***Task 9: Microprocessors  
Parking Space Control in ARM ASSEMBLY***



Name: Ioannis

Last Name: Nakas

AM:58661

Year: 4th

Date: 10/1/2024

Image containing text, software, monitor, computer icon

Auto-generated descriptionAs we can see from the screenshot below, the code normally compile without error or warnings. I believe the code works properly, so only verification by you is required.

In our case, the microcontroller checks the parking spaces. Every time a car is detected at the entrance of the car park (a real system could use a piezoelectric sensor on the roadway), a check will be made if there are any empty spaces. In this case, a piece of paper will be given to the user with its input number and the bar will be instructed to stand up. Otherwise, the corresponding light will light up. There are three cases: a) there is a car available for parking and at the same time available spaces, b) there is a car available for parking and at the same time there are no available parking spaces and c) there are no parking spaces at all.

**Now let's briefly explain its operation step by step and start from the case where there are parking spaces available and there is a car ready for parking**.

1. Image containing text, software, monitor, computer icon

   Auto-generated descriptionInitially, our microcontroller will check the parking lots. So at the beginning it enters the main it controls the free parking spaces and initializes three variables that are necessary as "sensors" in our microcontroller (weightSensor, ledState, motorControl). The motorcontrol sensor is for the bar that will allow the car to enter, the weightsensor will detect if there is a car that wants to enter the parking, the parking spot will check the available parking spaces and the ledstate will be a simple led.
2. Image containing text, screenshot, software, number

   Auto-generated descriptionWe then "denote" that there is a car ready to park (line 42), and the assembly logic is called via the Omada9 function (&weightsensor, &parkingSpots, &motorControl, &ledState).
3. After entering the assembly code, it makes a check of the sensors as well as whether there are parking spaces available. If there are any, he continues going to line 14 where he "opens" the bar for the car to enter. It saves the new status value of the motorControl (i.e. raises the bar) and continues below.   
   Image containing text, screenshot, font, software

   Auto-generated description
4. Image containing text, software, number, computer icon

   Auto-generated descriptionThen, in line 16, it reduces by one space the available parking spaces (if the car enters) and saves to the R3 register the new value of available parking spaces and gives the note to the user. Then it goes to the label end\_check where it comes out of the assembly code.
5. Then he "closes" the bar and sets the car detection sensor to zero. It then displays on the screen the new number of available parking spaces and continues again to the assembly code.

Image containing text, screenshot, software, computer icon

Auto-generated description

**Let's also consider the case where there is no available parking space by changing the code line with the available parking spaces (parkingSpots).**

Image containing text, font, line, number

Auto-generated description

1. We observe the same results with steps 1-3 but now because there are no parking spaces available it goes to the label no\_spots (line 21) where it activates the LED, showing that the parking lot is full and saves the status of the LED to the R4 register.

Image containing text, screenshot, software, computer icon

Auto-generated description

1. Then it goes to the label end\_check and exits the assembly code, going back to the code of C. There it checks the status of the LED and because our LED is activated it will display to the user the message ("Parking Full – LED ON").

Image containing text, screenshot, number, software

Auto-generated description

**Let's consider the case where there is not even a car (weightsensor=0). The price of our available parking spaces is indifferent since there is no car to park**.

1. The weight sensor has a value of 0 (there is no car) and then we enter the assembly code again.

Image containing text, screenshot, font, software

Auto-generated description

1. Then he checks and after seeing that the sensor value is 0 (line 7) he goes to the label no\_car.  
   Image containing text, screenshot, software, number

   Auto-generated description

1. After going to the label no\_car changes the value of the motor control to 0 and saves the value to the register R2. Then it comes out of the assembly code without making any changes.  
     
     
   Image containing text, screenshot, font, line

   Auto-generated descriptionThat was the end of the work, thank you for your attention :)