**Комплект оценочных материалов по дисциплине
«Компьютерные и информационные технологии в отрасли»**

**Задания закрытого типа**

**Задания закрытого типа на выбор правильного ответа**

*Выберите один правильный ответ*

1. Какой интерфейс обладает наивысшей помехоустойчивостью при обмене данными?

А) RS232;

Б) RS485;

В) CL (токовая петля);

Г) RS423;

Д) правильный вариант отсутствует.

Правильный ответ: В

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

2. Какой интерфейс обладает наихудшей помехоустойчивостью при обмене данными?

А) RS232;

Б) RS485;

В) CL (токовая петля);

Г) RS423;

Д) правильный вариант отсутствует.

Правильный ответ: А

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

3. Какой интерфейс работает только в полудуплексном режиме?

А) RS232;

Б) RS485;

В) RS422;

Г) RS423;

Д) правильный вариант отсутствует.

Правильный ответ: Б

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

4. Какой интерфейс использует в качестве информационного сигнала значение электрического тока?

А) RS232;

Б) RS485;

В) CL (токовая петля);

Г) RS423;

Д) правильный вариант отсутствует.

Правильный ответ: В

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

5. Какой интерфейс использует в качестве информационного сигнала значение электрического тока?

А) RS232;

Б) RS485;

В) CL (токовая петля);

Г) RS423;

Д) правильный вариант отсутствует.

Правильный ответ: Б

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

6. Какой директивой на языке Arduino C для AVR-микроконтроллерa ATmega328 определяется скорость универсального асинхронного приемника-передатчика UART?

А) Serial.end();

Б) Serial.begin(115200);

В) Serial.print("115200");

Г) Serial.write("115200");

Д) правильный вариант отсутствует.

Правильный ответ: Б

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

7. Какой директивой на языке Arduino C для AVR-микроконтроллеров определяется скорость универсального асинхронного приемника-передатчика UART равной 115200 бод с 8 битами данных, двумя стоповыми битами, без проверки на чётность/нечётность?

А) Serial.begin(115200, SERIAL\_8E2);

Б) Serial.begin(115200, SERIAL\_8O2);

В) Serial.begin(115200, SERIAL\_8N1);

Г) Serial.begin(115200, SERIAL\_8N2);

Д) правильный вариант отсутствует.

Правильный ответ: Г

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

8. Какой директивой на языке Arduino C для AVR-микроконтроллеров определяется скорость универсального асинхронного приемника-передатчика UART равной 19200 бод с 7 битами данных, одним стоповым битом, без проверки на чётность/нечётность?

А) Serial.begin(19200, SERIAL\_7E1);

Б) Serial.begin(19200, SERIAL\_7O2);

В) Serial.begin(19200, SERIAL\_7N1);

Г) Serial.begin(19200, SERIAL\_7N2);

Д) правильный вариант отсутствует.

Правильный ответ: В

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

9. Какой директивой на языке Arduino C для AVR-микроконтроллеров определяется скорость универсального асинхронного приемника-передатчика UART равной 9600 бод с 8 битами данных, одним стоповым битом, с проверкой на чётность?

А) Serial.begin(19200, SERIAL\_8E1);

Б) Serial.begin(19200, SERIAL\_8O2);

В) Serial.begin(19200, SERIAL\_8N1);

Г) Serial.begin(19200, SERIAL\_8N2);

Д) правильный вариант отсутствует.

Правильный ответ: А

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

**Задания закрытого типа на установление соответствия**

*Установите правильное соответствие.*

*Каждому элементу левого столбца соответствует только один элемент правого столбца.*

1. Установите соответствие между атрибутом функции последовательного порта Serial.begin на языке Arduino C и его описанием.

|  |  |
| --- | --- |
| 1) SERIAL\_7N1 | А) 8 бит данных, 2 стоповых бита, проверка на чётность; |
| 2) SERIAL\_8N1 | Б) 7 бит данных, 1 стоповый бит, без проверки на чётность/нечетность; |
| 3) SERIAL\_8E2 | В) 8 бит данных, 2 стоповых бита, проверка на нечётность; |
| 4) SERIAL\_8O2 | Г) 8 бит данных, 1 стоповый бит, без проверки на чётность/нечетность; |

Правильный ответ:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Б | Г | А | В |

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

2. Установите соответствие обозначений выводов AVR-микроконтроллеров их назначению.

|  |  |
| --- | --- |
| 1) SS | А) асинхронный обмен данными по интерфейсу UART |
| 2) MOSI, MISO | Б) тактирование ведомого устройства при синхронном обмене данными |
| 3) TXD, RXD | В) синхронный обмен данными по интерфейсу SPI |
| 4) SCK | Г) выбор ведомого устройства при синхронном обмене данными по интерфейсу SPI |

Правильный ответ:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Г | В | А | Б |

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

3. Установите соответствие команд и директив на языке Arduino C их описанию.

|  |  |
| --- | --- |
| 1) #include <Wire.h> | А) инициализация шины I2C  |
| 2) Wire.begin(); | Б) подключение библиотеки для I2C |
| 3) Wire.onRequest(requestEvent); | В) запрос данных по шине I2C |
| 4) Wire.write("hello"); | Г) запись строки побайтно по шине I2C |

Правильный ответ:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Б | А | В | Г |

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

**Задания закрытого типа на установление правильной последовательности**

*Установите правильную последовательность.*

*Запишите правильную последовательность букв слева направо.*

1. Установите правильную последовательность команд и директив при передаче данных (переменная f – типа float (число с плавающей точкой)) по интерфейсу UART от микроконтроллера ATmega328.

А). for (byte i:=0; i<16; i++) {

Б). Serial.begin(115200);

В).char С [16] = "";

Г). Serial.print(char С [16]);

Д). dtostrf(f, 16, 2, С);

Правильный ответ: В, Б, Д, А, Г

Компетенции (индикаторы): УК-2 (УК-2.2); УК-4(УК-4.2); ОПК-1 (ОПК-1.2)

2. Установите правильную последовательность команд процесса записи данных по шине I2C.

А) Wire.write((int)(eeaddress >> 8));

Б) Wire.beginTransmission(deviceaddress);

В) Wire.write(data);

Г) Wire.endTransmission();

Д) Wire.write((int)(eeaddress & 0xFF));

Правильный ответ: Б, А, Д, В, Г

Компетенции (индикаторы): УК-2 (УК-2.2); УК-4(УК-4.2); ОПК-1 (ОПК-1.2)

3. Установите правильную последовательность команд процесса запроса на чтение данных по шине I2C.

А). Wire.endTransmission();

Б). Wire.requestFrom(deviceaddress,1);

В) Wire.write((int)(eeaddress >> 8));

Г) Wire.beginTransmission(deviceaddress);

Д) Wire.write((int)(eeaddress & 0xFF));

Правильный ответ: Г, В, Д, А, Б

Компетенции (индикаторы): УК-2 (УК-2.2); УК-4(УК-4.2); ОПК-1 (ОПК-1.2)

**Задания открытого типа**

**Задания открытого типа на дополнение**

*Напишите пропущенное слово (словосочетание).*

1. Программа, управляющая каким-либо внешним или внутренним устройством компьютера (сканером, мышью, клавиатурой, дисководом:), называется \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Правильный ответ: Драйвером

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

2. Структура, представляющая собой несколько связанных между собой специальным кабелем компьютеров, между которыми осуществляется передача данных, называется \_\_\_\_\_\_\_\_\_

Правильный ответ: Локальной сетью

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

3. Программы и аппаратура, обеспечивающие связь между собой двух и более объектов, называется \_\_\_\_\_\_\_\_\_

Правильный ответ: Интерфейсом связи

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

4. Предназначенное для конкретного исполнителя точное описание последовательности действий, направленных на решение поставленной задачи, называется \_\_\_\_\_\_\_\_\_

Правильный ответ: Алгоритмом

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

5. Часть процессора, которая производит выполнение логических и арифметических операций, предусмотренных архитектурой процессора, называется \_\_\_\_\_\_\_\_\_\_\_

Правильный ответ: Арифметико-логическим устройством (АЛУ)

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

6. Единица пропускной способности канала связи, равная 1 бит/с, называется \_\_\_\_\_\_\_\_\_\_\_

Правильный ответ: Бодом

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

7. Механизм, обеспечивающий доступ к элементу массива посредством ссылки на массив и на одно или несколько выражений, значения которых определяют позицию этого элемента в массиве, называется \_\_\_\_\_\_\_\_\_\_\_

Правильный ответ: Индексацией

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

8. Часть сети, связывающая между собой каждую пару ее оконечных терминалов и состоящая из технических средств передачи и приема данных, включая линию связи, а также средств программного обеспечения и протоколов, называется \_\_\_\_\_\_\_\_\_\_\_

Правильный ответ: Каналом связи

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

9. Объект индексируемого типа, значения которого – это элементы из значений одного и того же типа, называется \_\_\_\_\_\_\_\_\_\_\_

Правильный ответ: Массивом

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

10. Элемент данных, который представляется выражением и над которым выполняются операции, называется \_\_\_\_\_\_\_\_\_\_\_

Правильный ответ: Операндом

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

11. Упорядоченная последовательность команд (инструкций), необходимых процессору для решения поставленной задачи, называется \_\_\_\_\_\_\_\_\_\_\_

Правильный ответ: Программой

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

12. Устройство для записи, хранения и выдачи по запросу потоков данных, называется \_\_\_\_\_\_\_\_\_\_\_

Правильный ответ: Запоминающим устройством (ЗУ)

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

**Задания открытого типа с кратким свободным ответом**

1. Сложите арифметически два, приведенных ниже, восьмиразрядных числа с указанием переносов. Проверьте результат сложения путем перевода слагаемых и результата из двоичной системы в десятичную систему.

|  |  |
| --- | --- |
| **+** | 10110110 |
| 01101110 |

Правильный ответ: 0b100100100 и 182+110=292 / 100100100 и 182+110=292

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

2. Выполните логическое умножение двух, следующих восьмиразрядных чисел. Предоставьте результат логического умножения и значения множителей, как в двоичной системе, так и в десятичной системе

|  |  |
| --- | --- |
|  | 01010110 |
| 01111101 |

Правильный ответ: 0b01010100, 86, 125 и 84 / 01010100, 86, 125 и 84

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

3. Выполните операцию «исключающее ИЛИ», следующих восьмиразрядных чисел. Предоставьте результат операции и значения операндов, как в двоичной системе, так и в десятичной системе.

|  |  |
| --- | --- |
| ⊕ | 00111110 |
| 01001101 |

Правильный ответ: 0b01110011, 62, 77 и 115 / 01110011, 62, 77 и 115

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

4. Выполните операцию «ИЛИ-НЕ», следующих восьмиразрядных чисел. Предоставьте результат операции и значения операндов, как в двоичной системе, так и в десятичной системе.

|  |  |
| --- | --- |
|  | 11101000 |
| 00101101 |

Правильный ответ: 0b00010010, 232, 45 и 18/ 00010010, 232, 45 и 18

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

5.Выполните операцию «И-НЕ», следующих восьмиразрядных чисел. Предоставьте результат операции и значения операндов, как в двоичной системе, так и в десятичной системе.

|  |  |
| --- | --- |
| x | 11010010 |
| 01001010 |

Правильный ответ: 0b10111101, 210, 74 и 189 / 10111101, 210, 74 и 189

Компетенции (индикаторы): УК-2 (УК-2.1); УК-4(УК-4.1); ОПК-1 (ОПК-1.1)

**Задания открытого типа с развернутым ответом**

1. Рассчитайте время передачи 45 символов при скорости передачи UART равной 9600 бод при тактовой частоте процессора f=1 Мгц. Приведите пояснения выполняемых расчётов. Дайте оценку полученному результату – насколько быстро будет происходить обмен данными.

Время выполнения – 45 мин.

Ожидаемый результат:

1. Определим время передачи одного бита:

tBOUD=1/BOUD=1/9600=1,0417⸱10-4 сек.

Здесь BOUD скорость передачи по универсальному асинхронному порту. Бод – это единица скорости при обмене данными, она равна отношению бит/сек.

2. Время передачи одного символа:

tASCII=11⸱tBOUD=11⸱1,0417⸱10-4 = 1,146⸱10-3 сек.

Здесь число 11 – это количество бит, необходимых для передачи одного символа: один стоповый бит, 8 бит данных, 2 стоповых бита (или 1бит четности/нечётности и 1 стоповый бит).

3. Время на передачу 45 символов определится как:

t45=11⸱tASCII=45⸱1,146⸱10-3 = 0,0516 сек.

В зависимости от объема передаваемых данных можно определить достаточно ли данной скорости или нет. При небольших объёмах данных, таких как в задаче – 45 символов, этой скорости достаточно для быстрого обмена данными – 0,0516 сек. Чем больше объём передаваемых данных, тем выше должна быть скорость UART. Если время передачи данных превышает допустимое, то необходимо увеличить скорость UART. Если это уже невозможно – достигнут максимум, то по возможности следует разбить данные на несколько пакетов.

Ответ: 0,0516 с./0,052 с./0,05 с./ 0,0516 сек./0,052 сек./0,05 сек./ 0,0516/ 0,052 /0,05.

Критерии оценивания:

- задание считается выполненным, если правильно рассчитано время передачи 45 символов.

Компетенции (индикаторы): УК-2 (УК-2.3); УК-4(УК-4.3); ОПК-1 (ОПК-1.3)

2. Рассчитайте время передачи 648 символов при скорости передачи UART равной 57600 бод при тактовой частоте процессора f=8 Мгц. Приведите пояснения выполняемых расчётов. Дайте оценку полученному результату – насколько быстро будет происходить обмен данными.

Время выполнения – 45 мин.

Ожидаемый результат:

1. Определим время передачи одного бита:

tBOUD=1/BOUD=1/57600=1,736⸱10-5 сек.

Здесь BOUD скорость передачи по универсальному асинхронному порту. Бод – это единица скорости при обмене данными, она равна отношению бит/сек.

2. Время передачи одного символа:

tASCII=11⸱tBOUD=11⸱1,736⸱10-5  = 1,91⸱10-4 сек.

Здесь число 11 – это количество бит, необходимых для передачи одного символа: один стоповый бит, 8 бит данных, 2 стоповых бита (или 1бит четности/нечётности и 1 стоповый бит).

3. Время на передачу 45 символов определится как:

t45=11⸱tASCII=648⸱1,91⸱10-4 = 0,1238 сек.

В зависимости от объема передаваемых данных можно определить достаточно ли данной скорости или нет. При значительных объёмах данных, таких как в задаче – 648 символов, этой скорости достаточно для быстрого обмена данными – 0,1238 сек. Чем больше объём передаваемых данных, тем выше должна быть скорость UART. Если время передачи данных превышает допустимое, то необходимо увеличить скорость UART. Если это уже невозможно – достигнут максимум, то по возможности следует разбить данные на несколько пакетов.

Ответ: 0,1238 с./ 0,124 с./ 0,12 с./ 0,1238 сек./0,124 сек./0,12 сек./ 0,1238/ 0,124 /0,12.

Критерии оценивания:

- задание считается выполненным, если правильно рассчитано время передачи 648 символов.

Компетенции (индикаторы): УК-2 (УК-2.3); УК-4(УК-4.3); ОПК-1 (ОПК-1.3)

3. Допишите блок команд, которые разрешат прерывание по пустому регистру данных универсального асинхронного приемника-передатчика UART? В подпрограмме обработки прерываний загрузить любой символ в регистр данных.

char С [16] = "";

void setup() {

interrupts();

Serial.print("Hello");

}

void loop() {

for (byte i:=0; i<=127; i++)

Serial.print(char (i));

for (byte i:=0; i<32; i++) {

if (Serial.available() > 0)

C[I]= Serial.read();

}

lcd.clear();

lcd.setCursor (0, 0);

lcd.print (С);

Serial.print("Hello");

}

Время выполнения – 45 мин.

Ожидаемый результат:

В данном коде необходимо разрешить индивидуально прерывание по пустому регистру данных UART – установить бит такого разрешения UDRIE0 в регистре UCSR0B. Так же необходимо разрешить прием и передачу данных через выводы RXD и TXD.

char С [16] = "";

void setup() {

interrupts();

UCSR0B = 1<<UDRIE0$

UCSR0B = (1<<RXEN0)|(1<<TXEN0);

UCSR0C = (1<<UCSZ01)|(1<<UCSZ00); //размер слова 8 разрядов

Serial.print("Hello");

}

void loop() {

for (byte i:=0; i<=255; i++)

Serial.print(char (i));

for (byte i:=0; i<16; i++) {

if (Serial.available() > 0)

C[I]= Serial.read();

}

lcd.clear();

lcd.setCursor (0, 0);

lcd.print (С);

Serial.print("Hello");

}

ISR(USART\_UDRE\_vect)

{

UDR0 = 'E';

}

Для вводимого прерывания необходим обработчик прерываний ISR(USART\_UDRE\_vect) – это стандартный обработчик прерывания по пустому регистру данных. В обработчике прерываний загружаем символ E в регистр данных UDR0.

Ответ: UCSR0B = 1<<UDRIE0 и ISR(USART\_UDRE\_vect){ UDR0 = 'E';}

Критерии оценивания:

- задание считается выполненным, если в регистр UCSR0B установлен бит UDRIE0 или записана подпрограмма обработки прерываний ISR(USART\_UDRE\_vect).

Компетенции (индикаторы): УК-2 (УК-2.3); УК-4(УК-4.3); ОПК-1 (ОПК-1.3)

4. Допишите блок команд, которые позволят прерывание по завершению приема данных универсального асинхронного приемника-передатчика UART? В подпрограмме обработки прерываний загрузить любой символ в регистр данных.

char С [16] = "";

void setup() {

interrupts();

Serial.print("Hello");

}

void loop() {

for (byte i:=0; i<=255; i++)

Serial.print(char (i));

for (byte i:=0; i<16; i++) {

if (Serial.available() > 0)

C[I]= Serial.read();

}

lcd.clear();

lcd.setCursor (0, 0);

lcd.print (С);

Serial.print("Hello");

}

Время выполнения – 45 мин.

Ожидаемый результат:

В данном коде необходимо разрешить индивидуально прерывание по завершению приёма символа через UART – установить бит такого разрешения RXCIE0 в регистре UCSR0B. Так же необходимо разрешить прием и передачу данных через выводы RXD и TXD.

char С [16] = "";

void setup() {

interrupts();

UCSR0B = 1<<RXCIE0;

UCSR0B = (1<<RXEN0)|(1<<TXEN0);

UCSR0C = (1<<UCSZ01)|(1<<UCSZ00); //размер слова 8 разрядов

Serial.print("Hello");

}

void loop() {

for (byte i:=0; i<=255; i++)

Serial.print(char (i));

for (byte i:=0; i<16; i++) {

if (Serial.available() > 0)

C[I]= Serial.read();

}

lcd.clear();

lcd.setCursor (0, 0);

lcd.print (С);

Serial.print("Hello");

}

ISR(USART0\_RX\_vect)

{

UDR0 = 'E';

}

Для вводимого прерывания необходим обработчик прерываний ISR(USART0\_RX\_vect) – это стандартный обработчик прерывания по завершению приема данных. В обработчике прерываний загружаем символ *E* в регистр данных UDR0.

Ответ: UCSR0B = 1<< RXCIE0 и ISR(USART0\_RX\_vect){ UDR0 = 'E';}

Критерии оценивания:

-задание считается выполненным, если в регистр UCSR0B установлен бит RXCIE0 или записана подпрограмма обработки прерываний ISR(USART0\_RX\_vect).

Компетенции (индикаторы): УК-2 (УК-2.3); УК-4(УК-4.3); ОПК-1 (ОПК-1.3)

5. В подпрограмме записи массива во внешнюю ЭСППЗУ укажите, где осуществляется установка внутреннего адреса ЭСППЗУ. Разделите внешнюю ЭСППЗУ объемом 256 байт на 16 банков и осуществите запись массива mass\_o в 4-й банк данных.

#include <Wire.h>

#define eeprom 0x50

unsigned int p\_address = 0;

byte mass\_o [16];

void setup() {

Wire.begin();

}

void PP\_SEND()

{

do { } while (SEND==true);

for (unsigned int p\_index=0; p\_index<=15; p\_index++) {

p\_address= p\_index;

writeEEPROM (eeprom, p\_address, mass\_o[p\_index]);

delay (80);

}

lcd.clear();

lcd.setCursor (1, 0);

lcd.print ("DATA IS SEND!");

lcd.setCursor (5, 1);

lcd.print ("Bank=");

lcd.print (p\_Bank);

}

void writeEEPROM(int deviceaddress, unsigned int eeaddress, byte data ) {

 Wire.beginTransmission(deviceaddress);

 Wire.write((int)(eeaddress >> 8));

 Wire.write((int)(eeaddress & 0xFF));

 Wire.write(data);

 Wire.endTransmission();

 }

Время выполнения – 45 мин.

Ожидаемый результат:

В данном коде необходимо ввести переменную-указатель, которая будет содержать номер банка данных, но первый банк будет иметь значение 0 , а не 1, второй 1, а не 0 и т.д. Во второй аргумент функции writeEEPROM записывают внутренний адрес ЭСППЗУ (подчеркнуто).

#include <Wire.h>

#define eeprom 0x50

unsigned int p\_address = 0;

byte mass\_o [16];

unsigned int p\_bank = 0; //ставим указатель на 0 – первый банк

void setup() {

Wire.begin();

}

void PP\_SEND()

{

do { } while (SEND==true);

p\_bank = 3; //устанавливаем указатель на 4-й банк

for (unsigned int p\_index=0; p\_index<=15; p\_index++) {

p\_address= p\_index;

writeEEPROM (eeprom, p\_address+(p\_bank\*16), mass\_o[p\_index]);

delay (80);

}

lcd.clear();

lcd.setCursor (1, 0);

lcd.print ("DATA IS SEND!");

lcd.setCursor (5, 1);

lcd.print ("Bank=");

lcd.print (p\_Bank);

}

void writeEEPROM(int deviceaddress, unsigned int eeaddress, byte data ) {

 Wire.beginTransmission(deviceaddress);

 Wire.write((int)(eeaddress >> 8));

 Wire.write((int)(eeaddress & 0xFF));

 Wire.write(data);

 Wire.endTransmission();

 }

Во второй аргумент функции writeEEPROM добавлено слагаемое p\_bank\*16 для того, чтобы было можно перемещаться между банками данных. ЭСППЗУ (EEPROM) состоит из 256 ячеек, которые разделены условно на 16 банков данных. Тогда в одном банке будет 256/16=16 ячеек. В одну ячейку можно записать один байт массива mass\_o.

Ответ: второй аргумент – p\_address+(p\_bank\*16) – в функции
writeEEPROM (…., …, ….) / writeEEPROM (…, p\_address+(p\_bank\*16), …); / p\_address+(p\_bank\*16) или(и) 16 ячеек в 1 банке/ 16 ячеек/ 16.

Критерии оценивания:

- задание считается выполненным, если указано в программе, где осуществляется установка внутреннего адреса ЭСППЗУ или же приведено пояснение как осуществляется разделение ЭСППЗУ на банки.

Компетенции (индикаторы): УК-2 (УК-2.3); УК-4(УК-4.3); ОПК-1 (ОПК-1.3)

6. В подпрограмме записи массива во внешнюю ЭСППЗУ укажите, где осуществляется установка внешнего адреса ЭСППЗУ. Разделите внешнюю ЭСППЗУ объемом 256 байт на 8 банков и осуществите запись массива mass\_o в 6-й банк данных.

#include <Wire.h>

#define eeprom 0x50

unsigned int p\_address = 0;

word mass\_o [16];

void setup() {

Wire.begin();

}

void PP\_SEND()

{

do { } while (SEND==true);

for (unsigned int p\_index=0; p\_index<=15; p\_index++) {

p\_address= p\_index;

writeEEPROM (eeprom, 2\*p\_address, HighByte(mass\_o[p\_index]));

delay (80);

writeEEPROM (eeprom, 2\*p\_address+1, LowByte (mass\_o[p\_index]));

delay (80);

}

lcd.clear();

lcd.setCursor (1, 0);

lcd.print ("DATA IS SEND!");

lcd.setCursor (5, 1);

lcd.print ("Bank=");

lcd.print (p\_Bank);

}

void writeEEPROM(int deviceaddress, unsigned int eeaddress, byte data ) {

 Wire.beginTransmission(deviceaddress);

 Wire.write((int)(eeaddress >> 8));

 Wire.write((int)(eeaddress & 0xFF));

 Wire.write(data);

 Wire.endTransmission();

 }

Время выполнения – 45 мин.

Ожидаемый результат:

В данном коде необходимо ввести переменную-указатель, которая будет содержать номер банка данных, но первый банк будет иметь значение 0 , а не 1, второй 1, а не 0 и т.д. Внешний адрес ЭСППЗУ (подчеркнуто) объявляется с помощью директивы #define, а затем используется в функции передачи данных на ЭСППЗУ writeEEPROM() – первый аргумент (подчеркнуто).

#include <Wire.h>

#define eeprom 0x50

unsigned int p\_address = 0;

byte mass\_o [16];

unsigned int p\_bank = 0; //ставим указатель на 0 – первый банк

void setup() {

Wire.begin();

}

void PP\_SEND()

{

do { } while (SEND==true);

p\_bank = 5; //устанавливаем указатель на 6-й банк

for (unsigned int p\_index=0; p\_index<=15; p\_index++) {

p\_address= p\_index;

writeEEPROM (eeprom, 2\*p\_address+(p\_bank\*32), HighByte(mass\_o[p\_index]));

delay (80);

writeEEPROM (eeprom, 2\*p\_address+1+(p\_bank\*32), LowByte (mass\_o[p\_index]));

delay (80);

}

lcd.clear();

lcd.setCursor (1, 0);

lcd.print ("DATA IS SEND!");

lcd.setCursor (5, 1);

lcd.print ("Bank=");

lcd.print (p\_Bank);

}

void writeEEPROM(int deviceaddress, unsigned int eeaddress, byte data ) {

 Wire.beginTransmission(deviceaddress);

 Wire.write((int)(eeaddress >> 8));

 Wire.write((int)(eeaddress & 0xFF));

 Wire.write(data);

 Wire.endTransmission();

 }

Во второй аргумент функции writeEEPROM добавлено слагаемое p\_bank\*32 для того, чтобы было можно перемещаться между банками данных. ЭСППЗУ (EEPROM) состоит из 256 ячеек, которые разделены условно на 8 банков данных. Тогда в одном банке будет 256/8=32 ячейки. В одну ячейку можно записать один байт массива mass\_o. Массив mass\_o состоит из 16-битных элементов типа word, т.е. под один элемент необходимо 2 байта (2 ячейки). Массив mass\_o, состоящий из 16 элементов типа word можно разместить в 32 ячейках, т.е. в одном банке данных ЭСППЗУ.

Ответ: второй аргумент – 2\*p\_address+(p\_bank\*32) – в функции
writeEEPROM (…., …, ….) / writeEEPROM (…,2\*p\_address+(p\_bank\*32), …); / 2\*p\_address+(p\_bank\*32) или(и) 32 ячейки в 1 банке/ 32 ячейки/ 32.

Критерии оценивания:

- задание считается выполненным, если указано в программе, где осуществляется установка внутреннего адреса ЭСППЗУ или же приведено пояснение как осуществляется разделение ЭСППЗУ на банки.

Компетенции (индикаторы): УК-2 (УК-2.3); УК-4(УК-4.3); ОПК-1 (ОПК-1.3)