EE 2905

Dr. Johnson

No calculator
No notes

Final Exam 2022

Name: _____

1 – Create a Flow Diagram 10pts

2 – Write each of the following numbers using the designated representation you must show your work . 15pts

~~3 – Evaluate each expression individually. operations 15pts~~

4 – Evaluate each expression individually. operations 10pts

~~5 – Evaluate each expression individually. if, switch 10pts~~

6 – Evaluate each expression individually. while, for 10pts

~~7 – Fill in the memory map at the end of the following code 10pts~~

8 – Write a function 10 pts

9 – Timer Elements 10 pts

10 – ADC 10pts

11 – DAC 10pts

12 – PWM 10pts

~~13 – Interrupts 15pts~~

14 – Functions without pointers 10pts

15 – Functions with pointers 10pts

16 – Write the **complete** program using mbed 20 pts

17 – Draw a schematic 15 pts

ASCII TABLE

Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	*[NULL]*
1	1	1	1	*[START OF HEADING]*
2	2	10	2	*[START OF TEXT]*
3	3	11	3	*[END OF TEXT]*
4	4	100	4	*[END OF TRANSMISSION]*
5	5	101	5	*[ENQUIRY]*
6	6	110	6	*[ACKNOWLEDGE]*
7	7	111	7	*[BELL]*
8	8	1000	10	*[BACKSPACE]*
9	9	1001	11	*[HORIZONTAL TAB]*
10	A	1010	12	*[LINE FEED]*
11	B	1011	13	*[VERTICAL TAB]*
12	C	1100	14	*[FORM FEED]*
13	D	1101	15	*[CARRIAGE RETURN]*
14	E	1110	16	*[SHIFT OUT]*
15	F	1111	17	*[SHIFT IN]*
16	10	10000	20	*[DATA LINK ESCAPE]*
17	11	10001	21	*[DEVICE CONTROL 1]*
18	12	10010	22	*[DEVICE CONTROL 2]*
19	13	10011	23	*[DEVICE CONTROL 3]*
20	14	10100	24	*[DEVICE CONTROL 4]*
21	15	10101	25	*[NEGATIVE ACKNOWLEDGE]*
22	16	10110	26	*[SYNCHRONOUS IDLE]*
23	17	10111	27	*[ENG OF TRANS. BLOCK]*
24	18	11000	30	*[CANCEL]*
25	19	11001	31	*[END OF MEDIUM]*
26	1A	11010	32	*[SUBSTITUTE]*
27	1B	11011	33	*[ESCAPE]*
28	1C	11100	34	*[FILE SEPARATOR]*
29	1D	11101	35	*[GROUP SEPARATOR]*
30	1E	11110	36	*[RECORD SEPARATOR]*
31	1F	11111	37	*[UNIT SEPARATOR]*
32	20	100000	40	*[SPACE]*
33	21	100001	41	!
34	22	100010	42	"
35	23	100011	43	#
36	24	100100	44	$
37	25	100101	45	%
38	26	100110	46	&
39	27	100111	47	'
40	28	101000	50	(
41	29	101001	51	)
42	2A	101010	52	*
43	2B	101011	53	+
44	2C	101100	54	,
45	2D	101101	55	-
46	2E	101110	56	.
47	2F	101111	57	/
48	30	110000	60	0
49	31	110001	61	1
50	32	110010	62	2
51	33	110011	63	3
52	34	110100	64	4
53	35	110101	65	5
54	36	110110	66	6
55	37	110111	67	7
56	38	111000	70	8
57	39	111001	71	9
58	3A	111010	72	:
59	3B	111011	73	;
60	3C	111100	74	<
61	3D	111101	75	=
62	3E	111110	76	>
63	3F	111111	77	?
64	40	1000000	100	@
65	41	1000001	101	A
66	42	1000010	102	B
67	43	1000011	103	C
68	44	1000100	104	D
69	45	1000101	105	E
70	46	1000110	106	F
71	47	1000111	107	G
72	48	1001000	110	H
73	49	1001001	111	I
74	4A	1001010	112	J
75	4B	1001011	113	K
76	4C	1001100	114	L
77	4D	1001101	115	M
78	4E	1001110	116	N
79	4F	1001111	117	O
80	50	1010000	120	P
81	51	1010001	121	Q
82	52	1010010	122	R
83	53	1010011	123	S
84	54	1010100	124	T
85	55	1010101	125	U
86	56	1010110	126	V
87	57	1010111	127	W
88	58	1011000	130	X
89	59	1011001	131	Y
90	5A	1011010	132	Z
91	5B	1011011	133	[
92	5C	1011100	134	\
93	5D	1011101	135	]
94	5E	1011110	136	^
95	5F	1011111	137	_
96	60	1100000	140	`
97	61	1100001	141	a
98	62	1100010	142	b
99	63	1100011	143	c
100	64	1100100	144	d
101	65	1100101	145	e
102	66	1100110	146	f
103	67	1100111	147	g
104	68	1101000	150	h
105	69	1101001	151	i
106	6A	1101010	152	j
107	6B	1101011	153	k
108	6C	1101100	154	l
109	6D	1101101	155	m
110	6E	1101110	156	n
111	6F	1101111	157	o
112	70	1110000	160	p
113	71	1110001	161	q
114	72	1110010	162	r
115	73	1110011	163	s
116	74	1110100	164	t
117	75	1110101	165	u
118	76	1110110	166	v
119	77	1110111	167	w
120	78	1111000	170	x
121	79	1111001	171	y
122	7A	1111010	172	z
123	7B	1111011	173	{
124	7C	1111100	174	|
125	7D	1111101	175	}
126	7E	1111110	176	~
127	7F	1111111	177	*[DEL]*

C – Operator Precedence

Precedence	Operator	Description	Associativity
1	++ --	Suffix/postfix increment and decrement	Left-to-right
	()	Function call	
	[]	Array subscripting	
	.	Structure and union member access	
	->	Structure and union member access through pointer	
	(*type*){*list*}	Compound literal(C99)	
2	++ --	Prefix increment and decrement	Right-to-left
	+ -	Unary plus and minus	
	! ~	Logical NOT and bitwise NOT	
	(*type*)	Type cast	
	*	Indirection (dereference)	
	&	Address-of	
	sizeof	Size-of	
	_Alignof	Alignment requirement(C11)	
3	* / %	Multiplication, division, and remainder	Left-to-right
4	+ -	Addition and subtraction	
5	<< >>	Bitwise left shift and right shift	
6	< <=	For relational operators < and ≤ respectively	
	> >=	For relational operators > and ≥ respectively	
7	== !=	For relational = and ≠ respectively	
8	&	Bitwise AND	
9	^	Bitwise XOR (exclusive or)	
10	|	Bitwise OR (inclusive or)	
11	&&	Logical AND	
12	||	Logical OR	
13	?:	Ternary conditional	Right-to-Left
14	=	Simple assignment	
	+= -=	Assignment by sum and difference	
	*= /= %=	Assignment by product, quotient, and remainder	
	<<= >>=	Assignment by bitwise left shift and right shift	
	&= ^= |=	Assignment by bitwise AND, XOR, and OR	
15	,	Comma	Left-to-right

AnalogOut Class Reference

Public Member Functions	
	AnalogOut (PinName pin)
	Create an AnalogOut connected to the specified pin. More...
	AnalogOut (const PinMap &&)=delete
	Create an AnalogOut connected to the specified pin. More...
void	write (float value)
	Set the output voltage, specified as a percentage (float) More...
void	write_u16 (unsigned short value)
	Set the output voltage, represented as an unsigned short in the range [0x0, 0xFFFF]. More...
float	read ()
	Return the current output voltage setting, measured as a percentage (float) More...

Public Member Functions

	AnalogIn (const PinMap &pinmap, float vref=MBED_CONF_TARGET_DEFAULT_ADC_VREF)
	Create an AnalogIn, connected to the specified pin. More...
	AnalogIn (PinName pin, float vref=MBED_CONF_TARGET_DEFAULT_ADC_VREF)
	Create an AnalogIn, connected to the specified pin. More...
float	read ()
	Read the input voltage, represented as a float in the range [0.0, 1.0]. More...
unsigned short	read_u16 ()
	Read the input voltage, represented as an unsigned short in the range [0x0, 0xFFFF]. More...
float	read_voltage ()
	Read the input voltage in volts. More...

Public Member Functions	
	DigitalIn (PinName pin)
	Create a DigitalIn connected to the specified pin. More...
	DigitalIn (PinName pin, PinMode mode)
	Create a DigitalIn connected to the specified pin. More...
	~DigitalIn ()
	Class destructor, deinitialize the pin. More...
int	read ()
	Read the input, represented as 0 or 1 (int) More...

Public Member Functions	
	Timer
	Create a timer object
void	start()
	Start the timer
void	stop()
	Stop the timer
void	reset()
	Reset the timer

Public Member Functions – From Chrono Library	
int	elapsed_time().count()
	Returns the time on the counter in us (note the return type is int and may truncate)
int	chrono::duration_cast<chrono::milliseconds>
	Cast the time on the counter to ms (note the return type is int and may truncate)
int	chrono::duration_cast<chrono::seconds>
	Cast the time on the counter to s (note the return type is int and may truncate)

Ticker & TimeOut

Public Member Functions		
template<typename F >		
MBED_FORCEINLINE void	attach (F &&func, float t)	Deprecated
	Attach a function to be called by the Ticker, specifying the interval in seconds. More...	
void	attach (Callback< void()> func, std::chrono::microseconds t)	Use 10us, 10ms, 10s not 10, 10000, ...
	Attach a function to be called by the Ticker, specifying the interval in microseconds. More...	
void	attach_us (Callback< void()> func, us_timestamp_t t)	Deprecated
	Attach a function to be called by the Ticker, specifying the interval in microseconds. More...	
void	detach ()	
	Detach the function. More...	

void	pulsewidth (float seconds)
	Set the PWM pulsewidth, specified in seconds (float), keeping the period the same. More...
void	pulsewidth_ms (int ms)
	Set the PWM pulsewidth, specified in milliseconds (int), keeping the period the same. More...
void	pulsewidth_us (int us)
	Set the PWM pulsewidth, specified in microseconds (int), keeping the period the same. More...
int	read_pulsewitdth_us ()
	Read the PWM pulsewidth. More...
void	suspend ()
	Suspend PWM operation. More...
void	resume ()

Public Member Functions

	DigitalOut (PinName pin)
	Create a DigitalOut connected to the specified pin. More...
	DigitalOut (PinName pin, int value)
	Create a DigitalOut connected to the specified pin. More...
void	write (int value)
	Set the output, specified as 0 or 1 (int) More...
int	read ()
	Return the output setting, represented as 0 or 1 (int) More...
int	is_connected ()
	Return the output setting, represented as 0 or 1 (int) More...
	copies the state from the DigitalOut argument. More...

Public Member Functions

	SPI (PinName mosi, PinName miso, PinName sclk, PinName ssel=NC)
	Create a SPI master connected to the specified pins. More...
	SPI (PinName mosi, PinName miso, PinName sclk, PinName ssel, use_gpio_ssel_t)
	Create a SPI master connected to the specified pins. More...
	SPI (const spi_pinmap_t &static_pinmap)
	Create a SPI master connected to the specified pins. More...
	SPI (const spi_pinmap_t &static_pinmap, PinName ssel)
	Create a SPI master connected to the specified pins. More...
void	format (int bits, int mode=0)
	Configure the data transmission format. More...
void	frequency (int hz=1000000)
	Set the SPI bus clock frequency. More...
virtual int	write (int value)
	Write to the SPI Slave and return the response. More...
virtual int	write (const char *tx_buffer, int tx_length, char *rx_buffer, int rx_length)
	Write to the SPI Slave and obtain the response. More...
virtual void	lock (void)
	Acquire exclusive access to this SPI bus. More...
virtual void	unlock (void)
	Release exclusive access to this SPI bus. More...

Mode	Polarity	Phase
0	0	0
1	0	1
2	1	0
3	1	1

void	select (void)
	Assert the Slave Select line, acquiring exclusive access to this SPI bus. More...
void	deselect (void)
	Deassert the Slave Select line, releasing exclusive access to this SPI bus. More...
void	set_default_write_value (char data)
	Set default write data. More...
template<typename Type >	
int	transfer (const Type *tx_buffer, int tx_length, Type *rx_buffer, int rx_length, const event_callback_t &callback, int event=SPI_EVENT_COMPLETE)
	Start non-blocking SPI transfer using 8bit buffers. More...
void	abort_transfer ()
	Abort the on-going SPI transfer, and continue with transfers in the queue, if any More...
void	clear_transfer_buffer ()
	Clear the queue of transfers. More...
void	abort_all_transfers ()
	Clear the queue of transfers and abort the on-going transfer. More...
int	set_dma_usage (DMAUsage usage)
	Configure DMA usage suggestion for non-blocking transfers. More...

Public Member Functions

	PwmOut (PinName pin)
	Create a PwmOut connected to the specified pin. More...
	PwmOut (const PinMap &pinmap)
	Create a PwmOut connected to the specified pin. More...
void	write (float value)
	Set the output duty-cycle, specified as a percentage (float) More...
float	read ()
	Return the current output duty-cycle setting, measured as a percentage (float) More...
void	period (float seconds)
	Set the PWM period, specified in seconds (float), keeping the duty cycle the same. More...
void	period_ms (int ms)
	Set the PWM period, specified in milliseconds (int), keeping the duty cycle the same. More...
void	period_us (int us)
	Set the PWM period, specified in microseconds (int), keeping the duty cycle the same. More...
int	read_period_us ()
	Read the PWM period. More...