Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Институт «Электронных и информационных систем»

Кафедра «Информационных технология и систем»

Лабораторная работа №6

по дисциплине:

« ЭВМ и периферийные устройства»

**Отчёт**

Принял преподаватель:

\_\_\_\_\_\_\_\_\_\_ / Шаклеин В.Г

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Выполнил студент группы 9091:

\_\_\_\_\_\_\_\_\_\_ / Чалый С. М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

**Великий Новгород**

**2022**

Цель работы:

1. Получить практические навыки работы с шиной I2C
2. Освоить работу сразу с двумя документами (описание интерфейсного контроллера и описание управляемого по интерфейсу устройства).

Ход работы:

Изучить посылки, которые следует посылать в целевое устройство

Подобрать подходящие методы формирования требуемых посылок

Создать программу и настроить шину I2C силами генератора кода. Линию SCL подключить к порту PB8. Линию SDA – к порту PB9. Убедиться, что в контроллере включены подтягивающие резисторы. Длительность нарастающего фронта и спадающего фронта задать, согласно целевой документации. Нарастающий фронт следует брать наихудший, так как резисторы в контроллере достаточно высокоёмные.

Написать слой кода для обслуживания шины I2C, обеспечивающий функционал, требуемый для целевого устройства

Написать слой кода, обслуживающий EEPROM, обеспечивающий чтение и запись энергонезависимой памяти

В функции main() обеспечить запись на адрес 0x40 какой-либо собственной строки.

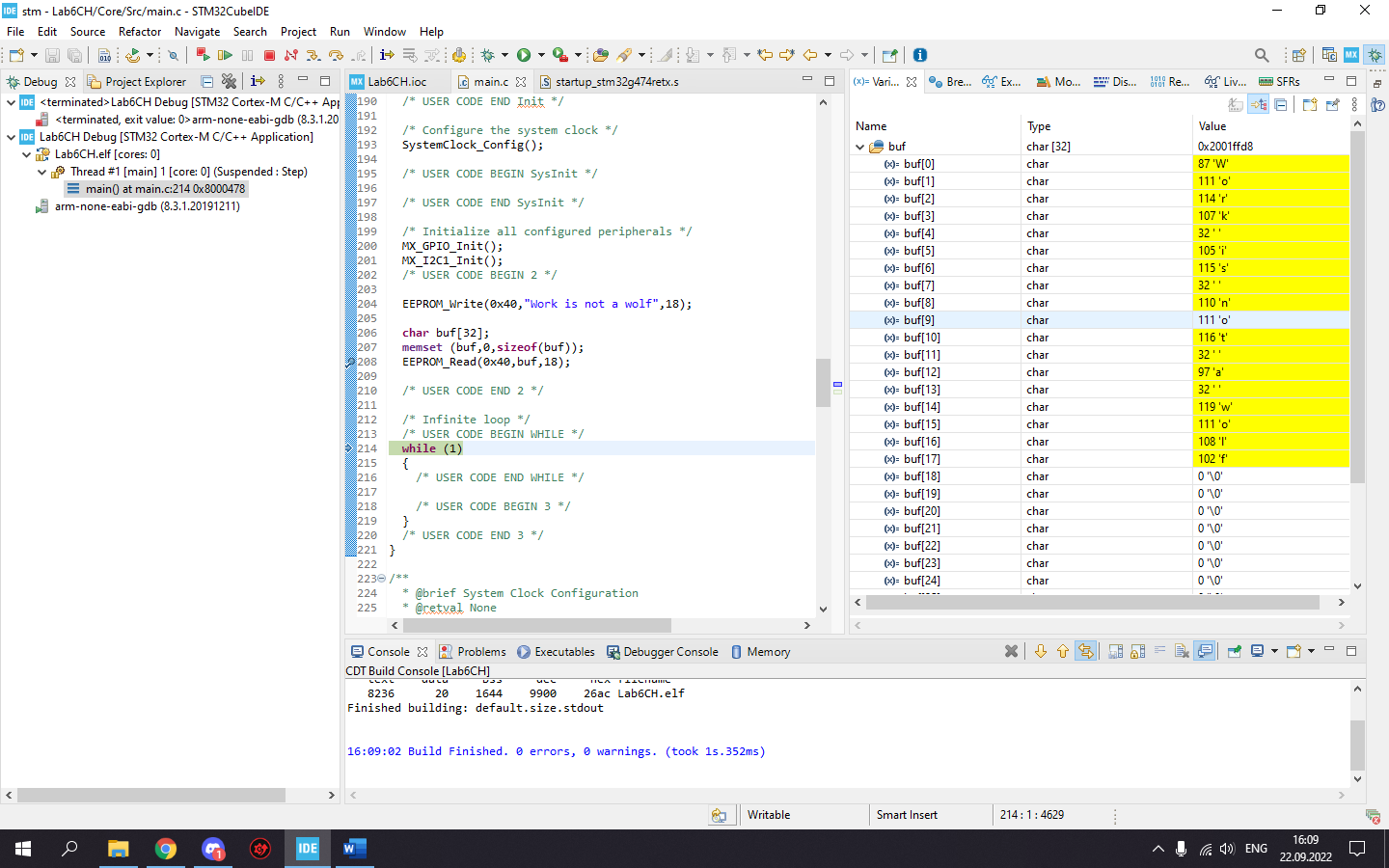
Подготовить буфер и заполните его нулями. Длина буфера должна быть не менее, чем длина только что записанной строки

Считать строку с EEPROM в этот буфер .

Убедиться, что данные принялись верно .

Повторить опыт, задав неверный адрес EEPROM на шине I2C.

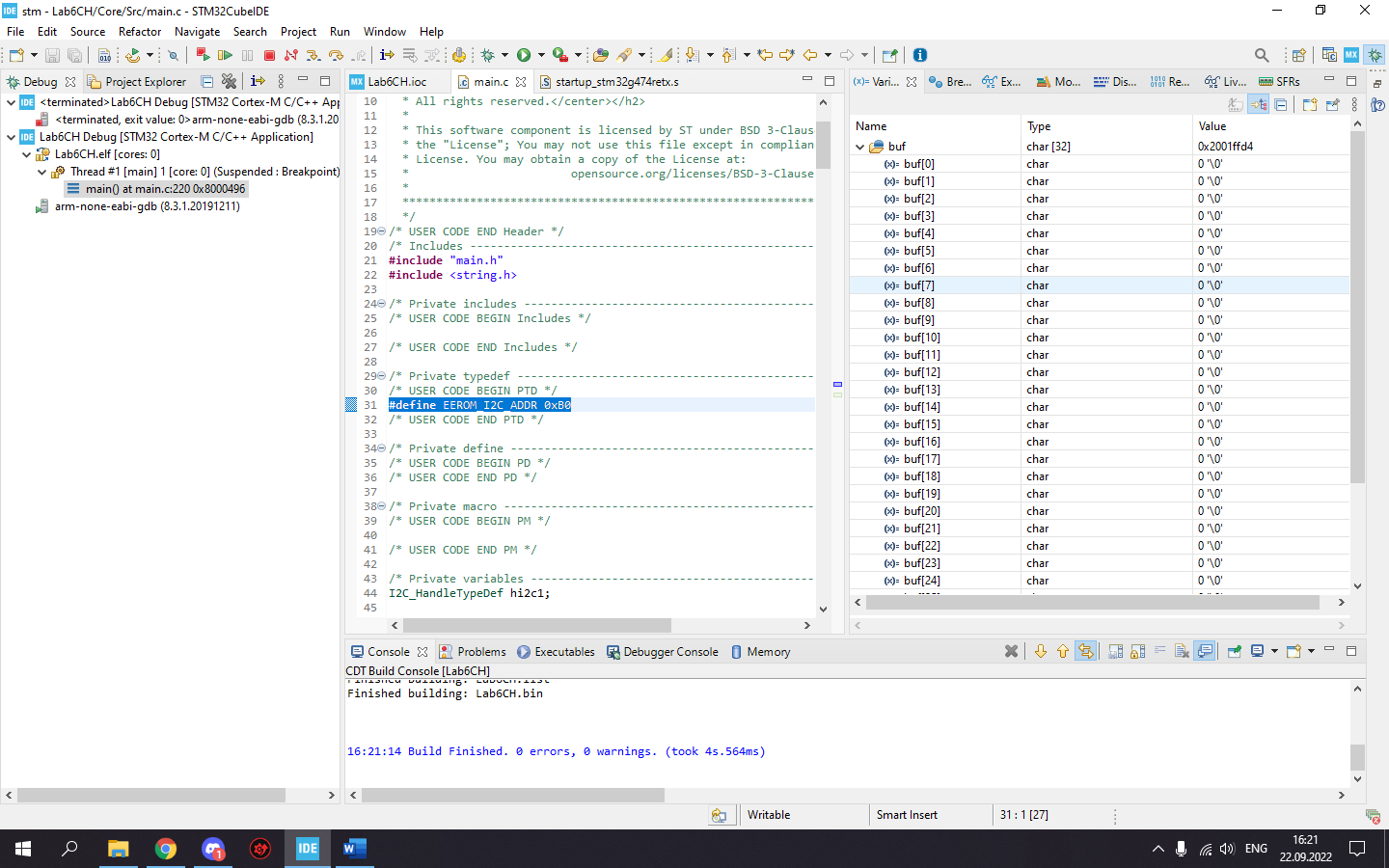
Убедиться, что не работает даже запись



Изображение выглядит как текст, внутренний, компьютер, снимок экрана

Автоматически созданное описание

Поменял адрес на шине:



**#include** "main.h"

**#include** <string.h>

/\* Private includes ----------------------------------------------------------\*/

/\* USER CODE BEGIN Includes \*/

/\* USER CODE END Includes \*/

/\* Private typedef -----------------------------------------------------------\*/

/\* USER CODE BEGIN PTD \*/

**#define** EEROM\_I2C\_ADDR 0xA0

/\* USER CODE END PTD \*/

/\* Private define ------------------------------------------------------------\*/

/\* USER CODE BEGIN PD \*/

/\* USER CODE END PD \*/

/\* Private macro -------------------------------------------------------------\*/

/\* USER CODE BEGIN PM \*/

/\* USER CODE END PM \*/

/\* Private variables ---------------------------------------------------------\*/

I2C\_HandleTypeDef hi2c1;

**int** **I2CWrite**(**char**\* pData, **int** cnt)

{

**int** i;

**for**(i=0; i<cnt; i++)

{

**while**((I2C1->ISR & I2C\_ISR\_TXIS)==0)

{

}

I2C1->TXDR = pData [i];

}

**while**((I2C1->ISR & I2C\_ISR\_TXE)==0)

{

}

**return** i;

}

/\* USER CODE BEGIN PV \*/

**int** **I2CStartForWrite**(uint32\_t i2c\_addr, **int** totalCnt)

{

**while**(I2C1->ISR & I2C\_ISR\_BUSY)

{

}

I2C1->ICR=I2C\_ICR\_NACKCF | I2C\_ICR\_STOPCF;

I2C1->CR2 &= ~(I2C\_CR2\_RD\_WRN | I2C\_CR2\_AUTOEND | I2C\_CR2\_NBYTES\_Msk | I2C\_CR2\_SADD\_Msk);

I2C1->CR2 |= (i2c\_addr<< I2C\_CR2\_SADD\_Pos) | (totalCnt<< I2C\_CR2\_NBYTES\_Pos);

I2C1->CR2 |= I2C\_CR2\_START;

**while** ((I2C1->ISR & I2C\_ISR\_TXIS)==0)

{

**if**(I2C1->ISR & I2C\_ISR\_NACKF)

{

**return** -1;

}

}

**return** 0;

}

**void** **I2CStop**(**void**)

{

I2C1->CR2 |= I2C\_CR2\_STOP;

**while**((I2C1->ISR & I2C\_ISR\_STOPF)==0)

{

}

}

**int** **I2CRead** (**char**\* pData, **int** cnt)

{

I2C1->CR2 &= ~(I2C\_CR2\_AUTOEND | I2C\_CR2\_NBYTES\_Msk);

I2C1->CR2 |= I2C\_CR2\_RD\_WRN |(((uint8\_t)cnt)<<I2C\_CR2\_NBYTES\_Pos);

I2C1->CR2 |= I2C\_CR2\_START;

**int** i;

**for**(i=0; i<cnt;i++)

{

**while**((I2C1->ISR & I2C\_ISR\_RXNE)==0)

{

}

pData[i] = (**char**)I2C1->RXDR;

}

**return** i;

}

**void** **EEPROM\_SendAddress**(uint32\_t addr)

{

uint8\_t buf[2];

buf[0] = (uint8\_t) (addr>>8);

buf[1] = (uint8\_t) (addr);

I2CWrite((**char**\*)buf,2);

}

**int** **EEPROM\_Write**(uint32\_t addr, **char**\* pData, **int** cnt)

{

**int** res= I2CStartForWrite(EEROM\_I2C\_ADDR, 2+cnt);

**if**(res>=0)

{

EEPROM\_SendAddress(addr);

res=I2CWrite(pData,cnt);

I2CStop();

}

**return** res;

}

**int** **EEPROM\_Read**(uint32\_t addr,**char**\* pData,**int** cnt)

{

**int** res= I2CStartForWrite(EEROM\_I2C\_ADDR, 2);

**if**(res>=0)

{

EEPROM\_SendAddress(addr);

res=I2CRead(pData,cnt);

I2CStop();

}

**return** res;

}

EEPROM\_Write(0x40,"Work is not a wolf",18);

**char** buf[32];

**memset** (buf,0,**sizeof**(buf));

**for** (**int** retry=0; retry<1000;retry++)

{

**if** (EEPROM\_Read(0x40,buf,18)>=0)

{

**break**;

}

}

Вывод:

Таким образом, я получил практические навыки работы с шиной I2C , освоив работу с описанием интерфейсного контроллера и описанием управляемого по интерфейсу устройства.