Part I
Evaluation Page Part A

Name: _____ Signature: _____

1 Evaluation

This part of the exam has 270 questions, with a total of 447 points and 4 bonus points.

Part	Max. Points	Scored Points
A	60	
B	180	
	Total	

Points:	240-220	219-201	200-182	181-162	161-144	143-0
Grade:	A	B	C	D	E	F
Score:						

2 Evaluation Part A

Page	Points	Bonus Points	Score
5	5	0	
6	5	2	
7	8	0	
8	10	0	
9	6	0	
10	8	0	
11	6	0	
12	5	0	
13	5	0	
14	10	0	
15	11	0	
16	9	0	
17	4	0	
18	2	2	
19	8	0	
20	3	0	
21	3	0	
23	5	0	
24	6	0	
26	4	0	
27	4	0	
28	8	0	
29	13	0	
30	13	0	
31	12	0	
32	7	0	
Total:	180	4	

Page	Points	Bonus Points	Score
33	8	0	
34	8	0	
35	8	0	
36	6	0	
37	9	0	
38	14	0	
39	12	0	
40	9	0	
41	9	0	
42	8	0	
43	15	0	
44	17	0	
45	19	0	
46	17	0	
47	18	0	
48	16	0	
49	12	0	
50	11	0	
51	9	0	
52	9	0	
53	8	0	
54	8	0	
55	9	0	
56	8	0	
Total:	267	0	

Part II
Rules

Answer the questions within the space provided. If you do not have enough space, you can use the backside of the sheet. In that case clearly indicate that your answer continues on the backside.

3 Supporting Materials

This is an examination in writing, without the usage of any electronic devices, except a scientific pocket calculator. No restriction on the model of calculator that may be used, but no device with communication capability shall be accepted as a calculator. All other electronic devices are prohibited. Writing paper is available, writing instruments (pencil, pens, etc) have to be organized by the student.

- **Part A**: Without any supporting material, with calculator.
- **Part B**: With a self written summary (format A4, 8 sheets or up to 16 pages), with calculator

4 Procedure

1. Duration:
   Part A: 1 hour = 60 minutes = 60 points.
   (short break)
   Part B: 3 hours = 180 minutes = 180 points.
2. Sign the first page in the provided space. With this you certify that you are only using permitted support material and you are complying to the rules.
3. Write your name on any detached or additional paper sheets. Sheets without a name will not be evaluated.
4. Use the provided paper for your solutions. Use the provided space in the forms and tables. If needed use scratch paper. Document your way to your solution as appropriate.
5. Each question has a defined number of maximum points associated.
6. If a question is unclear, make reasonable assumptions. Document your assumptions and provide a rationale.
7. Write clearly and legibly. Unclear or multiple solutions will not be evaluated.
8. There is a short break between part A and B. You have to sign into a list for a needed break during the examination parts. Only one person can leave the room for a short time.
9. If something is unclear, ask your supervisor in the room.

5 Time Management

Read first all questions. Make sure you distribute your available time to all the questions. To reduce disturbance, ask questions in the first 15 minutes of the exam period.

6 Multiple-Choice Questions

1. Try to answer all questions if possible. If you are not sure, choose the answer which seems the best one.

2. For the questions of type ◯: Choose **exactly one** option with ⊗ (or √), which you think is the best match. With a correct answer you get the given number of points for that question.

3. For the questions of type ±: After a question or possibly incomplete sentence there are four answers or extensions. Evaluate each of them if they are true or false and mark them accordingly with '+' (true) or '-' (false). Independent if the question is formulated grammatically in singular or plural, it is possible that **0, 1, 2, 3, 4** of the choices are true. For three correct answers out of four you receive half of the points.

4. Wrong answers will have no penalty. Each question which has no answer is treated like a wrong answers and will be evaluated with zero points.

5. If you are changing your mind: cross out your old answer and clearly mark which answer is the new one.

May Dilbert be with you! ☺

Question 1 ..... **Points: [1]**

What is a 'recap'?

- ◯ Learning summary with 5 questions.
- ◯ Sumo robot PCB capacitor.
- ◯ Collection of slides.
- ◯ Line sensor capacitance.
- ◯ Tips for students in next semester.

Question 2 ..... **Points: [1]**

What is a VCS?

- ◯ Variable Capacity System.
- ◯ Volatile Control Status.
- ◯ Version Control System.
- ◯ Variable Computer Software.
- ◯ Volatile Client Storage.

Question 3 ..... **Points: [1]**

The processor used on the FRDM board is the following:

- ◯ ARM Cortex-M0+
- ◯ HCS08
- ◯ MMA8780Q
- ◯ ARM Cortex M4
- ◯ MCP4728

Question 4 ..... **Points: [2]**

A directory listing which is under Git control contains the following directories/files:

```
.gitignore
readme.txt
list.txt
src\rotor.c
src\rotor.h
obj\rotor.o
obj\rotor.txt
```

The .gitignore file has following content:

```
/obj
/*.txt
!/r*
```

In above directory listing, ~~strike through~~ the files which are *ignored*.

Question 5 ..... Points: [2]

Explain in a single sentence what each of the following basic VCS actions mean in Git:

(a) Commiting [½]

.....

.....

.....

(b) Reverting [½]

.....

.....

.....

(c) Pushing [½]

.....

.....

.....

(d) Cloning [½]

.....

.....

.....

Question 6 ..... Points: [1]

What is the fundamental difference between SVN and Git?

.....

.....

.....

Question 7 ..... Points: [2]

Explain the difference between the optimistic and pessimistic approach in a VCS. Explain it with an example.

.....

.....

.....

.....

.....

.....

Question 8 ..... 2 Points (Bonus)

This could be your bonus question you have submitted... ☺.

8. _____

Question 9 ..... Points: [2]

(a) Provide an example for a *hard* real-time system: [1]

.....

.....

(b) Provide an example for a *soft* real-time system: [1]

.....

.....

Question 10 ..... Points: [2]

(a) Name some benefits implementing a state machine: [1]

.....

.....

.....

.....

(b) What should be the first steps when implementing a state machine? [1]

.....

.....

.....

.....

Question 11 ..... Points: [5]

In INTRO we implemented an 'Events' driver.

(a) Why did we implement it as an array of bits? [1]

.....

.....

.....

.....

(b) What is the fundamental disadvantage of such an array of bits? [1]

.....

.....

.....

.....

(c) It implements critical section (e.g. to set an event bit) with `EnterCritical()` and `ExitCritical()`. Under which conditions such a critical section would *not* be required? [2]

 Question 11 continuous on the next page... Reached: _____ from 8 points

(d) List reasons why an interrupt service routine *should* use such an Event module: [1]

Question 12 ..... Points: [2]

An RTOS can be either pre-emptive or cooperative: Explain the difference:

Question 13 ..... Points: [4]

(a) In an RTOS, each task can be in one of 5 fundamental states: List them: [2]

(b) What's the purpose of the scheduler in an RTOS? [2]

Question 14 ..... Points: [3]

Provide a short definition of the term *Interrupt Latency*, and which factors/aspects are contributing to it:

Question 15 ..... Points: [2]

Provide an example of a typical *Reactive System*, and explain why this is a reactive system:

Question 16 ..... Points: [4]

(a) A PWM signal on a H-Bridge is labeled as *low active*. Explain what this means and how this impacts the speed of a DC motor: [2]

(b) Draw a timing diagram for that PWM signal: the PWM period is 5 ms, and the motor shall at 20% speed. Indicate how many milliseconds the signal is high and low. [2]

Question 17 ..... Points: [3]

Given the following program:

```
#define ADC_CONFIG (*(volatile uint8_t*)0x123)

static void Interrupt(void) {
  uint8_t i;

  while(ADC_CONFIG & ~0x10);
  for(i=0;i<10;i++) {
    __asm("nop");
  }
}
```

This program is using

- ◯ Interrupt synchronization.
- ◯ Gadfly synchronization.
- ◯ Realtime synchronization.
- ◯ Realtime and Gadfly synchronization.
- ◯ No synchronization.

Question 18 ..... Points: [2]

Your Eclipse project stores the make files, object files and the final (binary) application file in a sub folder inside your project. Are you going to store this folder and files in a version control system? Justify your answer:

.....

.....

.....

.....

.....

.....

.....

Question 19 ..... Points: [3]

Given the source of a PID control loop implementation. Identify in the source lines for the P, I and D part: mark them clearly and label it with P, I and D. Mark/circle this in the following source listing:

```
#define max 0x33ff
static int32_t old=0, b=0;
void PID_Control(void) {
  int32_t f, s, a;

  v = 0;
  f = should-actual;
  a = f-old;
```

```
  old = f;
  v += a/10;
  v += f*35;
  b += f;
  if (b > max) { b = max; }
  v += b/4;
  setAcuator(v);
}
```

Question 20 ..... Points: [3]

Processor Expert components are using the concept of *Methods*, *Properties* and *Events*. What would you expect for an ADC (Analog to Digital Converter) component?

(a) 2 typical *Methods* for an ADC component: [1]

. . . . . .

. . . . . .

. . . . . .

(b) 2 typical *Properties* for of an ADC component: [1]

. . . . . .

. . . . . .

. . . . . .

(c) 2 typical *Events* for an ADC component: [1]

. . . . . .

. . . . . .

. . . . . .

Question 21 ..... Points: [3]

Given following variable definition:

```
static char *string = "hello";
```

What is the difference between the two following usages

```
sizeof(string)
```

```
strlen(string)
```

in respect to the result and the expected code generated?

.....

.....

.....

.....

.....

.....

.....

Question 22 ..... Points: [3]

Given following interface implementation for a DC motor driver:

```
/* motor.h */
#include "LED.h" /* LED interface */
#include "PWM.h" /* PWM interface */

static uint16_t MOT_motorSpeed;

void MOT_Init(void);
/* end of motor.h */
```

Identify three issues with such an interface implementation (issues which could lead to linker/compiler failure, or things which are not considered as good programming style):

.....

.....

.....

.....

.....

.....

Question 23 ..... Points: [2]

Given following program:

```
#define DEC(i) {int b=0; i--;}

void main(void) {
  int a = 5, b = 5;
  DEC(a);
  DEC(b);
  printf("a is %d, b is %d\n", a, b);
}
```

What is the output of `printf()`?

. . . . . .
. . . . . .
. . . . . .
. . . . . .

Question 24 . . . . . Points: [3]

Given the *Mealy Sequential State Machine* in Figure 1 with five states, two input values and two output values:

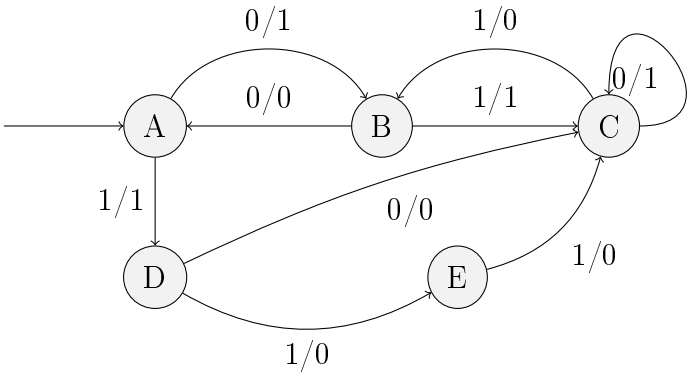

Figure 1: Mealy Sequential State Machine

(a) The state machine in Figure 1 is in the state **'B'**. Determine the *input sequence* in order to generate the following *output*: [2]
1, 1, 1, 0

. . . . . .
. . . . . .

(b) The State Machine in Figure 1 is not complete and has an undefined transition from one state to another: fix this with a solution in Figure 1. [1]

Question 25 . . . . . Points: [4]

(a) Provide an example of a typical *Transforming System*, and explain why this is a Transforming System: [2]

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

(b) Explain why *Optimized Memory Usage* is a typical attribute for a *Transforming System*: [2]

Question 26 ..... Points: [3]

Explain the reason why some processors push all their core registers onto the interrupt stack, and some only push a subset of the registers:

Question 27 ..... Points: [3]

(a) List 2 typical reactive systems: [1]

(b) List 2 typical interactive systems: [1]

(c) List 2 typical transformative systems: [1]

Total: 3

Question 28 ..... Points: [2]

List reasons, why a company would *not* allow any interrupt synchronization methods:

Question 29 ..... Points: [2]

List the things a processor has to do in order to jump to an interrupt service routine and to return from it.

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Question 30 ..... Points: [2]

Explain multiple things which affects the interrupt latency time.

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Question 31 ..... Points: [1]

What does it mean, if somebody says "I have masked the interrupts"?

. . . . . .

. . . . . .

Question 32 ..... Points: [2]

Does the ARM Cortex M0+ support nested interrupts?

32. _____

Question 33 ..... Points: [5]

Answer the questions for following C code, assuming default compiler settings:

```
typedef signed short MyType;
static unsigned char myVar[3];
typedef enum { RED=5, GREEN, YELLOW } Colors;
```

(a) What gives `sizeof(MyType)` for the tinyK20/Remote board/project: [1]

(a) _____

(b) What gives `sizeof(MyType)` for the Robot board/project? [1]

(b) _____

(c) What gives `sizeof(MyVar)` for the tinyK20/Remote board/project? [1]

(c) _____

(d) What gives `sizeof(MyVar)` for the Robot board/project? [1]

(d) _____

 Question 33 continuous on the next page... Reached: _____ from 11 points

(e) Which value has `YELLOW`? [1]

(e) _____

Total: 5

Question 34 ..... Points: [3]

Given following source code:

```
uint16_t abcd[16];
uint8_t buf[10];
static uint16_t values[3];
```

(a) Determine the value of following expression: [1]

```
sizeof("abcd")
```

(a) _____

(b) Determine the value of following expression: [1]

```
sizeof(buf)
```

(b) _____

(c) Determine the value of following expression: [1]

```
sizeof(values)
```

(c) _____

Question 35 ..... Points: [3]

In eclipse you have different ways how you could reference external files within your project structure:

1. Linked Folder
2. Linked Files
3. Virtual Group

List pros and cons for each approach:

.....

.....

.....

.....

.....

.....

Question 36 ..... Points: [2]

A hard realtime system or a soft realtime system: which do you consider easier to implement and test? List one pro and one cons for each:

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Question 37 . . . . . Points: [2]

Consider following source:

```
#define MACRO(a,b) a = j \
  =b
```

Write down the text which would be produced by the preprocessor of the compiler, if you call the `MACRO` as following:

```
MACRO(i,5);
```

. . . . . .

. . . . . .

. . . . . .

Question 38 . . . . . Points: [1]

Given following C source:

```
#define MACRO(var, mask1, mask2) \
  (var = (var & (~(uint8_t)(mask1))) | (uint8_t)(mask2))
static uint8_t var;

void foo(void) {
  var = 0x22;
  MACRO(var, 16, 0x13);
}
```

What is the value of `var` after execution of `foo()`?

38. _____

Question 39 . . . . . Points: [1]

Which sequence is the correct one to configure a keyboard interrupt?

◯ Enable Keyboard Interrupts;
Set Port direction as input;
Enable Pull-Up Resistors;
Acknowledge Pending Interrupt;

 Question 39 continuous on the next page... Reached: _____ from 4 points

◯ Acknowledge Pending Interrupt;
Set Port direction register as input;
Enable Keyboard Interrupts;
Enable Pull-Up Resistors;

◯ Set Port direction register as input;
Enable Pull-Up Resistors;
Acknowledge Pending Interrupt;
Enable Keyboard Interrupts;

◯ Enable Pull-Up Resistors;
Enable Keyboard Interrupts;
Acknowledge Pending Interrupt;
Set Port direction register as input;

Question 40 ..... Points: [1]

For the implementation of a driver for an interrupt hardware following has to be considered:

± Interrupts have to be enabled globally during the driver initialization.

± The driver shall reset the device interrupt flag during initialization.

± After a power-on reset, it might be necessary to wait a certain time until the hardware signals have stabilized.

± The interrupt handler shall be as efficient as possible in order to increase the interrupt latency time.

Question 41 ..... 2 Points (Bonus)

And here could be your recap question. ☺.

Question 42 ..... Points: [1]

Given following program:

```
void main(void) {
unsigned char *src=(unsigned char*)0x100, buffer[0x100], i;
  for(i=0;i<100; i++) {
    buffer[i]=*src;
  }
}
```

For the above program, following applies:

± It reads the values from the address 256 and 512 and stores it in a buffer.

± It reads 100 times the value at the address 0x100 and stores the values one after each other in a buffer.

± At termination of the program, the whole buffer is filled with the values from address 0x100.

± With disabled interrupts, the program behaves in a deterministic way.

Question 43 ..... Points: [7]

Given the Mealy Sequential State Machine in Figure 2.

 Question 43 continuous on the next page... Reached: _____ from 2 points

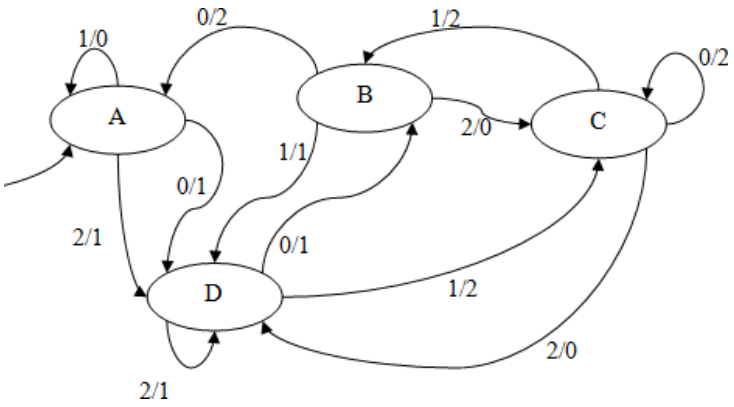

Figure 2: Mealy Machine

(a) The machine in Figure 2 is currently in state 'C'. Determine the output sequence for following input values: 0, 1, 0, 1, 1, 0 [1]

.....

.....

(b) Given following Mealy program: [6]

```
typedef enum {A=0, B, C, D} States;
void Run(void) {
  char j, i = 0

  for(;;) {
    j = Input();
    Output(tbl[i][j][1]);
    i = tbl[i][j][0];
  }
}
```

To implement the machine in Figure 2, complete the initialization of table `tbl`:

```
const char tbl[4][3][2] = {
```

.....

.....

.....

.....

.....

.....

.....

.....

Question 44 ..... Points: [1]

For realtime systems following applies:

± Realtime systems have to have reaction times below 1 ms in order to be realtime compliant.

± For a realtime system not the average system load matters, but the highest possible system load.

± Hard realtime systems are more difficult to verify, because the realtime conditions are not exactly specified.

± A system can be a realtime system, if it is using true random number generator for its decision instead of a pseudo random number generator.

Question 45 ..... Points: [1]

Given following program:

```
char buf[0x100];
int i,j;

static void test(void) {
  for(i=0; i<sizeof(buf); i++) {
    CFG = 0x80; PORTB = 4;
    buf[i] = PORTA;
    PORTB = 0;
  }
}
```

For this program following applies:

± Implements an interrupt synchronization.

± Implements a gadfly synchronization.

± Implements a realtime synchronization.

± None of above.

Question 46 ..... Points: [1]

For all reentrant functions in C, following has to apply:

± The function shall not be recursive.

± The function shall not be called from an ISR.

± The access to shared data has to be protected from mutual access.

± The function shall not modify itself (self modifying code).

Question 47 ..... Points: [1]

The following program gets compiled for the FRDM board with default compiler options:

```
static char ch;  /* Linker places this variable at address 0x10 */
void foo(void) {
  static char i, j=4;
  volatile char v;
  v = i;
  v = j;
  ch = v;
```

 Question 47 continuous on the next page... Reached: _____ from 3 points

```
}
```

Following applies:

- ± The variables i, j and v are allocated on the stack.
- ± The compiler cannot optimize the two assignments to v because of volatile.
- ± At execution time of foo(), the variable v gets initialized with a value of 4.
- ± After execution of foo(), the memory at address 0x10 will have a value of 4.

Question 48 . . . . . Points: [1]

For the interrupt system of the ARM Cortex-M0+/M4 following applies:

- ± The interrupt latency is the sum of execution time of the current instruction, pushing of the registers, calculating the ISR PC address and the branching to the ISR itself.
- ± With 'masking the interrupts' we are enabling the interrupts.
- ± In order for the ISR program to return to the interrupted program, the return address of the interrupted program is stored on the stack by the hardware.
- ± In order to reduce the interrupt latency time, the core can decide not to push all registers on the stack.

Question 49 . . . . . Points: [2]

The diagram in Figure 3 shows an interrupt system with multiple interrupts (IRQ1 and IRQ2) and the corresponding interrupt service routines (ISR) #1 and #2). The lines on the time axis denote the execution time boundaries of the instructions. Following applies:

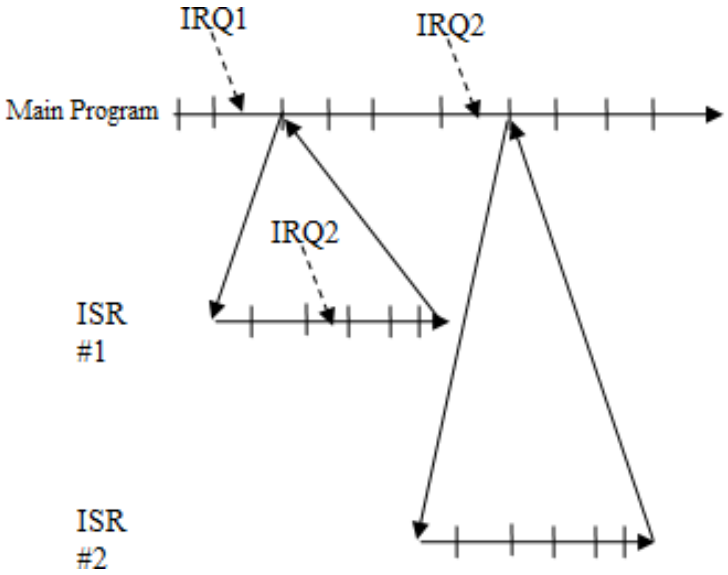

Figure 3: Interrupts

- ◯ At the beginning of ISR #1 all interrupts get disabled, and at the end of ISR #1 the flag for IRQ1 gets acknowledged.
- ◯ The main program has at the beginning all interrupts disabled and has the IRQ1 flag acknowledged. After execution of ISR #1 the main program enables all interrupts.
- ◯ ISR #1 turns off all interrupts at the beginning. At the end of ISR #1 it acknowledged the IRQ1 and IRQ2 flag and enables all interrupts again.
- ◯ At the beginning of ISR #1 the flags for IRQ1 and IRQ2 are acknowledged. All interrupts get disabled at the end of ISR #1.
- ◯ ISR #1 has not acknowledged the IRQ1 flag. ISR #2 acknowledged the flags for IRQ1 and IRQ2 at the beginning of ISR #2.

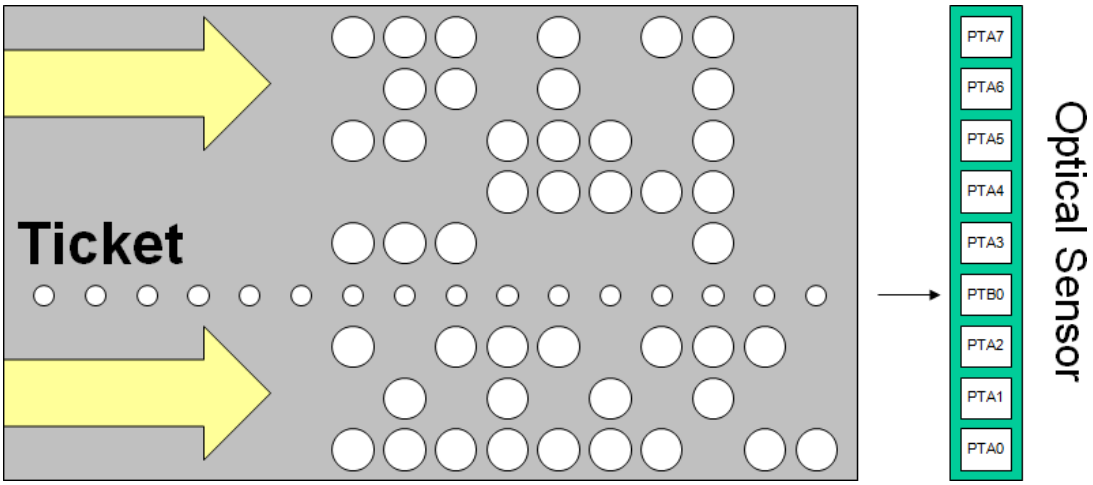

Figure 4: Parking Ticket

Question 50 . . . . . Points: [5]

A punched paper ticket is used in a parking system. The punched paper ticket is using following format for each data line in Figure4:

- 1 guidance bit (small holes)
- 8 data bits (large holes)

The punched paper tape gets pulled into the machine with constant speed of 50 ms for each data line. The data lines are scanned with an optical sensor, and the sensor digital output is attached to the port of a microcontroller. The state of the sensor/holes is available on the microcontroller PORTA, bit 0 to 7:

- Value of bit is 0: no hole, light does not go through
- Value of bit is 1: hole, light goes through

The state of the guidance hole is available on bit 0 of PORTB. The bit 0 of PORTB is configured to raise an keyboard interrupt on falling edge.

Given following program:

 Question 50 continuous on the next page... Reached: _____ from 0 points

```
extern WaitMs(unsigned int ms); /* wait for the given ms */
unsigned char buffer[16]; /* contains the data read */

void Read(void) {
  uint8_t i;

  for(i=0; i<sizeof(buffer); i++) {
    WaitMs(50);
    buffer[i] = PORTA;
  }
}

void KBI_Interrupt(void) {
  /* Guidance Hole Sensor */
  AcknowledgeKBI();
  DisableInterrupts();
  Read();
  EnableInterrupts
}

void main(void) {
  for(;;);
}
```

(a) Which synchronization method is used for the detection of **insertion** of the parking ticket? [1]

- ◯ Combination of interrupt and realtime synchronization.
- ◯ Interrupt synchronization.
- ◯ Realtime synchronization.
- ◯ Combination of gadfly synchronization and realtime synchronization.
- ◯ Gadfly synchronization.

(b) Which synchronization method is used for the synchronization on the **first data hole**? [1]

- ◯ Combination of interrupt and realtime synchronization.
- ◯ Interrupt synchronization.
- ◯ Realtime synchronization.
- ◯ Combination of gadfly synchronization und realtime synchronization.
- ◯ Gadfly synchronization.

(c) Implement a new function `ReadGadfly()` which does the same as `Read()`, but uses a gadfly synchronization method. [3]

```
void ReadGadfly(void) { ...
```

Total: 5

Question 51 ..... Points: [1]

Given following program:

```
double power(double x, int exp) {
  if (exp<=0) return 1;
  return(x*power(x, exp-1));
}
```

Evaluate following:

± In order to have this program reentrant, it is sufficient that `x` and `exp` are variables on a hardware stack.

± It depends on the compiler and the generated code, if this program is reentrant or not.

± The program is reentrant if it is called from an interrupt service routine only.

± The recursive implementation of this program ensures that it is reentrant.

Question 52 ..... Points: [5]

Given a system in Table 1 with programs, priorities and timing:

Program	*Main Priority*	*Sub Priority*	*Time*
HP	0	0	5 ms
UP1	1	1	2 μs
UP2	1	2	3 μs
UP3	2	1	5 μs
UP4	2	2	2 μs

Table 1: Interrupt System

 Question 52 continuous on the next page... Reached: _____ from 6 points

The timing required for a context switch is given in table 2, which is illustrated in Figure 5.

Context Switch	***Time***
Total time for the interrupt, switch to a new program and starting execution of the waiting program	1 μs
Total time for the interrupt, switch to the interrupted program, immediate interruption of this program and switching and starting execution of the waiting program	1 μs

Table 2: Context Switch Timing

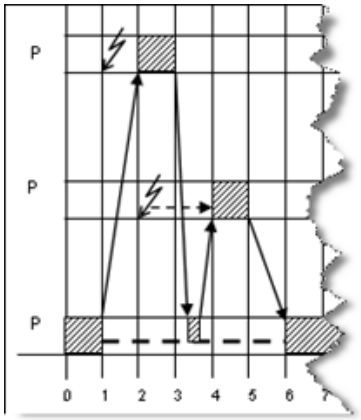

Figure 5: Example Context Switch

The interrupt system is using following rules (as used in the lecture):

$$if(MP(s) <= MP(fn)) \rightarrow \ ws = ws \cup s \tag{1}$$

$$if(MP(s) > MP(fn)) \rightarrow INT(fn) \tag{2}$$

$$if(MP(s) <= MP(fn)) \rightarrow \ ws = ws \cup s \tag{3}$$

$$if(SP(s) > SP(fn)) \rightarrow \ ws = ws \cup s \tag{4}$$

$$if(ws \neq \{\}) \rightarrow fn(MAX(SP(ws))) \tag{5}$$

$$if(MP(ws) > MP(in)) \rightarrow \ fn = in \rightarrow ws = in \tag{6}$$

The programs run according following information:

1. At the time 0 μs HP starts.
2. At the time 2 μs an interrupt for UP1 is raised.
3. At the time 4 μs an interrupt for UP4 is raised
4. At the time 6 μs an interrupt for UP2 is raised
5. At the time 9 μs an interrupt for UP3 is raised
6. At the time 23 μs an interrupt for UP2 is raised

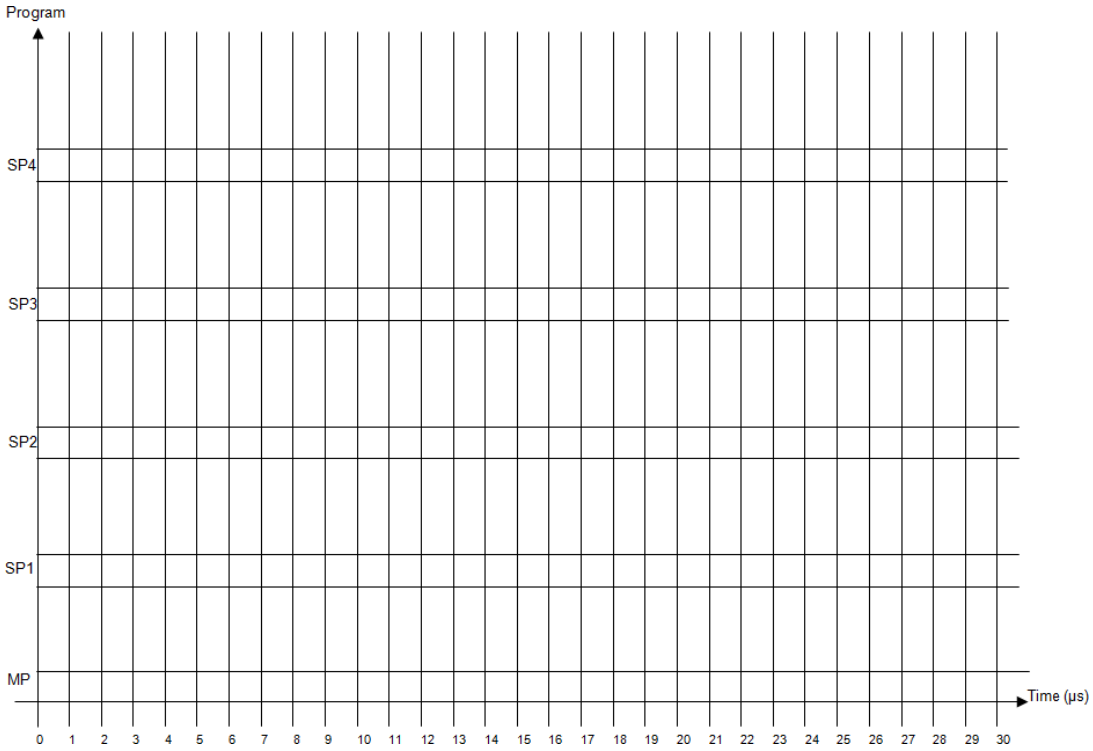

Figure 6: Program Timing

Show the sequence of programs and interrupts in Figure 6. Use the same notation as in Figure 5 for interrupts (Exception, Pending), program switches, program (aktive, suspended).

Question 53 ..... Points: [3]

Consider following implementation:

```
#define FUNC(a,b) i+a+b
int foo(int i, int j) {
  return FUNC(i,j);
}
```

Determine the return value for `foo(5,6);`:

53. _____

Question 54 ..... Points: [3]

Given following source code:

```
uint16_t abcd[8];
uint32_t buf[10];
static uint16_t values[2];
```

(a) Determine the value of following expression: [1]

```
sizeof("abcd")
```

(a) _____

(b) Determine the value of following expression: [1]

```
sizeof(buf)
```

(b) _____

(c) Determine the value of following expression: [1]

```
sizeof(values)
```

(c) _____

Question 55 ..... Points: [2]

Consider following program:

```
void delay(void) {
  uint8_t i;
  for(i=0;i<50;i++);
}
```

This program

- ± always waits for 50 ms
- ± can be optimized by a smart compiler to a function which only contains a `return;` statement
- ± will wait for a certain time which is depending on the speed of the microcontroller used
- ± will never terminate

Question 56 ..... Points: [2]

Consider following Mealy Sequential State Machine with 3 states and two inputs:

```
typedef enum {A, B, C, D, E, } States;
const char tbl[3][2][2] =
{ {{A,0}, {B,1}},
  {{C,3}, {A,4}},
  {{C,0}, {B,5}}
};

void Run(void) {
  char j, i = 0

  for(;;) {
    j = Input();
    Output(tbl[i][j][1]);
    i = tbl[i][j][0];
  }
}
```

 Question 56 continuous on the next page... Reached: _____ from 4 points

(a) Given following sequence of `Input()` values: 0, 1, 0, 1, 1, 1. Determine the sequence of `Output()` values: [1]

(a) _____

(b) Draw the corresponding state diagram: [1]

Total: 2

Question 57 ..... Points: [3]

Consider following program:

```
void main (void) {
  char buf[0x100];
  int i,j;

  PORTB = 0;
  for(i=0; i<sizeof(buf); i++) {
    CFG = 0x80; PORTB = 4;
    while(CFG!=0);
    buf[i] = PORTA;
    PORTB = 0;
  }
}
```

The following applies:

± It implements an interrupt synchronization.

± It implements a Gadfly synchronization.

± It implements a Realtime synchronization.

± It implements no synchronization.

Question 58 ..... Points: [3]

Evaluate following statements about reentrancy:

± A function which modifies its own code is not reentrant.

± A function which calls an interrupt service routine is not reentrant.

± Recursive functions are always reentrant.

± Interrupt service routines are always reentrant if they do not call another routine.

Question 59 ..... **Points: [3]**

If discussing interactive, reactive and transforming systems, then

- ± relative short answer times are typical for interactive systems.
- ± reactive systems are common in systems which do measurement and control.
- ± transforming systems are typically optimized for high throughput.
- ± an example for an transforming system could be a network router.

Question 60 ..... **Points: [3]**

In the context of real time following applies:

- ± Realtime means to produce a result as fast as possible.
- ± A computer is realtime, if is is able to produce at average system load the correct result as fast as possible.
- ± For realtime it is sufficient to have an accurate timing system.
- ± An RTOS is required for a realtime system.

Question 61 ..... **Points: [3]**

For all reentrant functions implemented in C the following applies:

- ± A reentrant function shall not be interrupted.
- ± Interrupt functions does not have to be reentrant, but all functions called from that interrupt routine.
- ± A function which modify itself is reentrant, as long the self modification happens with disable interrupts.
- ± On the ARM Cortex-M0+/M4F the usage of local stack variables does not violate reentrancy.

Question 62 ..... **Points: [2]**

Explain in a few words the reasons why a switch (like a button) needs a resistor. Illustrate it with a small drawing.

Question 63 ..... **Points: [2]**

List important points to be considered for the implementation of an ISR:

Question 64 ..... **Points: [2]**

Explain two different ways how a microcontroller can implement interrupts:

.....

.....

.....

.....

.....

.....

Question 65 ..... **Points: [2]**

What happens if two developers work at the same project but in different files or at different parts in one file and commit on Git?

.....

.....

Question 66 ..... **Points: [2]**

Which is the most important difference between SVN and Git?

.....

.....

Question 67 ..... **Points: [2]**

What happens if two developers work at the same file and at the same part and commit on Git?

.....

.....

Question 68 ..... **Points: [2]**

What is the meaning of the `.gitignore` file?

.....

.....

Question 69 ..... **Points: [1]**

Name one example each for a transforming system, reactive system and interactive system?

.....

.....

Question 70 ..... **Points: [2]**

Why is it necessary to put parenthesis for macros like in the example below?

```
#define CALC1 (2+5)
#define CALC2 (5*3)
```

. . . . . .

. . . . . .

Question 71 . . . . . **Points: [2]**

List advantages and disadvantages using macros:

. . . . . .

. . . . . .

Question 72 . . . . . **Points: [2]**

What needs to be present in a header file to avoid recursive inclusions? Give an example.

. . . . . .

. . . . . .

Question 73 . . . . . **Points: [2]**

How do you declare global variables? In what kind of file?

. . . . . .

. . . . . .

Question 74 . . . . . **Points: [1]**

Is it possible to include other files than `*.h` with `#include`? Can you give an example?

. . . . . .

. . . . . .

Question 75 . . . . . **Points: [1]**

What is the disadvantage of Gadfly synchronization?

. . . . . .

. . . . . .

Question 76 . . . . . **Points: [2]**

How can you prevent that two interrupts access the same data at the same time?

. . . . . .

. . . . . .

Question 77 . . . . . **Points: [2]**

How can you prevent that two interrupts access the same data at the same time?

. . . . . .

. . . . . .

Question 78 ..... **Points: [1]**

What have you to do with unneeded interrupts?

.....

.....

Question 79 ..... **Points: [2]**

What kind of two different events do exist?

.....

.....

Question 80 ..... **Points: [2]**

What is the purpose of a 'sentinel'?

.....

.....

Question 81 ..... **Points: [2]**

What are the advantages and disadvantages of handling events from the main loop?

.....

.....

Question 82 ..... **Points: [1]**

What kind of processor is used on the FRDM-KL25Z?

.....

.....

Question 83 ..... **Points: [1]**

List the two different hardware stack pointer present on an ARM Cortex M:

.....

.....

Question 84 ..... **Points: [1]**

What are the two different categories of timeliness in realtime systems?

.....

.....

Question 85 ..... **Points: [1]**

What kind of external clock is used on the FRDM-KL25Z?

.....

.....

Question 86 ..... **Points: [1]**

List 3 different types of synchronization:

.....

.....

Question 87 ..... **Points: [1]**

How can the logic level of a pin be enforced?

.....

.....

Question 88 ..... **Points: [1]**

What is the purpose of pull-up and pull-down resistors for input pins?

.....

.....

Question 89 ..... **Points: [1]**

List at three different state machine design patterns:

.....

.....

Question 90 ..... **Points: [1]**

List at three different ways how to implement a state machine:

.....

.....

Question 91 ..... **Points: [1]**

What determines the output in a Mealy Sequential State Machine?

.....

.....

Question 92 ..... **Points: [1]**

What is the main advantage of using the *Trigger* module?

.....

.....

Question 93 ..... **Points: [1]**

You are using the *Trigger* module with a 10 ms timer interrupt. Now you want to trigger something in 50 ms. Is this possible?

.....

.....

Question 94 ..... **Points: [1]**

Why is it important to debounce a mechanical switch?

.....

.....

Question 95 ..... **Points: [1]**

List two ways to debounce a mechanical switch:

.....

.....

Question 96 ..... **Points: [1]**

What is the advantage of using USB as a virtual UART serial connection (OpenSDA and USB CDC) over a direct USB connection to a USB port of the MCU ?

.....

.....

Question 97 ..... **Points: [1]**

What is the meaning of an asynchronous serial protocol ?

.....

.....

Question 98 ..... **Points: [1]**

What's the command parser table?

.....

.....

Question 99 ..... **Points: [1]**

What's the difference between Memory Scheme 1 and 2 in FreeRTOS?

.....

.....

Question 100 ..... **Points: [1]**

Provide a good example how FreeRTOS queues can be used between multiple tasks:

.....

.....

Question 101 ..... **Points: [1]**

What is the difference between `xQueueReceive()` and `xQueuePeek()` in FreeRTOS?

.....

.....

Question 102 ..... **Points: [1]**

What are the H-bridges of the motors needed for?

.....

.....

Question 103 ..... **Points: [1]**

Why can the PWM channels not run with different frequencies?

.....

.....

Question 104 ..... **Points: [1]**

On the robot, why can you not use the coast mode of the engines when stopping?

.....

.....

Question 105 ..... **Points: [1]**

What would be good reasons to use industrial SD cards?

.....

.....

Question 106 ..... **Points: [1]**

What can we do to ensure the stored NVMC data are not corrupted?

.....

.....

Question 107 ..... **Points: [1]**

What's the difference between a binary Code and the Grey Code? And what is the advantage of the Grey Code?

.....

.....

Question 108 ..... **Points: [1]**

What are the three possibilities to do the data acquisition?

.....

.....

Question 109 ..... **Points: [1]**

Why do we use the sampling method and not the interrupt method for our robot?

.....

.....

Question 110 ..... **Points: [1]**

What happens with the system if our robot drives to fast to handle the encoder signals? Compare the interrupt vs. the sampling methods.

.....

.....

Question 111 ..... **Points: [1]**

What are the input and output signals of `QuadCounter.c`?

.....

.....

Question 112 . . . . . Points: [1]
Why is there a NULL pointer at the end of the `CmdParserTable`-Array?

. . . . . .

. . . . . .

Question 113 . . . . . Points: [1]
In the shell the command line parser compares strings with `sizeof(“cmpString”)-1`. Explain why this -1 is necessary:

. . . . . .

Question 114 . . . . . Points: [1]
Which settings do you have to configure for the shell communication between computer and device?

. . . . . .

Question 115 . . . . . Points: [1]
Which ANSI-C keyword can you use to prevent loop optimization (and others) in the compiler?

. . . . . .

Question 116 . . . . . Points: [1]
Why is it necessary to use synchronization between two systems?

. . . . . .

Question 117 . . . . . Points: [1]
Why do have functions which are called both from an interrupt and the main program to be reentrant?

. . . . . .

Question 118 . . . . . Points: [1]
Does every microcontroller implement nested interrupts?

. . . . . .

Question 119 . . . . . Points: [1]
Which three interrupts have predefined interrupt priorities on the ARM Cortex-M0+ and cannot be changed?

. . . . . .

Question 120 . . . . . Points: [1]
On the ARM Cortex-M0+, can a HardFault be interrupted by another interrupt?

.....

Question 121 ..... **Points: [2]**

Using a quadrature counter: Provide guidelines when you would use the delta-time and when the delta-pos approach:

.....

.....

Question 122 ..... **Points: [2]**

Explain the advantage of using a ring buffer with quadrature steps for estimating the speed:

.....

.....

Question 123 ..... **Points: [3]**

Why is it necessary to use an anti-windup for a PID?

.....

.....

.....

.....

Question 124 ..... **Points: [1]**

You measure the maximum speed of your quadrature encoder signal, and you measure a quadrature step every 100 μs. Determine the sampling period needed:

.....

Question 125 ..... **Points: [1]**

In your robot application, from where do you call the PID control loop?

.....

Question 126 ..... **Points: [2]**

Why is it not possible to directly measure the output signal of the optical quadrature encoder we have used?

.....

.....

Question 127 ..... **Points: [3]**

Discuss the pros and cons of using either sampling or interrupt method for a quadrature signal:

. . . . . .

. . . . . .

. . . . . .

Question 128 . . . . . Points: [2]

What fundamental problem exists for absolute position encoders, and how can it be solved:

. . . . . .

. . . . . .

Question 129 . . . . . Points: [2]

Can you list the main features of the MCP4728?

. . . . . .

. . . . . .

Question 130 . . . . . Points: [2]

What are the special things or attributes of the Gray code?

. . . . . .

. . . . . .

Question 131 . . . . . Points: [2]

If the robot moves with a speed of 1 m/s, and you measure the reflectance sensor with 100 Hz, what would be an estimated way distance over the white line until you detect the white sumo line in the application?

. . . . . .

. . . . . .

Question 132 . . . . . Points: [2]

The reflectance sensor has two red LEDs to indicate if the sensor is on. For the red LEDs there is a 1K Ohm resistor in series to limit the current through the LED. But why is there another 220 Ohm resistor in series to the LED with that 1K Ohm resistor?

. . . . . .

. . . . . .

Question 133 . . . . . Points: [2]

What is the advantage of the capacitive discharge circuit used for the reflectance sensor?

. . . . . .

. . . . . .

Question 134 ..... **Points: [1]**

List three typical requirements for an RTOS:

.....

.....

Question 135 ..... **Points: [1]**

List three reasons why to use an RTOS:

.....

.....

Question 136 ..... **Points: [1]**

What is the difference between preemptive and non-preemptive scheduling?

.....

.....

Question 137 ..... **Points: [1]**

What is the advantage of scheduling with an RTOS?

.....

.....

Question 138 ..... **Points: [1]**

What is the difference between an RTOS and a normal OS?

.....

.....

Question 139 ..... **Points: [1]**

List three states of the debounce state machine we have used:

.....

.....

Question 140 ..... **Points: [1]**

List two different solutions to debounce a push button:

.....

.....

Question 141 ..... **Points: [1]**

Explain why debouncing is necessary:

.....

.....

Question 142 ..... **Points: [1]**

Explain briefly how to add support for inter-clicks (press one button, then add another

button, then release one of the buttons) in the debouncing state machine we have used:

.....

.....

Question 143 ..... **Points: [3]**

Explain the principle of 'fast decay' and 'slow decay' motor stopping for a full H-Bridge:

.....

.....

Question 144 ..... **Points: [1]**

You are using a 100% duty cycle PWN. Does this mean your motor is at maximum speed?

.....

.....

Question 145 ..... **Points: [1]**

While driving your H-Bridge with a PWM to drive a DC motor, you hear an audible noise. What could you do to fix the problem?

.....

.....

Question 146 ..... **Points: [1]**

What is the difference between a full and a half-H Bridge?

.....

.....

Question 147 ..... **Points: [1]**

You decide to use a 'common' folder for the INTRO project. Which files do you place into that folder?

.....

.....

Question 148 ..... **Points: [1]**

You decide *not* to use a 'common' folder for the INTRO project. What does this mean for your project?

.....

.....

Question 149 ..... **Points: [1]**

Name three reasons why a project should be carefully structured with folders?

. . . . . .

. . . . . .

Question 150 . . . . . Points: [1]

How can you direct the compiler settings to go up one directory in the folder structure in Eclipse?

. . . . . .

. . . . . .

Question 151 . . . . . Points: [1]

Why should you use relative paths in your project and not absolute path settings?

. . . . . .

. . . . . .

Question 152 . . . . . Points: [1]

List a disadvantage of using macros:

. . . . . .

. . . . . .

Question 153 . . . . . Points: [1]

What's the difference between using ".." or <..> for includes?

. . . . . .

. . . . . .

Question 154 . . . . . Points: [1]

Explain the reason why microcontroller pins have names like TSI0_CH11/PTB18-/TPM2_CH0:

. . . . . .

. . . . . .

Question 155 . . . . . Points: [1]

If using pin muxing, what do you have to consider?

. . . . . .

. . . . . .

Question 156 . . . . . Points: [2]

Why is using a function like `CLS1_SendString()` better than using `printf()`?

. . . . . .

. . . . . .

Question 157 ..... **Points: [1]**

Why needs a string with 5 characters 6 bytes in memory, and not 5?

.....

.....

Question 158 ..... **Points: [1]**

Inside your timer interrupt service routine you are using

```
static int counter = 0;
```

Now you remove the `static`. What is the effect?

.....

.....

Question 159 ..... **Points: [2]**

You consider to handle the event bits set from the main loop, instead of using a check/clear in several places in your application. Discuss the pros and cons of this approach:

.....

.....

Question 160 ..... **Points: [3]**

Can you give reasons why the KL25Z128 bus clock is limited to 24 MHz, while the K22FX512 can run a bus clock of 60 MHz?

.....

.....

Question 161 ..... **Points: [2]**

What are the pros and cons of using an external clock vs. internal clock?

.....

.....

.....

Question 162 ..... **Points: [3]**

Briefly explain the purpose of CPU clock, Bus Clock and System Clock:

.....

.....

Question 163 ..... **Points: [3]**

What are problems and difficulties with using a normal delay (busy waiting) loop if using different platforms?

. . . . . .

. . . . . .

Question 164 . . . . . Points: [3]

Can you provide reasons why using an interrupt synchronization might not be the best option?

. . . . . .

. . . . . .

Question 165 . . . . . Points: [2]

Can you give examples for where synchronization is necessary?

. . . . . .

. . . . . .

Question 166 . . . . . Points: [2]

What is the HardFault component doing?

. . . . . .

. . . . . .

Question 167 . . . . . Points: [2]

What do you have to consider in the Events module when you clear/set event bits?

. . . . . .

. . . . . .

Question 168 . . . . . Points: [2]

How can you quickly check the interrupt level of your interrupts of your application?

. . . . . .

. . . . . .

Question 169 . . . . . Points: [3]

Explain the principle of the Event handling in the Event module:

. . . . . .

. . . . . .

Question 170 . . . . . Points: [3]

What are Thumb and Thumb-2 instructions, and what is special about them?

. . . . . .

. . . . . .

Question 171 ..... **Points: [3]**

List advantages of using an I/O structure:

.....

.....

Question 172 ..... **Points: [3]**

List reasons why you should use the Utility component methods:

.....

.....

Question 173 ..... **Points: [3]**

Which possibilities do you have on the FRDM board to communicate to a COM port on the computer?

.....

.....

Question 174 ..... **Points: [3]**

For what are the callbacks used in the Trigger data structure?

.....

.....

Question 175 ..... **Points: [2]**

What is the purpose of `TRG_NOF_TRIGGERS` and why does it have to be at the end of the list?

.....

.....

Question 176 ..... **Points: [1]**

How many FreeRTOS tasks can be in the RUNNING state?

.....

.....

Question 177 ..... **Points: [2]**

List differences between a standard and an embedded OS:

.....

.....

Question 178 ..... **Points: [2]**

If you multiple push buttons in your system, do you need to debounce all of them?

.....

.....

Question 179 ..... **Points: [2]**

List ways how to debounce push buttons:

.....

.....

Question 180 ..... **Points: [3]**

List the scheduling policies of FreeRTOS. What are the differences?

.....

.....

Question 181 ..... **Points: [2]**

What's the difference between `vTaskDelay()` and `vTaskDelayUntil()`?

.....

.....

Question 182 ..... **Points: [3]**

What happens if the task runs longer than T in `vTaskDelayUntil(T)`?

.....

.....

Question 183 ..... **Points: [2]**

Whats the difference in units between the 'total heap size' and the 'minimal stack size' in FreeRTOS?

.....

.....

Question 184 ..... **Points: [2]**

What is the purpose of 'max syscall interrupt priority'?

.....

.....

Question 185 ..... **Points: [2]**

List two different FreeRTOS memory schemes out of five:

.....

.....

Question 186 ..... **Points: [1]**

What does IPC stand for?

.....

.....

Question 187 ..... **Points: [2]**

List one disadvantages of priority ceiling vs. priority inheritance protocol:

.....

.....

Question 188 ..... **Points: [4]**

Provide an example how FreeRTOS semaphores can be used for IPC:

.....

.....

Question 189 ..... **Points: [2]**

How does the RTOS ensure that using the semaphore/mutex API is reentrant for interrupts?

.....

.....

Question 190 ..... **Points: [2]**

Is it necessary to use a xSemaphoreGive() after xSemaphoreTake()?

.....

.....

Question 191 ..... **Points: [2]**

What is the fundamental difference between “VM” and “NVM”?

.....

.....

Question 192 ..... **Points: [2]**

Why is it not possible e.g. to write only 1023 bytes in FLASH?

.....

.....

Question 193 ..... **Points: [2]**

What is the value of a 16bit memory location in FLASH if it is erased?

.....

.....

Question 194 ..... **Points: [2]**

Why should flash memory not erased and programmed very frequently?

.....

.....

Question 195 ..... **Points: [2]**

What are the advantages of industrial SD card memory devices?

.....

.....

Question 196 ..... **Points: [2]**

What is the key feature of the Gray code compared to the normal binary code?

.....

.....

Question 197 ..... **Points: [2]**

How can you convert a binary code to a Gray code?

.....

.....

Question 198 ..... **Points: [2]**

Why did we have to do a calibration for the quadrature signal?

.....

.....

Question 199 ..... **Points: [2]**

Given a quadrature encoder disk with 100 holes and 6000 revolutions per minute. Determine the needed sampling frequency:

.....

.....

Question 200 ..... **Points: [2]**

What is a fast and efficient way to decode a quadrature signal?

.....

.....

Question 201 ..... **Points: [2]**

What is the problem with the quantization and the change of bits?

.....

.....

Question 202 ..... **Points: [2]**

Which two methods do you have to estimate the speed, and which one is better?

.....

.....

Question 203 ..... Points: [3]

Discuss the pros and cons using a ring buffer with sampled positions to estimate the speed:

.....

.....

Question 204 ..... Points: [2]

When choosing the PWM frequency for the H-Bridge, what do you have to consider?

.....

.....

Question 205 ..... Points: [2]

In `MOT_Deinit()`, what should you do?

.....

.....

Question 206 ..... Points: [1]

In our motor driver, what happens if the duty is set higher than 100%?

.....

.....

Question 207 ..... Points: [1]

In our motor driver, what happens if the duty is set with a negative value?

.....

.....

Question 208 ..... Points: [1]

While driving the robot fast forward and backward, you might loose the Bluetooth communication. How can you fix this?

.....

.....

Question 209 ..... Points: [1]

In C, how can you make sure a variable is always created when you enter a function?

.....

.....

Question 210 ..... Points: [1]

How can you increase your confidence that your implementation is reentrant?

.....

.....

Question 211 ..... **Points: [1]**

Discuss 'self modifying code' for embedded systems:

.....

.....

Question 212 ..... **Points: [1]**

Which problems can happen if for a critical section the interrupts get disabled?

.....

.....

Question 213 ..... **Points: [2]**

What is the advantage of using sub-priorities in an interrupt system?

.....

.....

Question 214 ..... **Points: [2]**

Which points you have to consider and check if using a LED with a microcontroller pin?

.....

.....

Question 215 ..... **Points: [1]**

A LED is connected on the cathode side to the microcontroller 0-3.3V pin. Which voltage will you apply to the pin to turn the LED on?

.....

Question 216 ..... **Points: [1]**

Specify the four methods which should always be implemented for a hardware driver?

.....

Question 217 ..... **Points: [1]**

Why should a driver deinitialized with Deinit() if it is not used any more?

.....

Question 218 ..... **Points: [1]**

How to protect calculations with macros?

.....

Question 219 ..... **Points: [1]**

How do you protect header files from re-inclusions?

.....

Question 220 ..... **Points: [1]**

What is an implicit declaration warning?

.....

Question 221 ..... **Points: [1]**

What does the keyword volatile?

.....

Question 222 ..... **Points: [1]**

Disadvantages of active waiting?

.....

Question 223 ..... **Points: [1]**

What synchronization possibilities do we use?

.....

Question 224 ..... **Points: [1]**

What does the abbreviation **RTOS** mean?

.....

Question 225 ..... **Points: [1]**

List one advantage and one disadvantage using an RTOS:

.....

Question 226 ..... **Points: [1]**

What is *Preemptive Scheduling*?

.....

Question 227 ..... **Points: [1]**

Provide one method a task can use to request a context switch:

.....

Question 228 ..... **Points: [1]**

How can you ensure in your task that the task never ends?

.....

Question 229 . . . . . **Points: [1]**

List the four conditions for a realtime program:

. . . . . .

Question 230 . . . . . **Points: [1]**

What is the main difference between hard and soft realtime?

. . . . . .

Question 231 . . . . . **Points: [1]**

List four key elements of the Processor Expert concept:

. . . . . .

Question 232 . . . . . **Points: [1]**

What is the difference between absolute and relative timeliness?

. . . . . .

Question 233 . . . . . **Points: [1]**

Your application has do to a lot of integer divisions in a short amount of time. Why would an ARM Cortex-M0+ not be an ideal choice for this?

. . . . . .

Question 234 . . . . . **Points: [1]**

Is it possible to disable the reset interrupt on ARM Cortex-M with PRIMASK=1? Why?

. . . . . .

Question 235 . . . . . **Points: [1]**

List a design patterns which typically use shared data with typical data access routines:

. . . . . .

Question 236 . . . . . **Points: [1]**

In your complex project using interrupts you get sporadic faults. List possible problems:

. . . . . .

Question 237 . . . . . **Points: [1]**

Provide two good design rules for writing an interrupt service routine:

. . . . . .

Question 238 ..... Points: [1]

List three different design patterns to implement a state machine:

.....

Question 239 ..... Points: [1]

On the INTRO V2 robot you are getting sporadic interrupts for the push button, but the button has not been pressed. What is the most likely reason for this?

.....

Question 240 ..... Points: [1]

You are using polling in the INTRO project to check the status of the push button on the robot. This works fine at the beginning of the semester, but later in the project not any more and push button presses are not always detected. What could be the problem?

.....

Question 241 ..... Points: [1]

How can you find out on a linux system which port/device you can use for communication with the robot?

.....

Question 242 ..... Points: [1]

Why should you not use printf() e.g. for printing some strings?

.....

Question 243 ..... Points: [1]

Which three types of information are stored in a Trigger `TriggerDesc` structure?

.....

Question 244 ..... Points: [1]

You have implemented a debouncing for a push button using an interrupt pin. While testing it, you realize the the button handling is working just one time after system startup, but not afterwards any more. Provide a likely reason for this:

.....

Question 245 ..... Points: [1]

You have installed a thread awareness debugging plugin in Eclipse. You run the FreeRTOS application, but while running you don't see any thread information. Why is that?

.....

Question 246 . . . . . Points: [1]

You have configured the SEGGER Systemview library in your application to use the *post mortem mode*. When connecting to the running target with the SystemView Viewer on the host, you don't see any data. Why?

. . . . . .

Question 247 . . . . . Points: [1]

In a FreeRTOS system running on an ARM Cortex-M4 you have configMAX_SYSCALL_INTERRUPT_PRIORITY set to 5. List the interrupt priorities which are still enabled after calling portDISABLE_INTERRUPTS():

. . . . . .

Question 248 . . . . . Points: [1]

In a FreeRTOS system running on an ARM Cortex-M0+ you have configMAX_SYSCALL_INTERRUPT_PRIORITY set to 5. List the interrupt priorities which are still enabled after calling portDISABLE_INTERRUPTS():

. . . . . .

Question 249 . . . . . Points: [1]

Which kind of FreeRTOS API calls are allowed to be called from an ISR?

. . . . . .

Question 250 . . . . . Points: [1]

Using FreeRTOS queues, can you change the size of a queue?

. . . . . .

Question 251 . . . . . Points: [1]

What are FreeRTOS hooks? List three example hooks:

. . . . . .

Question 252 . . . . . Points: [1]

What is the difference between CLS1_ParseWithCommandTable() and CLS1_ReadAndParseWithCommandTable()?

. . . . . .

Question 253 . . . . . Points: [1]

What is the difference between UTIL1_strcmp() and UTIL1_strncmp()?

. . . . . .

Question 254 ..... **Points: [1]**

What did we need to provide a 48 MHz clock to the USB module on the processors we used in INTRO?

.....

Question 255 ..... **Points: [1]**

What is the differential voltage between the DP (Data Plus) and DM (Data Minus) wires in USB?

.....

Question 256 ..... **Points: [1]**

List several possible reasons why in your INTRO project the USB CDC port does not get enumerated on the host:

.....

Question 257 ..... **Points: [1]**

Can you change the lenght of a FreeRTOS queue at run time?

.....

Question 258 ..... **Points: [1]**

In FreeRTOS queues, are queue items added/removed by reference or by value?

.....

Question 259 ..... **Points: [1]**

Explain in a few words the problem of *Priority Inversion*:

.....

Question 260 ..... **Points: [1]**

What is the difference between a semaphore and a mutex:

.....

Question 261 ..... **Points: [1]**

What the fundamental difference between the concept of 'Semaphore' and 'Critical Section'?

.....

Question 262 ..... **Points: [1]**

The refelctance sensor on your robot only shows 0xFFFF as raw values. Provide the most likely reason for this:

.....

Question 263 . . . . . Points: [1]

Your reflectance sensor reports rather low values for 'black'. How can you fix this?

. . . . . .

Question 264 . . . . . Points: [1]

In FreeRTOS, what is the difference in the protocol used for Semaphore and Mutex?

. . . . . .

Question 265 . . . . . Points: [1]

On some microcontrollers, why is it necessary to run the programming algorithm in RAM?

. . . . . .

Question 266 . . . . . Points: [1]

If during FLASH programming the power supply goes off, what could be the result?

. . . . . .

Question 267 . . . . . Points: [1]

Can you provide some guidance for setting the PWM frequency for a given motor full H-Bridge?

. . . . . .

Question 268 . . . . . Points: [1]

In the MOT_MotorDevice device structure for the motors we did not had variable for the motor direction. How is the current direction (forwared/backward) stored?

. . . . . .

Question 269 . . . . . Points: [1]

You are using a single hall sensor with a single magnet on your robot wheel as a position encoder. You can get the speed with this setup, but what information will you not get?

. . . . . .

Question 270 . . . . . Points: [1]

With your robot you are driving with a constant speed. The problem is that you get about the same number of 0, 1, -1 and ERR steps from your quadrature decoder. What could be the problem?

. . . . . .