**Комплект оценочных материалов по дисциплине
«Микропроцессорные системы релейной защиты и автоматики»**

**Задания закрытого типа**

**Задания закрытого типа на выбор правильного ответа**

*Выберите один правильный ответ*

1. Какой директивой на языке Pascal-SCM для AVR-микроконтроллеров определяется скорость универсального асинхронного приемника-передатчика UART?

А) StackSize = $0032, iData;

Б) FrameSize = $0032, iData;

В) SerPort = 9600 Stop1;

Г) TxBuffer = $0032;

Д) верного ответа нет.

Правильный ответ: В

Компетенции (индикаторы): ПК-3 (ПК-3.1)

2. Какой директивой на языке Pascal-SCM для AVR-микроконтроллеров определяется объем буфера приемника UART?

А) StackSize = $0032, iData;

Б) FrameSize = $0032, iData;

В) SerPort = 9600 Stop1;

Г) RxBuffer = 8;

Д) верного ответа нет.

Правильный ответ: Г

Компетенции (индикаторы): ПК-3 (ПК-3.1)

3. Какой директивой на языке Pascal-SCM для AVR-микроконтроллеров определяется скорость универсального асинхронного приемника-передатчика UART равной 115200 бод с двумя стоповыми битами?

A) StackSize=115200,2iData;

Б) SpeedSize=115200, 2iData;

В) SerPort=115200, Stop2;

Г) UARTPort=115200, Stop2;

Д) верного ответа нет.

Правильный ответ: В

Компетенции (индикаторы): ПК-3 (ПК-3.1)

**Задания закрытого типа на установление соответствия**

*Установите правильное соответствие.*

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

1. Установите соответствие предложенных названий регистров управления их периферийным устройствам в AVR-микроконтроллерах.

|  |  |
| --- | --- |
| 1) StackSize = $0032, iData; | А) размер кадра в памяти данных |
| 2) FrameSize = $0032, iData; | Б) объем буфера приемника |
| 3) SerPort = 9600, Stop1; | В) скорость UART и количество стоповых бит |
| 4) RxBuffer = 8; | Г) Размер стека в памяти данных |

Правильный ответ:

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

Компетенции (индикаторы): ПК-3 (ПК-3.1)

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): ПК-3 (ПК-3.1)

3. Установите соответствие предложенных инструкций на языке Pascal-SCM для AVR-микроконтроллерах их описанию.

|  |  |
| --- | --- |
| 1) excl (PB0); | А) сформировать выдержку времени 200 мс. |
| 2) mdelay (200); | Б) сформировать выдержку времени 200 мкс. |
| 3) udelay (20); | В) установить «0» на выводе PB0 |
| 4) DisableInts; | Г) запрет глобального прерывания |

Правильный ответ:

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

Компетенции (индикаторы): ПК-3 (ПК-3.1)

**Задания закрытого типа на установление правильной последовательности**

*Установите правильную последовательность.*

*Запишите правильную последовательность букв слева направо.*

1. Установите правильную последовательность команд программного кода на языке Pascal-SCM для AVR-микроконтроллерах при выводе информации на жидкокристаллический индикатор с задержкой для фиксации оператором.

А) LCDxy(0, 0);

Б) mdelay(1000);

В) Write(LCDout, floattostr (t:4:1));

Г) Write(LCDout, 't=');

Д) Write(LCDout, 'C ');

Правильный ответ: А, Г, В, Д, Б

Компетенции (индикаторы): ПК-3 (ПК-3.1)

2. Расположите вектора прерываний и сброса для AVR-микроконтроллеров, начиная с вектора имеющего наивысший приоритет и далее по убыванию.

А) INT1;

Б) RESET;

В) INT0;

Г) PCINT1;

Д) PCINT0.

Правильный ответ: Б, В, А, Д, Г

Компетенции (индикаторы): ПК-3 (ПК-3.1)

3. Установите правильную последовательность команд программного кода при формировании управляющего импульса в микропроцессорной системе фазового управления скоростью асинхронного двигателя (**x** – фазовый угол, в десятках мкс).

А) incl (PULSE\_OUT);

Б) udelay (x);

В) udelay(20);

Г) excl (PULSE\_OUT);

Правильный ответ: Б, А, В, Г

Компетенции (индикаторы): ПК-3 (ПК-3.1)

**Задания открытого типа**

**Задания открытого типа на дополнение**

*Напишите пропущенное слово (словосочетание).*

1. Вид релейной защиты, действие которой связано с увеличением силы тока в защищаемой цепи при возникновении короткого замыкания на участке данной цепи, называется \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Правильный ответ: Максимальной токовой защитой (МТЗ)

Компетенции (индикаторы): ПК-3 (ПК-3.1)

2. Максимальная токовая защита с ограниченной зоной действия, имеющая в большинстве случаев мгновенное действие, называется \_\_\_\_\_\_\_\_\_

Правильный ответ: Токовой отсечкой

Компетенции (индикаторы): ПК-3 (ПК-3.1)

3. Один из видов релейной защиты, отличающийся абсолютной селективностью и быстродействием (без искусственной выдержки времени), называется \_\_\_\_\_\_\_\_\_

Правильный ответ: Дифференциальной защитой

Компетенции (индикаторы): ПК-3 (ПК-3.1)

4. Вычислительный процесс, при котором внутри функции происходит вызов этой же функции, называется \_\_\_\_\_\_\_\_\_

Правильный ответ: Рекурсией

Компетенции (индикаторы): ПК-3 (ПК-3.1)

**Задания открытого типа с кратким свободным ответом**

1. Рассчитайте, какой код для регистра ADCSR настраивает аналогово-цифровой преобразователь для работы в циклическом режиме без прерываний и без деления частоты процессора?

Правильный ответ: ADCSR:=$E0; / $E0; / ADCSR:=0xE0;/ 0xE0; / ADCSR:=224; / 224; / ADCSR:=0b11100000; / 0b11100000;

Компетенции (индикаторы): ПК-3 (ПК-3.2)

2. Рассчитайте код для регистра статуса и управления аналогового компаратора ACSR, который разрешает прерывание по падающему фронту выхода компаратора. Биты регистра ACSR, не влияющие на этот режим считать равными «0».

Правильный ответ: ACSR:=0x0A;/ 0x0A; / ACSR=$0А; / $0А; / ACSR=10; / 10; / ACSR=0b00001010; / 0b00001010;

Компетенции (индикаторы): ПК-3 (ПК-3.2)

3. Рассчитайте минимальный коэффициент предделителя АЦП при тактовой частоте процессора f=1 МГц, какая директива или команда реализует запись этого коэффициента в ADMUX?

Правильный ответ: ADCpresc = 8; / ADMUX:=8;

Компетенции (индикаторы): ПК-3 (ПК-3.2)

4. Рассчитайте при тактовой частоте процессора 16 МГц, 9-разрядном ШИМ и коэффициенте предделителя 256, какая будет частота на выходе ОС1 для ATmega328.

Правильный ответ: 16/256/1022=61,15 Гц/ 61,15 Гц/ 61,16 Гц

Компетенции (индикаторы): ПК-3 (ПК-3.2)

**Задания открытого типа с развернутым ответом**

1. Производится ввод уставок релейной защиты в микропроцессорную систему с помощью матричной клавиатуры. Для этого в сегменте нижеприведенного кода программы выполняется определение числа x по нажатой клавише, если запятая была уже нажата ранее. Допишите сегмент кода, который позволяет использование этого оператора case, если была нажатая клавиша **« , » (**использовать логическую переменную, например, p). Дайте пояснения.

loop

 …

 case GetKey of

 Key1:x:=x+1/pow10(k);|

 Key2:x:=x+2/pow10(k);|

 .

 Key9:x:=x+9/pow10(k);|

 Key10:x:=x+0/pow10(k);|

 endcase;

…

endloop;

Время выполнения – 45 мин.

Ожидаемый результат:

Решение:

Логическая переменная p может принимать значения только true и false. Если при вводе числа была нажата запятая, то следует использовать формулу, в которой вводимая цифра будет дописываться в конец числа после запятой:

x=x+N/pow10(k);

Здесь x – текущее и последующее число при вводе цифры; N – вводимая цифра от 0 до 9; pow10(k) – это возведение в степень k числа 10. При вводе цифры после запятой, каждая последующая вводимая цифра смещается на один разряд вправо, т.е. вводимую цифру нужно делить на 10, 100, 1000 и т.д., что соответствует 101, 102, 103 … Таким образом k – это номер разряда после запятой.

Если p=true, то запятая уже была нажата, и можно пользоваться представленным оператором case. Для проверки на нажатие введем оператор if (p=true) then

loop

…

if (p=true) then

case GetKey of

Key1: x:=x+1/pow10(k);|

Key2: x:=x+2/pow10(k);|

.

Key9: x:=x+9/pow10(k);|

Key10: x:=x+0/pow10(k);|

endcase;

enfif;

…

endloop;

Критерии оценивания:

- задание считается выполненным, если в программный код будет введён условный оператор if и даны пояснения, отвечающие смысловому содержанию в ожидаемом результате.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

2. Производится ввод уставок релейной защиты в микропроцессорную систему с помощью матричной клавиатуры. Для этого в сегменте нижеприведенного кода программы выполняется определение числа x по нажатой клавише, если запятая не была нажата. Допишите сегмент кода, который позволяет использование этого оператора case, если счетчик символов i не превышает количество 8 в переменной x (используйте оператор while):

loop

…

case GetKey of

Key1: x:=x\*10+1;|

Key2: x:=x\*10+2;|

…

Key9: x:=x\*10+9;|

Key10: x:=x\*10+0;|

endcase;

…

endloop;

Время выполнения – 45 мин.

Ожидаемый результат:

Решение:

Если при вводе числа не была нажата запятая, то следует использовать формулу, в которой вводимая цифра будет дописываться в конец числа до запятой:

x=x⸱10+N;

Здесь x – текущее и последующее число при вводе цифры; N – вводимая цифра от 0 до 9. При вводе цифры до запятой, каждая последующая вводимая цифра помещается на место в числе, где располагаются единицы, т.е. предыдущее число необходимо умножать на 10 и после этого складывать с вводимой цифрой.

Количество вводимых цифр ограничено переменной i. Это связано с тем, что большое их количество может не поместиться на ЖКИ, а так же с тем, что тип данных float имеет свои ограничения. Для организации ограничения в 8 символов введем оператор while (i<8) do, где i – это счётчик вводимых символов. Как только их станет 8 и более, то ввод заблокируется. Число х перестанет расти.

loop

while (i<8) do

…

case GetKey of

Key1: x:=x\*10+1;|

Key2: x:=x\*10+2;|

…

Key9: x:=x\*10+9;|

Key10: x:=x\*10+0;|

endcase;

endwhile;

…

endloop;

Критерии оценивания:

- задание считается выполненным, если в программный код будет введён цикл while (или его аналог) и даны пояснения, отвечающие смысловому содержанию в ожидаемом результате.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

3. Осуществляется максимально-токовая защита при помощи встроенного в микроконтроллер АЦП. К каким изменениям в нижеприведенном программном коде приведет замена канала оцифровки на 4-й и 7-й канал, измерительного диапазона на диапазон для напряжения U от 10 до 350В, а для тока I от 3 до 18A?

loop

…

Uc:=GetADC(1);

U:=(275/1023)\*float(Uc);

Ic:=GetADC(2);

I:=(50/1023)\*float(Ic);

LCDxy(0,0);

Write(LCDout, 'U=' +floattostr(U:5:1)+ 'B I=' +floattostr(I:4:1)+'A');

…

endloop;

Время выполнения – 45 мин.

Ожидаемый результат:

Решение:

В функции запуска АЦП GetADC(x) аргумент x – это номер канала от 1 до 8. Поэтому в программе заменим каналы 1 и 2 на 4 и 7 соответственно.

Вычисление реального значения U происходит путем составления линейной зависимости U=k⸱Uc+b между цифровым и реальным значением.

Найдем k и b, решив систему уравнений:

350= k⸱1023+b

10= k⸱0+b

Отсюда b=10, а k=(350-10)/1023=340/1023

Аналогично найдем k и b, решив систему уравнений для тока:

18= k⸱1023+b

3= k⸱0+b

Отсюда b=3, а k=(18-3)/1023=15/1023

loop

…

Uc:=GetADC(4);

U:=(340/1023)\*float(Uc)+10;

Ic:=GetADC(7);

I:=(15/1023)\*float(Ic)+3;

LCDxy(0,0);

Write(LCDout, 'U=' +floattostr(U:5:1)+ 'B I=' +floattostr(I:4:1)+'A');

…

endloop;

Строка U:=(340/1023)\*float(Uc)+10; не является оптимальной, это объясняется тем, что на процессор AVR-микроконтроллера возлагаются задачи, которые можно выполнить за пределами программы, а именно деление 340 на 1023. Поэтому оптимально будет так:

U:=0,3324\*float(Uc)+10;

Строка I:=(15/1023)\*float(Ic)+3; так же не является оптимальной – а на процессор AVR-микроконтроллера возлагаются задачи, которые можно выполнить за пределами программы, а именно деление 15 на 1023. Поэтому оптимально будет так:

I:=0,01466\*float(Ic)+3;

Критерии оценивания:

- задание считается выполненным, если в программный код будет заменена хотя бы одна из строк вычисления реальных значений тока и напряжения на U:=(340/1023)\*float(Uc)+10; и(или) I:=(15/1023)\*float(Ic)+3; и осуществлена замена хотя бы одного из каналов АЦП.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

4. Осуществляется синхронизация с сетью микропроцессорной релейной защиты при помощи встроенного в микроконтроллер аналогового компаратора. Какая команда в нижеприведенном коде выполняется перед выполнением команды udelay(t) в блоке acomp. Поясните почему?

{ functions }

interrupt acomp;

begin

udelay(t);

INCL (PortD,7);

udelay(20);

EXCL (PortD,7);

end;

interrupt int1;

begin

if t>0 then

t:=t-1;

else t:=0;

endif;

end;

{ Main Program }

{$IDATA}

begin

ACSR:=$08;

GIMSK:=$C0;

MCUCR:=$0F;

loop

DisableInts;

Repeat Until ON\_OFF=false;

Repeat Until ON\_OFF=true;

EnableInts;

Repeat Until ON\_OFF=false;

Repeat Until ON\_OFF=true;

endloop;

Время выполнения – 45 мин.

Ожидаемый результат:

Решение:

Блок acomp является обработчиком прерывания по аналоговому компаратору. Поскольку происходит синхронизация с сетью, то необходимо отслеживание перехода через 0 и по фронту и по спаду, т.е. переход от отрицательной полуволны к положительной и наоборот. Поэтому в регистр управления и состояния аналогового компаратора ACSR внесён код, позволяющий работу компаратора, и разрешающий прерывание по перепаду сигнала на его входах. Запрос прерывания может возникнуть только при выполнении следующих условий: разрешено глобальное прерывание; разрешено прерывание по изменению сигнала на входах аналогового компаратора и само изменение сигнала.

Разрешение глобального прерывания происходит по команде EnableInts. Разрешение прерывание по изменению сигнала на входах аналогового компаратора выполняется командой ACSR:=$08 до бесконечного цикла loop.

Поэтому прерывания могут возникать при выполнении команд заключенных между EnableInts и DisableInts при переходе синусоиды через 0.

{ functions }

interrupt acomp;

begin

udelay(t);

INCL (PortD,7);

udelay(20);

EXCL (PortD,7);

end;

interrupt int1;

begin

if t>0 then

t:=t-1;

else t:=0;

endif;

end;

{ Main Program }

{$IDATA}

begin

ACSR:=$08;

GIMSK:=$C0;

MCUCR:=$0F;

loop

DisableInts;

Repeat Until ON\_OFF=false;

Repeat Until ON\_OFF=true;

EnableInts;

Repeat Until ON\_OFF=false;

Repeat Until ON\_OFF=true;

endloop;

Такими командами являются циклы Repeat Until ON\_OFF=false и Repeat Until ON\_OFF=true, а так же команда endloop (подчеркнуты).

Критерии оценивания:

- задание считается выполненным, если указана команда и даны пояснения, отвечающие смысловому содержанию в ожидаемом результате.

Компетенции (индикаторы): ПК-3 (ПК-3.3)