# microcontroller lab, 1st measurement check questions

### 1. What types of objects can an integrated parts library contain in Altium Designer?

Wiring diagram illustration, PCB drawing (footprint), 3D model, simulation model.

### 2. What is the difference between a component and a shore? How do they relate to each other?

The part is a separate part of the component, which can be handled separately in the circuit diagram, but in the PCB drawing the one

the shores belonging to the part are already displayed together.

### 3. What is the powder for?

For better visibility, instead of long wires, ports can be added to the parts to be connected,

and used to connect different pages. Ports with the same name are connected.

### 4. What is a bus and when should it be used?

Multiple signal wires joined together, many wires running side by side should be joined together to make the figure more transparent.

### 5. What simulation options does the Altium Designer program offer? List at least three.

Operating point analysis (DC), Frequency transmission (Bode), Time domain analysis (Transient), Temperature, Noise.

### 6. How can we check the wiring diagram in Altium Designer? How to set this up?

Depending on the settings, you will automatically receive error messages during compilation. For example, you can set it to 2 outputs

pin connection should generate an errort, while an input and brain IO pin connection should be accepted.

### 7. Why digital equipment used in industrial environments requires special inputs

### receiving circuits?

Due to interference, the sensors use a higher voltage, so an adapter circuit is required.

As well as surge protection.

### 8. What is the role of the Zener diode in the 24V input receiving circuit examined during the measurement?

The Zener diode on the input is designed to withstand the reverse polarity and overvoltage connected to the input.

provides protection.

### 9. Why use a hysteresis comparator in the input receiving circuit?

Due to the noisy input, the output of the smooth comparator would be hazy.

### 10. What causes the low-pass behavior of the input stage?

The capacity of the stage.

# Microcontroller lab, 2nd measurement check questions

### 1. What layers do we use for routing?

Top Layer and Bottom Layer, possibly additional inner layers.

### 2. What do we use vias for?

It is a small diameter metallized hole with which an electrical contact is made between the layers.

### 3. What is the Solder Mask layer for?

Includes solder mask. Where there is a shape on it, there is no layer of lacquer.

### 4. What is a polygon pour?

This means vibrating the surface that covers a certain area. Used to connect GND.

### 5. What is a footprint?

The drawing and imprint of the parts on the wiring.

### 6. What are the design rules?

It is determined by the technology available from the manufacturer, such as the minimum wire thickness.

### 7. What is on the Overlay (Silkscreen) layers?

On this layer are the screened inscriptions.

### 8. How is an SMD resistor implanted?

Applying solder paste, implanting parts, refilling.

### 9. What is Clearance?

Minimum distance between a spill and a different wire.

### 10. What justifies the size difference between the Solder Mask and Top / Bottom layers on the pads?

To prevent accidental contact of solder with the solder surface.

# Microcontroller lab, 3rd measurement check questions

### What is WDT good for?

WDT is a security monitoring tool that is typically a gender of your program system

protects the application against sticking in a desirable endless loop. When activated, ensure that

if the WDT counter is not re-initialized at a given time, it reaches zero

counter resets the processor by hardware.

What is a priority crossbar decoder good for? How the pin assignment on ports P0-P3 changes if a

the following peripherals are allowed in the crossbar: UART0, SPI, INT0?

The essence of this is that a switchboard determines which peripherals (UART, SPI, I2C, Timer,

ADC, etc ....) which port should be assigned to pins.

UART0 TX0 P0.0

RX0 P0.1

SPI0

SCK P0.2

MISO P0.3

MOSI P0.4

NSS P0.5

INT0 P0.6

### What clock sources can the C8051F040 microcontroller have?

Internal programmable oscillator up to 25 MHz or external oscillator (Crystal, RC, C, Clock).

What are the IT priority levels of the controller?

2 priority levels: high or low. Default: low

What is the difference between MOV, MOVC and MOVX?

MOV: Normal operating command

MOVC: Copy code byte

MOVX: external data

Pop-up:

- Watch dog

- MOV, MOVC, MOVX

SZN

# Microcontroller lab, 4th measurement check questions

### - What is Analog-to-Digital Converter (ADC)? What will we use for the measurement?

The ADC is a converter unit that uses an analog signal (in this case the output voltage of the potentiometer)

produces the digital equivalent.

### - How can our program know that the ADC has completed a conversion, and the new result is available to us?

For example, you can configure the ADC to trigger an interrupt after each conversion.

### - If we create PWM output using software using an 8-bit counter, what is the fill factor

### do the following thresholds result: 0, 64, 128, 192, 255?

The output for the LED is exactly 1 if counter> limit.

0: 100% 64: 75% 128: 50% 192: 25% 255: 0%

### - What is the function of the Input Enable signal on the LCD display?

It is a kind of clock (or strobe) signal. Before setting the desired input, pull it down to 0, set the input (LCD\_DB),

then raise to 1, which causes the display to execute.

### - What appears on a PWM output when we stop the microcontroller using a 50% fill factor

using a debugger?

When the microcontroller stops, the PWM output is stuck at 0 or 1.

### - Why should the values ​​to be displayed be displayed continuously on the 7-segment display? For the LCD display

### why is this not necessary?

The displayed characters are stored in the LCD display, so there is no need to update them continuously, the 7-segment display does not store them.

### - What is the role of the priority crossbar decoder?

To prevent unused peripherals from taking up unnecessary footprint, the priority crossbar decoder should be turned on.

distributes the legs of the first 4 ports (0-3) to the peripherals in a row, and the remainder remains as a general-purpose I / O leg.

### - In a finished circuit, why pay close attention when a person turns on a new peripheral in the microcontroller?

The new peripheral should not be attached to an existing foot.

### - How can we check the value of a variable when the program runs to a specific line?

We add a breakpoint to the line. When the program stops at the breakpoint, the variable i (as text in the source code)

you can select and right-click to add the variable i to the watch window (Add “i” to watch window> # 1).

Pop-up:

### - Why should the values ​​to be displayed be displayed continuously on the 7-segment display? For the LCD display, do this

### why is there no need?

### - What's the mistake?

char \* str;

sprintf (str, “something”) Answer: There is no reservation for str

# Microcontroller lab, 5th measurement check questions

###  What clock source does the controller use after a reset?

After a reset, the controller operates from the internal 16MHz RC oscillator (HSI).

###  What is the maximum system clock of the controller?

168MHz (using external quartz oscillator and PLLs properly)

###  What modes does a GPIO port have?

o GPIO\_MODE\_INPUT // Input Floating Mode

o GPIO\_MODE\_OUTPUT\_PP // Output Push Pull Mode

o GPIO\_MODE\_OUTPUT\_OD // Output Open Drain Mode

o GPIO\_MODE\_AF\_PP // Alternate Function Push Pull Mode

o GPIO\_MODE\_AF\_OD // Alternate Function Open Drain Mode

o GPIO\_MODE\_ANALOG // Analog Mode

o GPIO\_MODE\_IT\_RISING // External Interrupt Mode with Rising edge trigger detection

o GPIO\_MODE\_IT\_FALLING // External Interrupt Mode with Falling edge trigger detection

o GPIO\_MODE\_IT\_RISING\_FALLING // External Interrupt Mode with Rising / Falling edge trigger detection

o GPIO\_MODE\_EVT\_RISING // External Event Mode with Rising edge trigger detection

o GPIO\_MODE\_EVT\_FALLING // External Event Mode with Falling edge trigger detection

o GPIO\_MODE\_EVT\_RISING\_FALLING // External Event Mode with Rising / Falling edge trigger detection

###  What are the steps to initialize a common peripheral? What library functions are available for this?

The GPIO port you want to use must always enable the clock in the \_\_GPIOx\_CLK\_ENABLE () macro

using. After a reset, all feet are in input floating mode, so the HAL\_GPIO\_Init () function is correct

can be changed by parameterizing. One of the input parameters is the given port (GPIOA, GPIOB, ..), the other is

settings description structure (GPIO\_InitTypeDef).

Each peripheral has a separate source and header file (eg stm32f4xx\_hal\_gpio.c, stm32f4xx\_hal\_gpio.h),

to use the stm32f4xx\_hal.h and stm32f4xx\_hal\_cortex.h header files must be included.

###  List the library functions that almost all peripherals have!

Most peripherals have the following functions:

o PPP\_DeInit (...) - resets the PPP registers to a reset state

o PPP\_Init (...) - set peripheral parameters

o PPP\_Cmd (ENABLE / DISABLE) - enable / disable PPP peripherals (does not set peripheral clock)

o PPP\_ITConfig (...) - set the interrupt sources for PPP peripherals

o PPP\_GetFlagStatus (...) - read peripheral flags (polling)

o PPP\_ClearFlag (...) - clear peripheral flags

o PPP\_ClearITPendingBit (...) - Clear IRQ flags

###  What are the priority groups and what is the main difference between them?

There are a total of five priority models to choose from that determine the four available

how many bits per bit to use for group priority and how many for intra-group sub-priority. The interruption

controller always allows interrupts with a higher group priority to take effect, these are the lower group

priority interrupts may also be interrupted. A sub-priority dictates that it occurs simultaneously

the order in which the controller will enforce interrupts of the same group. In the interrupt controller a

lower numbers represent higher priorities.

SZN

###  What functions do we need to write in the STM32 HAL library to handle an interrupt?

### What are the main tasks these functions perform?

Functions that can be used to configure the interrupt controller:

 HAL\_NVIC\_SetPriorityGrouping - setting the priority model

 HAL\_NVIC\_SetPriority - set the priority of a specific interrupt

 HAL\_NVIC\_EnableIRQ - enable a specific interrupt

 HAL\_NVIC\_DisableIRQ - disable a specific interrupt

The library performs all the checks for us and also clears the interrupt flag, only the

we need to write application-specific handling in the callback function.

Pop-up: The first and last two questions

SZN

# Microcontroller Lab, 6th Measurement Control Questions

###  What is the Call stack window for? How to find out where the program is right now, which sequence of function calls did you make?

This shows through which functions we got to where we are right now. For example, you can find out exactly where the function we are in now called from. You can also jump there by double-clicking on its name.

###  What does the extern keyword mean for a global variable?

An external reference is in another file as a global variable and we want to use it.

###  What is the difference between & and && operators in C? Set an example different to work!

& the bit operation. Apply the AND operation bit by bit. 1010 & 1100 = 1000

&& is a logical operation. AND relationship between 2 boolean variables. Return 1 if both are non-zero and zero otherwise. 4 && 8 = 1

###  Write a “void for allCharacters (const char \* text)” function that is the resulting string

calls the “void elkuld (const char c)” function for each character, and in the meantime

it only goes through the string once.

void for allCharacters (const char \* text) {

char \* temp = text;

for (; \* temp! = '/ 0'; temp ++) {

gold (\* temp);

}

}

###  What’s wrong with the following snippet of code trying to write “Hello” into a string?

char \* text;

strcpy (text, "Hello");

There is no space in the memory for him.

### Your program runs to and endless loop. You suspend the execution and see it is while(1), the function names seems to relate to and assert function. How do you figure out what led to this state?

Call using stack window

### Why use predefined constants instead of wiring specific values into source code?

Easier to modify later, more transparent.

# I2C communication

###  Describe the I2C protocol: one-way, half-duplex, or duplex?

The I2C two-wire synchronous data transmission system, developed by Philips to connect integrated circuits, consists of two wires: SCL (clock) and SDA (data).

Serial, 8-bit, two-way data traffic with a maximum speed of 100 kbit / s in standard mode and 400 kbit / s in fast mode.

The data is bidirectional, but only on one wire: half-duplex.

###  What is the SCL line for?

SCL (clock)

###  What is the SDA cable for?

SDA (data)

###  What is the role of the master?

There is a simple master / slave connection between the communication devices. The master always initiates and controls the communication (the master provides the clock signal) and the master can act as both a transmitter and a receiver.

### What special communication conditions (in other words: phases) exist in the I2C protocol and when are they met, for example, how do we signal the beginning of a communication?

To learn about the characteristics of communication on the I2C bus, see the timing of sending a byte in the figure below. When sending data, the status of the data line can only change when the clock is low. The data line must remain stable during the high state of the clock. If the status of the data line changes at high clock levels, it always means a special condition (START, RESTART or STOP). The start of data transmission is indicated by the START condition (at high clock levels, the data line changes from high to low). The end of the data transmission (not necessarily after the first byte!) Is indicated by a STOP condition at which the data line changes from low to high at a high clock level. The RESTART condition consists only in generating a new START condition without replacing the previous transmission with a STOP signal.

we would have closed it.

###  What is the abbreviation for I2C?

Inter-Integrated Circuit = between integrated circuits

### What steps need to be taken to read a from the I2C thermometer temperature? (In detail, when to instruct the I2C peripheral?)

If the initialization is present, the data must be read from the register.

HAL\_I2C\_Mem\_Read (& hi2c, deviceAddress, registerAddress, 1, pData, dataSize, i2c\_timeout);

Read the "dataSize" byte from the "registerAddress" register of the "deviceAddress" device to the buffer shown by "pData"

# SPI communication

###  How many wires are required to use the SPI interface, what are they?

3 + x wire, where x is the number of devices. Synchronous serial system, three signal connections:

DI (data input), DO (data output) and CLK (clock); and an enabling wire for each device.

###  Describe the SPI protocol: one-way, half-duplex, or duplex?

The SPI (Serial Peripheral Interface) bus is a two-way (duplex) synchronous serial

implements communication between two devices.

Simultaneous two-way communication on two data lines, full duplex.

###  What do MISO and MOSI mean?

MOSI (Master output, Slave input)

MISO (Master input, Slave output)

###  What allows each slave unit?

The Chip Select enable sign.

###  How many bits are transmitted during the transmission of a message?

The SPI is 8-bit

###  What are the advantages of SPI protocol over I2C, what are its disadvantages?

I2C is a protocol that can handle multiple master and multiple slave units, while

SPI has a privileged tool in communication (master).

The main advantage of the I2C bus is address-based communication. A device connected to each bus

has a 7 (or 10) bit address that can be identified on the network, i.e. with the SPI

in contrast, there is no need to connect a separate authorization cable to each device.

SPI: full duplex, faster but more wires (CS for everyone)

I2C: half duplex, slower but requires few wires

###  Why you can easily implement the same function in SPI communication during writing and reading?

# UART communication

###  What are the parts of a character frame?

Start bit

Data bits

It can also be a parity bit (not here)

Stop bit

###  What is the role of RX and what is the TX wire?

RxD: data reception

TxD: data transmission

###  Describe the UART interface: one-way, half-duplex, or duplex?

Duplex.

###  What is the main difference between UART and the previous two protocols (I2C and SPI)?

UART is out of sync, no clock is sent.

###  How is the synchronization between transmitter and receiver implemented during communication?

Both the device sending the data and the receiver must send the data at a predetermined frequency the device to sample the incoming data at specified intervals. Before 2 devices to communicate the basic rules for the exchange of information between them need to be clarified. (Baud rate)

###  What condition must be met during communication to avoid frame failure probability is too high?

Synchronization between transmitter and receiver requires very precise timing, so the same communication speed must be used on both sides of the bus

###  What appears on the oscilloscope when you press “Auto Scale” when on the UART

there is no communication and then we want to see a UART frame? Why?

We will only see noise at a logically high level. The scope then examines the data just arriving and so on

adjusts the scaling so that the signal is "as visible as possible" in certain respects

SZN

# CAN communication

###  What types of messages are there?

o Data frame

o Data request message (Remote frame)

o Error frame

o Overload frame message

###  What is the media message for?

It transmits data from one sender to one or more recipients at the initiative of the sending party

###  What is a data request message good for?

Each node can initiate the sending of the information it needs to the data provider

node. This requires a data request message with the same ID as the requested media message

send.

###  What is the priority order of the messages?

An 11-bit ID field indicates the priority, the lower the number, the higher the priority.

###  What are the field types in a message?

Message start bit

Arbitration field

IDE bit (The ID extension bit)

Control field

Data field

CRC field

Acknowledgment field

Message end field

Pop-up: For me, void was a function for all Characters (const char \* text)

and what will appear on the oscilloscope if there is no communication and you press Auto Scale

# Microcontroller Lab, 7th Measurement Control Questions

###  What is QEP and what is its role?

From the signals of the incremental tachometer, the unit that determines the relative position and direction of rotation

It is called QEP (Quadratic Encoder Periphery). Microcontrollers designed for drive control tasks

DSPs contain such peripherals.

###  What are the options for the speed and direction of rotation of a DC magnet motor

### to change?

Speed:

From the equation we can see that the speed of the machine can be changed in several ways.

On the one hand, by installing resistors at a given terminal voltage Uk (lossy solution),

on the other hand, by changing the terminal voltage itself. In practice, the latter is used.

Changing the direction of rotation of the machine (reversal) by changing the polarity of the terminal voltage

possible.

###  Draw the bridge / DC main circuit connection. How to determine the converter output voltage?

###  What controls can be used for a bridge-connected DC / DC converter, what are the advantages of these?

During the measurement, there are basically two types of control, counterclockwise and offset control.

With the counterclockwise control, the output voltage in the range -Ui ÷ Ui can be changed steplessly,

while for offset control only between 0 and Ui.

With offset control, the maximum current ripple is reduced by a quarter compared to counterclockwise control.

Another advantage of offset control is that the pulsating frequency of the output voltage is twice the switching frequency

frequency, so that half the switching frequency can achieve the same current ripple as the countercurrent

in case of control.

###  Why is it necessary to provide a control dead time between the switching elements in a bridge branch?

A real switching element has an on and off delay, so a

control dead time must be provided between semiconductors, otherwise a short circuit would occur.

Pop-up: first and penultimate

SZN

# Microcontroller lab, 8th measurement check questions

### 9.1. 113, 203, 301, 303, 402, 501, and so on. what is the role of circles, how do they work?

??

### 9.2. What is an HMI?

Human Machine Interface implements a human-machine interface.

### 9.3. What are Simatic Manager and WinCC?

Simatic Manager: Writing and simulating PLC programs, programming a device

WinCC: A graphical interface that implements HMI

### 9.4. What is the difference between a passive and an active slave?

The active slave can also intervene.

The measurement has 3 PLCs, 1 master, 1 active slave, 1 passive slave.

### 9.5. What modes do the controllers implemented in the PLC have?

Controllers implemented with software have the following three operating modes:

M (manual): manual mode, the executive signal is set by the operator.

A (automatic): automatic mode, the executive signal is given by the control algorithm.

R (remote): can be remotely controlled via the control network.

### 9.6. What is a positioner? Draw your sketch! What can be its transient function?

A servomotor whose displacement is measured by a sensor and via an internal feedback

regulates. Is the transition function nonlinear?

### 9.7. Why are most regulatory circles non-linear?

The actuators (here valves) have a non-linear operation.

SZN

### 9.8. Draw a transient function and give a dead-time single-storage approximation.

The straight line is the approximation.

### 9.9. Draw a transient function and give an integrative single-storage approach.

SZN

### 9.10 What do we denote by PB?

Proportional Band

PB = 1 / KC.100 [%], where KC is the gain of the controller.

### 9.11. The OPPELT method cannot be applied to a circuit. Which one and why? How would you scale a PI controller?

For the integrating process, the OPPELT method cannot be used (level measurement) as the method for unit jumping

build. Take the transient function of the process and approximate it with an integrating single-storage term.

### 9.12. The total damping factor is a fairly oscillating solution. How would the regulator change parameters to get calmer behavior?

I would reduce the gain of the controller, increase the integration time.

### 9.13. In the circuit containing an equal percentage valve, which is the most unfavorable in terms of stability work point?

Where the steepest is the volume-to-flow characteristic.

### 9.14 From the transient function, we determine the dead time and time constant. What could be the physical cause of these theoretically and a in reality?

Downtime: May be caused by gear shifting by actuators moving the valves

Time constant: due to the finite speed of valve movement

Pop-up: Everyone was different, all questions were included

# Microcontroller Lab, 9th Measurement Control Questions

### - What kind of fund, resp. additional services does an embedded operating system provide in general?

The basic task of embedded operating systems is to schedule between the tasks to be performed

provide support. Additional features include secure management of shared resources,

and communication between pockets.

### - What do we call a task?

An execution unit that can be scheduled by the operating system.

### - In what conditions can a task be?

The task can be in five possible states: - sleeping - ready to run - running - waiting - interrupted

Tasks in the sleep state are in memory, but their existence is unknown to the operating system

have been deleted. Tasks that are ready to run are ready to run, but the scheduler has not yet specified the

processor usage rights. The task is running if it owns the processor. The task can wait

the occurrence of an event (free resource release, termination of I / O operation, etc.), in which case

is in a state. Finally, the task is in an aborted state if an interrupt occurred while it was running.

### - How does a preemptive scheduler work?

You can always use the processor with the highest priority ready to run. The operating system is everything

when scheduling, check to see if there is a ready-to-run task with a higher priority than the currently running task, if

deprives the currently running task of the processor and gives it the highest priority to run

ready for task.

### - Illustrate the problem of competitive access to a common resource.

Imagine that we have a global variable (common resource) whose value is queried. setting is not

can be implemented with a single assembly statement (such as int variables on an 8-bit processor). Assume,

that one of our tasks wants to set the value and the other wants to read it. It is conceivable that

situation that the scheduler is deprived of the right to run from the task that sets the value of the variable exactly when

when you have already set one byte of the int type variable but not the other. If this is the query task

get run rights, it will not read from the int variable the value that the other task wanted to set,

but another invalid value (since it could only overwrite one byte).

### - What are the ways to protect shared resources from competing accesses?

Protection can be implemented in several ways:

- by prohibiting interruptions

by disabling the scheduler

- using a flag variable that can be queried / set by elementary operation (testand-set)

- using semaphore-like objects

- Illustrate the concept of the semaphore!

So a semaphore is an object that can be used to indicate that a resource is busy. If the resource is busy, and

if someone else wants to access it, they will have to wait for the reservation to be released.

SZN

### - What is the difference between binary and numeric semaphores?

A binary semaphore is a special case of a counter-type semaphore. The counting type of semaphores is the number of accesses

counted when someone accesses the semaphore, the counter decreases (waiting for the semaphore) when you release

(indicated condition), it increases. If the counter drops to zero, further accesses wait until

someone will not let go of the semaphore.

### - Describe the possible ways of synchronization between bags!

Using semaphores (The task waits until the other one lets out the semaphore).

### - Describe at least two µC / OS-II functions for handling tasks.

Create Task: Create a task before the scheduler starts or it is also possible from an already running task

Using the OSTaskCreate () function.

INT8U OSTaskCreate (void (\* task) (void \* pd), void \* p\_data, OS\_STK \* ptos, INT8U prio);

The first parameter of the function is a pointer to the function that implements the task, the second is to pass it to the task function

bypass parameter, the third is a pointer to the task stack, and the last is the priority assigned to the task.

number is the higher priority), which also serves to identify the task.

Deleting a Task: You can delete a task using the OSTaskDel () function. Caution, OSTaskDel ()

after a function call, that task can no longer be started. INT8U OSTaskDel (INT8U prio);

The parameter of the function is the priority of the task to be deleted, which is also its ID.

### - Describe the most important µC / OS-II functions for handling semaphores!

Create semaphore: OS\_EVENT \* OSSemCreate (INT16U cnt);

In the parameter you can enter the initial value of the counter.

Set to semaphore: INT8U OSSemPost (OS\_EVENT \* pevent);

Returns the semaphore specified in the parameter to the indicated state (increments the counter).

Returns the semaphore specified in the parameter to the indicated state (increments the counter).

Waiting for semaphore: void OSSemPend (OS\_EVENT \* pevent, INT16U timeout, INT8U \* err)

The semaphore specified in the first parameter is reserved (decreases the counter) if the semaphore is not allowed,

then it waits for the number of clocks specified in the second parameter by blocking the calling task (in isr we never call a blocking

instruction). The third parameter tells whether the semaphore has been reserved or not. timeout or other

an error has occurred.

Waiting for semaphore without blocking: INT16U OSSemAccept (OS\_EVENT \* pevent)

In case you don't necessarily need the resource provided by the semaphore, you can use it

OSSemAccept function to examine the value of the semaphore, if the semaphore is free, the function

book, if not, it returns an error.

### - Describe the most important µC / OS-II functions for handling message queues.

Create queue: OS\_EVENT \* OSQCreate (void \*\* start, INT16U size)

The first parameter of the function is a pointer to the start address of the memory area reserved for the line, the second parameter

and the size of the allocated memory space in bytes.

Queue item: INT8U OSQPost (OS\_EVENT \* pevent, void \* msg)

Puts the element specified in the second parameter in the row specified in the first parameter of the function.

Waiting for new item: void \* OSQPend (OS\_EVENT \* pevent, INT16U timeout, INT8U \* err)

Removes the first element from the row specified in the first parameter, if the row is empty, in the second parameter

waiting for a new item to arrive. The third parameter specifies whether the element was successfully removed from the queue or not.

timeout or other error has occurred.

Query item in queue: void \* OSQAccept (OS\_EVENT \* pevent, INT8U \* perr)

Checks if there is an item in the line specified in the parameter, if so, returns it, otherwise it returns an error.

Pop-up: Preemptive scheduler to protect shared resources from competing accesses