	0 *	0 *
bit 7			bit 0				
* Default Configuration after Power-On Reset							
Legend:							
R = Readable bit		W = Writable bit		U = Unimplemented bit, read as '0'			
-n = Value at POR		'1' = Bit is set		'0' = Bit is cleared		x = Bit is unknown	

bit 7	$\overline{\text{RDY}}$: Ready Bit This bit is the data ready flag. In read mode, this bit indicates if the output register has been updated with a latest conversion result. In One-Shot Conversion mode, writing this bit to "1" initiates a new conversion. Reading $\overline{\text{RDY}}$ bit with the read command: 1 = Output register has not been updated. 0 = Output register has been updated with the latest conversion result. Writing $\overline{\text{RDY}}$ bit with the write command: Continuous Conversion mode: No effect One-Shot Conversion mode: 1 = Initiate a new conversion. 0 = No effect.
bit 6-5	C1-C0: These bits are not effected for the MCP3421.
bit 4	$\overline{\text{O/C}}$: Conversion Mode Bit 1 = Continuous Conversion Mode (Default). The device performs data conversions continuously. 0 = One-Shot Conversion Mode. The device performs a single conversion and enters a low power standby mode until it receives another write or read command.
bit 3-2	S1-S0: Sample Rate Selection Bit 00 = 240 SPS (12 bits) (Default) 01 = 60 SPS (14 bits) 10 = 15 SPS (16 bits) 11 = 3.75 SPS (18 bits)
bit 1-0	G1-G0: PGA Gain Selection Bits 00 = x1 (Default) 01 = x2 10 = x4 11 = x8

Рисунок 10 – Конфигурационный байт MPC3421

```

#ifndef MCP3421_H
#define MCP3421_H

#define MCP3421      0b11010000 /**MCP3421 device address*/
#define Max_12bit    2047
#define Max_14bit    8191
#define Max_16bit    32767
#define Max_18bit    131071

typedef enum { CONTI_CONVERSION_MODE    = (uint8_t)(1<<4),
               SINGLE_CONVERSION_MODE  = (uint8_t)(0<<4)
               } MCP3421_MODE_TypeDef;

typedef enum { SPS240_12bit    = (uint8_t)(0b00&(3<<2)),
               SPS60_14bit     = (uint8_t)(0b01&(3<<2)),
               SPS15_16bit     = (uint8_t)(0b10&(3<<2)),
               SPS4_18bit      = (uint8_t)(0b11&(3<<2))
               } MCP3421_SPS_TypeDef;

typedef enum { GAIN1          = (uint8_t)(0b00&(3<<0)),
               GAIN2          = (uint8_t)(0b01&(3<<0)),
               GAIN4          = (uint8_t)(0b10&(3<<0)),
               GAIN8          = (uint8_t)(0b11&(3<<0))
               } MCP3421_GAIN_TypeDef;

#endif

```

Рисунок 11 – Код конфигурационного байта МРС3421

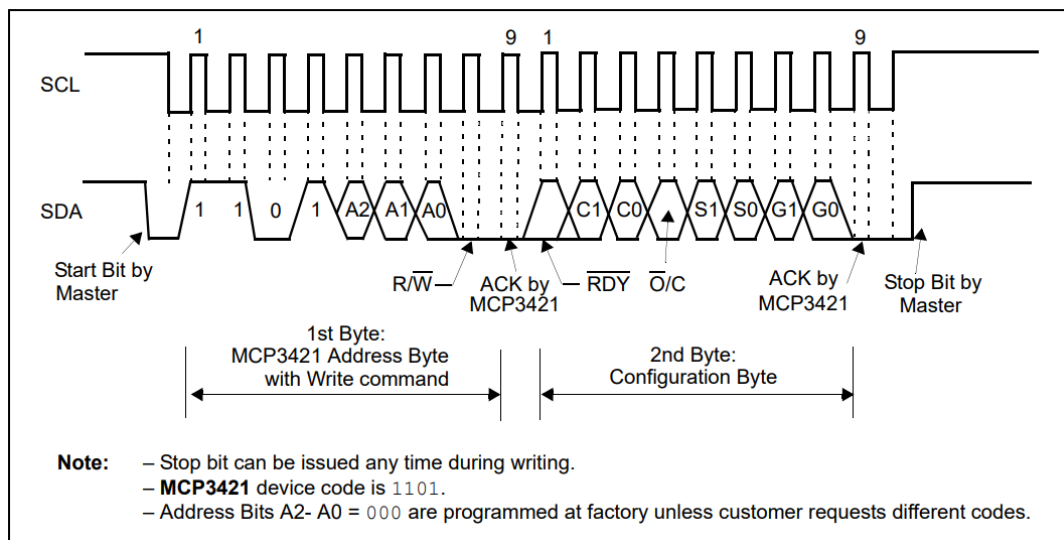


Рисунок 12 – Временная диаграмма записи в МРС3421

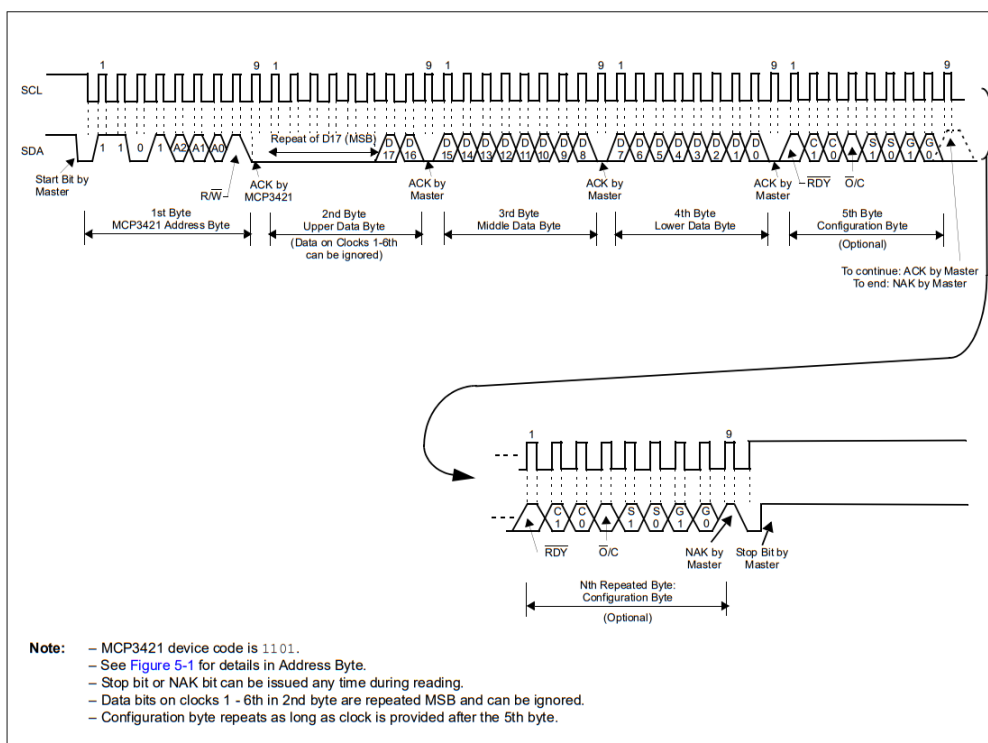


Рисунок 12 – Временная диаграмма чтения с MCP3421

Настройка для интерфейса I2C - это первый шаг, который должен быть выполнен. Затем I2C запускается через бит START. Обратите внимание, что также должен быть установлен бит ACK. Прерывание будет выполнено немедленно, когда I2C запускается и устанавливается флаг SB. Внутри обработчика прерываний первым делом отправляется адрес ведомого устройства, чтобы проверить, совпадает ли он или нет. Если адрес совпадает, процесс прерывания повторяется, поскольку установлен флаг ADDR. Затем проверяется флаг TRA, чтобы определить, будет ли I2C в режиме чтения или записи. В каждом режиме соответственно считывается или записывается регистр DR. В конце процесса чтения или записи должен быть установлен бит STOP, чтобы освободить шину перед переходом к новому общению.

На основе функционального описания I2C в таблице данных STM8S[5] и временной диаграммы чтения/записи MCP3421 (рисунок 11,12) было написано краткое описание, которое также используется для алгоритма (рисунок 13).

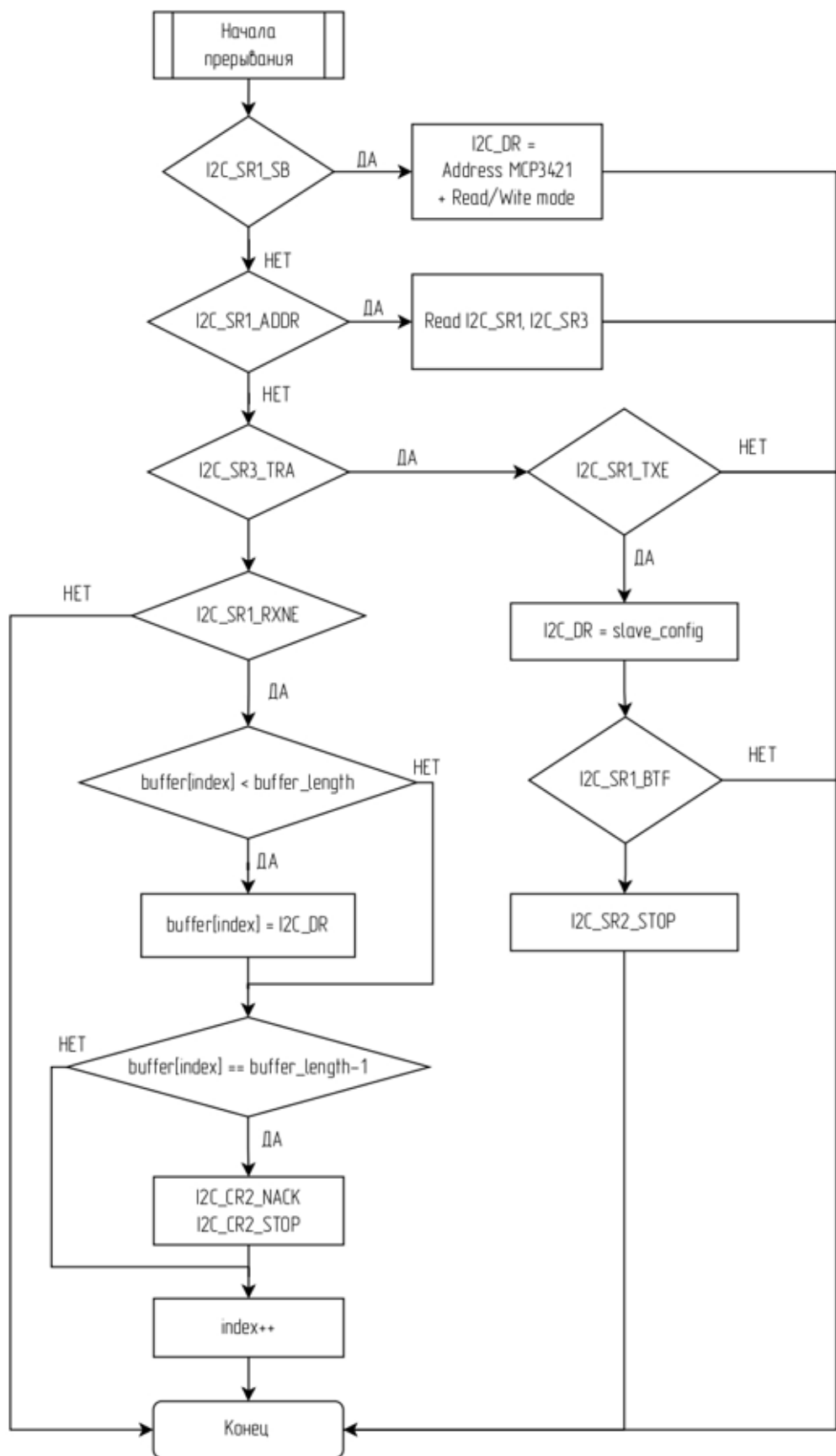


Рисунок 13 – Алгоритм прерывания I2C STM8S

Настройка для HC-05

Для взаимодействия с HC05-модулем используются AT-команды. Интерфейсом для взаимодействия служит UART. Первоначальные настройки HC05 модуля заключались в следующем [4]:

- Тестовая команда: AT
- Проверить адрес AT+ADDR
- Настроить UART(baud rate, stop bit, parity): AT+UART=9600,1,0
- Установить/проверить режим модуля (master): AT+ROLE=1
- Установить/проверить режим подключения (фиксированный адрес): AT+CMODE=0
- Подключиться к подчиненному устройству: AT+BIND=xxxx,xx,xxxxxx

Чтобы включить режим AT для модуля HC-05, подключите вывод P1011 к высокому уровню, используя скорость передачи данных 34800. Пример показан на рисунке xx ниже.

```
03:22:15.658 -> OK
03:22:33.927 -> +NAME:HC05Slave
03:22:33.927 -> OK
03:22:43.485 -> +UART:9600,1,0
03:22:43.485 -> OK
03:22:59.762 -> +ROLE:0
03:22:59.809 -> OK
03:23:27.956 -> +CMODE:1
03:23:27.956 -> OK
03:23:34.510 -> +BIND:0:0:0
03:23:34.543 -> OK
03:23:44.678 -> +ADDR:98D3:51:F66A3C
03:23:44.678 -> OK
```

Рисунок 14 – Ответ от HC05 при использовании AT-команды

Функция автоматического усиления

Как только значение будет получено от АЦП, будет запущен процесс проверки, чтобы убедиться, что значение все еще находится в пределах диапазона. Если выход за пределы диапазона достигнут, коэффициент усиления АЦП снижается. Этот процесс повторяется до тех пор, пока не будет достигнут минимальный коэффициент усиления. Алгоритм всей программы показан на рисунке 15 ниже. Код программы указан в приложении Е.

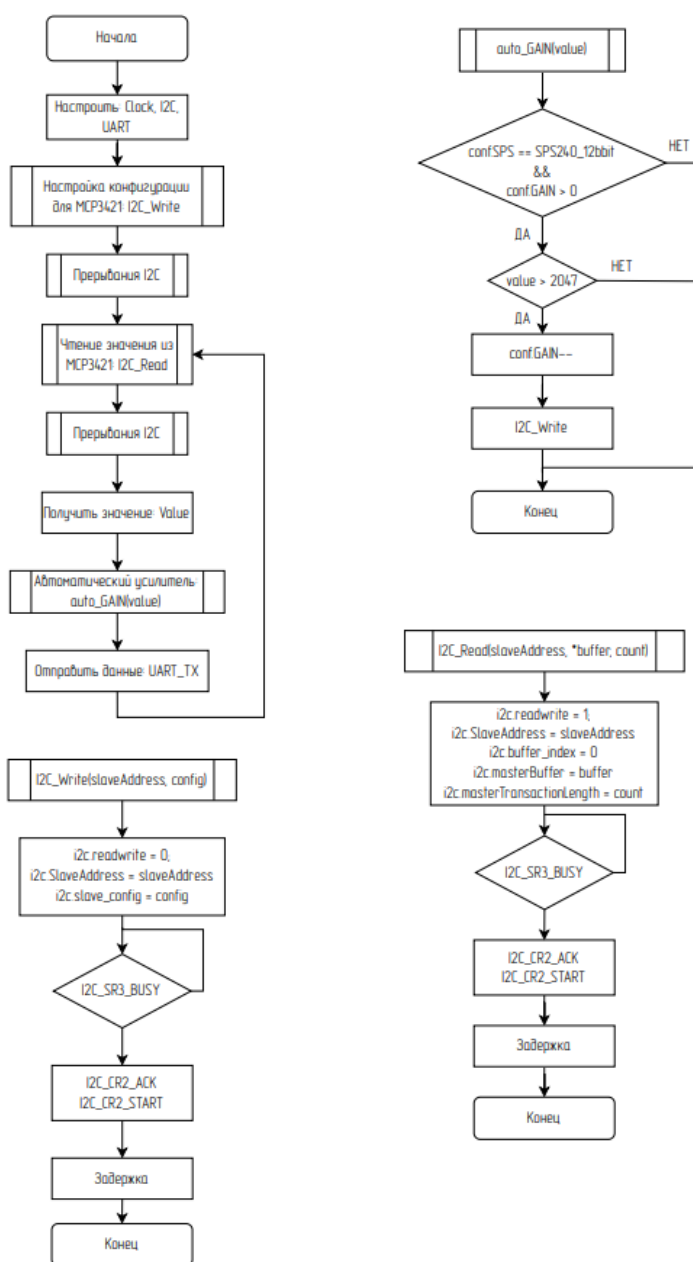


Рисунок 15 – Алгоритм всей программы

Экспериментальные результаты

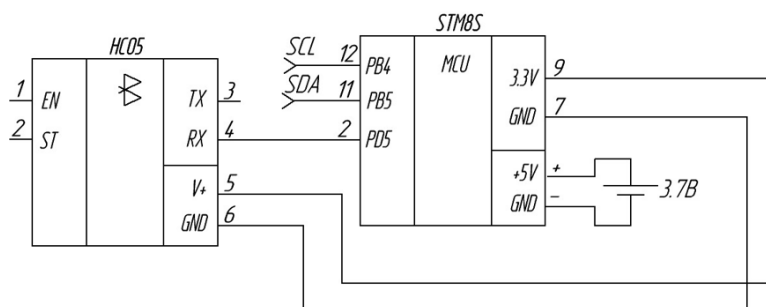


Рисунок 10 – Схема подключения МК STM8S и HC-05

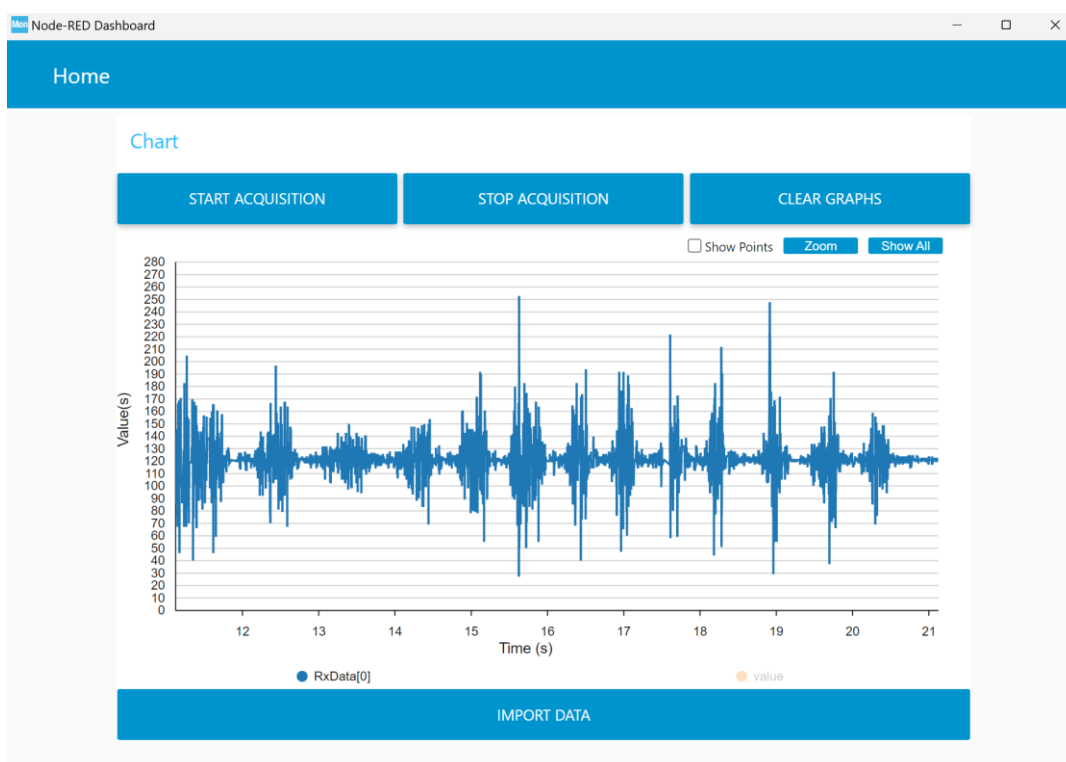


Рисунок 11 – Результат вывода сигнала, полученного от HC-05

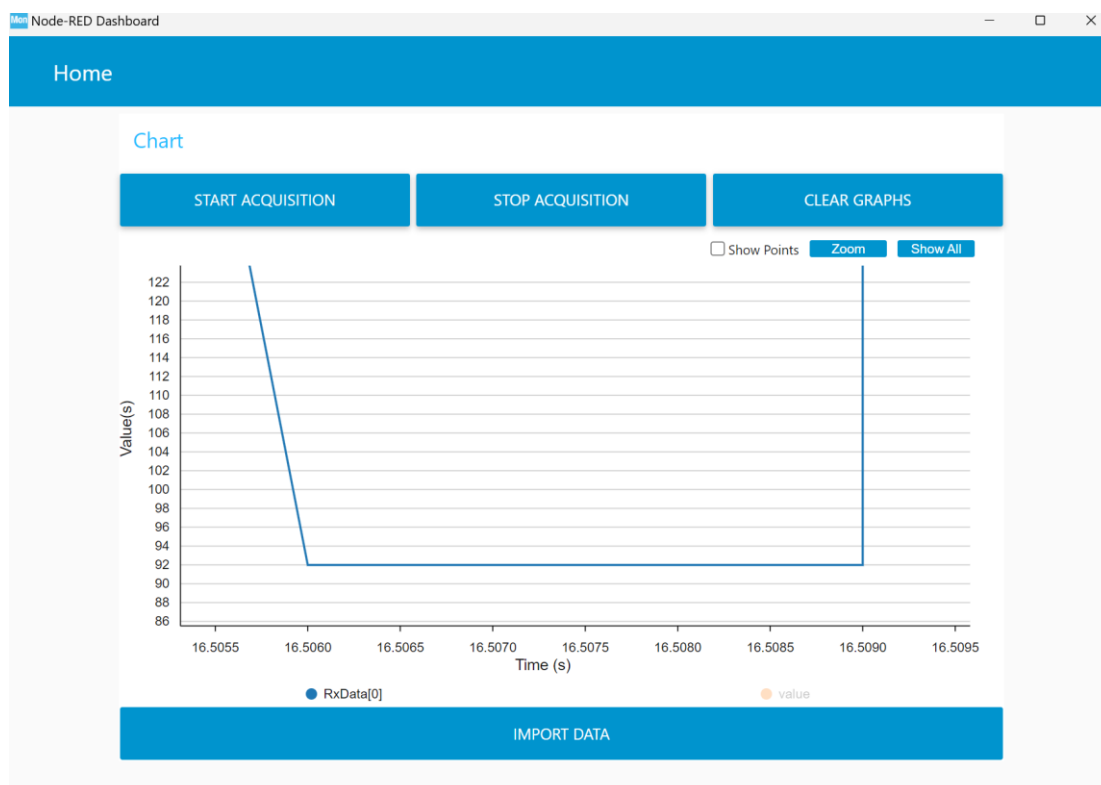


Рисунок 12 – Уменьшите масштаб сигнала. Очевидно, что частота дискретизации сигнала составляет 3 мс.

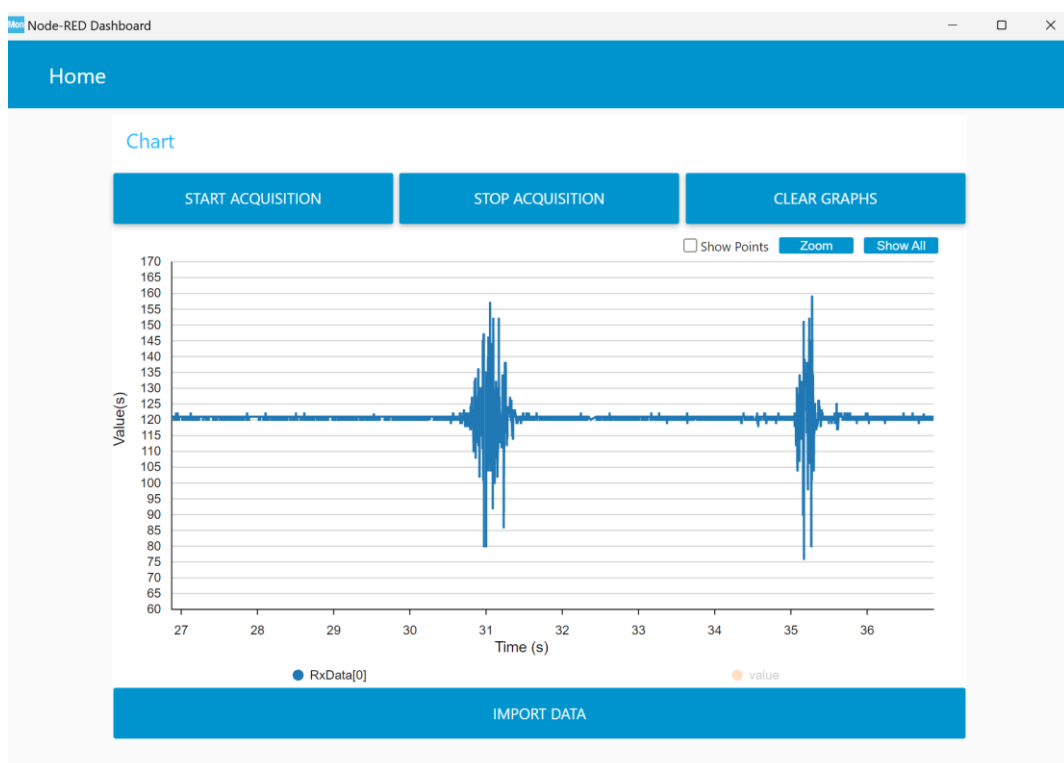


Рисунок 13 – Жест сигнала покоя и жест удержания. Очевидно, что сигнал чистый и нет никаких шумов и помех.

Список литературы

1. Datasheet – AD8221, Инструментальный усилитель, [Электронный ресурс] – Режим доступа: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad8221.pdf> (Дата обращения: 07.04.2024)
2. Datasheet – LM2664, Преобразователь напряжения с переключаемым конденсатором, [Электронный ресурс] – Режим доступа: <https://www.ti.com/lit/ds/symlink/lm2664.pdf?ts=1713246289404> (Дата обращения: 07.04.2024)
3. Datasheet – MCP3421, Цифровой потенциометр, [Электронный ресурс] – Режим доступа: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad5259.pdf> (Дата обращения: 07.04.2024)
4. Datasheet – HC-05, Модуль Bluetooth для последовательного порта, [Электронный ресурс] – Режим доступа: https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf (Дата обращения: 07.04.2024)
5. Reference manual – STM8S Series and STM8AF Series 8-bit microcontroller, [Электронный ресурс] – Режим доступа: https://www.st.com/content/st_com/en.html (Дата обращения: 07.04.2024)

Приложение А

Модуль регистрации ЭМГ

Спецификация

ФЮРА.942519.001

Приложение Б

Модуль регистрации ЭМГ

Схема электрическая принципиальная

ФЮРА.942519.001 ПЭЗ

Приложение В

Модуль регистрации ЭМГ

Сборочный чертеж платы

ФЮРА.942519.002.СБ

Приложение Г

Модуль регистрации ЭМГ

Печатная плата

ФЮРА.942519.003

Приложение Д
Модуль регистрации ЭМГ
Перечень элементов
ФЮРА.942519.004 ПЭ

Приложение Е

Модуль регистрации ЭМГ

Код микроконтроллера

Код main.c

```
1  /* MAIN.C file
2  *
3  * Copyright (c) 2002-2005 STMicroelectronics
4  */
5
6  #include<stm8s.h>
7  #include<stm8s103f3p.h>
8  #include<mcp3421a.h>
9
10
11
12  struct i2c_struct {
13      char SlaveAddress;
14      char* masterBuffer;
15      char masterTransactionLength;
16      char buffer_index;
17      char reg;
18      char readwrite;
19      char slave_config;
20  }i2c;
21
22  struct MCP3421_conf{
23      char mode;
24      char SPS;
25      char GAIN;
26  }conf;
27
28  void config_MCP3421(MCP3421_MODE_TypeDef mode,
29                    MCP3421_SPS_TypeDef SPS,
30                    MCP3421_GAIN_TypeDef GAIN)
31  {
32      conf.mode = mode;
33      conf.SPS = SPS;
34      conf.GAIN = GAIN;
35  }
36
37  uint8_t data[3];
38  char index = 0;
39  char MCP3421_Conf;
40  unsigned short value;
41  char check,i;
42
43
44  void UART_TX(unsigned char val){
45      while (!(UART1->SR & UART1_SR_TXE));
46      UART1->DR = val;
47  }
48
49  unsigned char UART_RX(void){
50      while (!(UART1->SR & UART1_SR_RXNE));
51      return UART1->DR;
52  }
53
54  void delay(unsigned long count){
55      while (count-->0)
56          nop();
57  }
```

```

58
59 void i2c_Init(void){
60     /*----- I2C FREQ Configuration -----*/
61     /* Clear frequency bits */
62     I2C->FREQR &= (uint8_t) (~I2C_FREQR_FREQ);
63     /* Write new value */
64     I2C->FREQR |= 10; //InputClockFrequencyMHz
65
66     /*----- I2C CCR Configuration -----*/
67     I2C->TRISER = 11;
68
69     /* Write CCR with new calculated value */
70     I2C->CCRL = 0x32;
71     I2C->CCRH = 0x00;
72
73     I2C->OARH = 0b01000000; // This bit must be set by software
74     I2C->ITR |= I2C_ITR_ITEVTEN; //Event Enables :SB, ADDR, ADD10,
STOPF, BTF, WUFH
75     I2C->ITR |= I2C_ITR_ITBUFEN;
76     _asm("rim");
77     /*Enable I2C peripheral*/
78     I2C->CR1 |= I2C_CR1_PE;
79
80 }
81
82
83 void I2C_Read(char slaveAddress, char* buffer, char count)
84 {
85     i2c.readwrite |= 0x01;
86     i2c.SlaveAddress = slaveAddress;
87     i2c.buffer_index = 0;
88     i2c.masterBuffer = buffer;
89     i2c.masterTransactionLength = count;
90     //wait for the Bus to get Free to avoid collisions
91     while((I2C->SR3 & I2C_SR3_BUSY) == I2C_SR3_BUSY);
92
93     I2C->CR2 |= I2C_CR2_ACK; //Acknowledge Enable : Acknowledge returned after a
byte is received (matched address or data)
94
95     I2C->CR2 |= I2C_CR2_START; //Launch the process
96     delay(100);
97 }
98
99 void I2C_Write(char slaveAddress, char config)
100 {
101     i2c.readwrite &= 0x00;
102     i2c.SlaveAddress = slaveAddress;
103     i2c.slave_config = config;
104
105     //wait for the Bus to get Free to avoid collisions
106     while((I2C->SR3 & I2C_SR3_BUSY) == I2C_SR3_BUSY);
107
108     I2C->CR2 |= I2C_CR2_ACK; //Acknowledge Enable : Acknowledge returned after a
byte is received (matched address or data)
109     //The start will enter the Master Mode (when the Busy bit is cleared)
110     //If already in Master Mode, then ReStart will be generated at the end of the
current transfer
111     I2C->CR2 |= I2C_CR2_START; //Launch the process
112     delay(100);
113 }
114
115 void auto_GAIN(unsigned short check){
116     if((conf.SPS == SPS240_12bit) && (conf.GAIN > 0)){
117         if (check >= (2047)){
118             conf.GAIN--;
119             I2C_Write(MCP3421, (conf.mode|conf.SPS|conf.GAIN));
120         }
121     }
122 }

```

```

123
124 main()
125 {
126     CLK_CKDIVR = 0x00; //set clock 16MHZ
127     CLK_PCKENR1 = 0xFF; // set all peripherals
128
129
130     GPIO_Init(GPIOB , GPIO_PIN_4 , GPIO_MODE_IN_PU_NO_IT);
131     GPIO_Init(GPIOB , GPIO_PIN_5 , GPIO_MODE_IN_PU_NO_IT);
132     //I2C_SPR5 = 0b01000000;
133
134     UART1->CR3 |= UART1_STOPBITS_1; //stop bit
135     UART1->BRR2 = 0x03; UART1->BRR1 = 0x68; // 0x0683 coded funky way (see ref
manual) 9600
136     UART1->CR2 |= (uint8_t)UART1_CR2_TEN;//enable TX
137
138     I2C_DeInit();
139     i2c_Init();
140     config_MCP3421(CONTI_CONVERSION_MODE, SPS240_12bit, GAIN1);
141     check = conf.mode|conf.SPS|conf.GAIN;
142     I2C_Write(MCP3421, (conf.mode|conf.SPS|conf.GAIN));
143     //delay(100);
144     while (1){
145
146         I2C_Read(MCP3421,data,3);
147         value = ((data[0]<<8) | data[1]);
148         //auto_GAIN(value);
149         UART_TX(data[0]);
150         delay(300);
151         UART_TX(data[1]);
152         delay(300);
153         UART_TX(data[2]);
154         delay(300);
155     }
156 }
157
158 @far @interrupt void EXTI_I2C(void){
159     if (I2C->SR1 & I2C_SR1_SB){
160         i2c.reg = I2C->SR1;
161         I2C->DR = (i2c.SlaveAddress) | i2c.readwrite;
162     }
163     /*else if(I2C->SR1&I2C_SR1_ADDR){
164         i2c.reg = I2C->SR1;
165         i2c.reg = I2C->SR3;
166         i2c.buffer_index = 0;
167     }*/
168     else if (I2C->SR3 & I2C_SR3_TRA)
169     {
170         if (I2C->SR1 & I2C_SR1_TXE)
171         {
172             I2C->DR = i2c.slave_config;
173             //while(!(I2C->SR1&I2C_SR1_BTF == I2C_SR1_BTF));
174             I2C->CR2 |= I2C_CR2_STOP; //generate STOP
175         }
176     }
177     else
178     {
179         if(I2C->SR1&I2C_SR1_RXNE)
180         {
181             char Reg = I2C->DR;
182             if(i2c.buffer_index < i2c.masterTransactionLength)
183             {
184                 i2c.masterBuffer[i2c.buffer_index] = Reg;
185             }
186             if(i2c.buffer_index == (i2c.masterTransactionLength - 1))//The last byte is
received
187             {
188                 I2C->CR2 &= (~I2C_CR2_ACK);//Clear ACK
189                 I2C->CR2 |= I2C_CR2_STOP; //generate STOP
190             }
191             i2c.buffer_index++;
192         }
193     }
194 }
195

```