# Name: Hany Ali Elesawy

# **Task 2:**

**Question:** *Explain the difference between the vectored and the non-vectored interrupts and how each of them can be implemented. Which one of these two methods can be applied using one of the AVR family microcontrollers?*

***Ans:***

* **Vectored interrupts**: The address of the ISR (Interrupt Surface Routine) is known to the CPU. These addresses are predefined in memory (interrupt table). This table can be found in the datasheet of microcontrollers under the Interrupt section.  
    
  When an interrupt is fired, the interrupting module or service sends its specific vector number to the CPU. The CPU will then look up this number in the vector table. And then it branches to the ISR associated to the interrupting device/module.
* **Non-Vectored interrupts:** The CPU has a hardware fixed address called the interrupt vector. The CPU will push the PC to the stack (location in C program memory) then to the interrupt vector address and then branches to ISR which is a hardcoded ISR in a specific portion of memory.  
    
  The CPU has no idea which module/device fired the interrupt signal. So, we have to **poll** peripheral (check the flags) to find out which device requires servicing.
* Vectored Interrupts are used in **AVR Microcontrollers**.

## **Task 3:**

***Question:*** *Define what a communication protocol is. And briefly discuss its specifications (medium, data flow, synchronicity… etc).*

**Ans:**

* **Communication Protocol:** A set of rules and regulations that allow two electronic devices to connect to exchange the data with one and another.
* **Specifications:  
    
  - Medium *(wired or wireless)*:** The medium at which data is transferred between devices.  
    
  **- Transmission Technique or data flow *(Serial or Parallel):*** this topic has a lot of depth but in short, in Serial Communication data is transferred serially over a common link one after the other and not simultaneously . in Parallel Communication multiple data bits are simultaneously transmitted over multiple communication links.  
    
  **- Topology:** Determines the way in which different devices of the network are arranged and how they communicate with each other. Examples: point-to-point , Bus , Mesh , Star, Ring.  
    
  **- Data Direction (Simplex , Half Duplex, Full Duplex)**  
  **-** **Synchronicity: Synchronous:** Shared Timing Scheme **Asynchronous:** Timing Scheme Not Shared.

# **Bonus Task**

**Question:** Explain the concept of a “callback” function in C. In which cases can we use it? And briefly illustrate how it can be implemented.

**Ans:**

* **Callback Function:** A callback function in C is function that takes a function pointer as its argument. For example, “function 1” is passed by reference (A pointer passed) to “function 2” using a pointer to function.   
    
  **SetCallBack( void (\*Ptr\_funcToBeCalled)(void) );**To understand this syntax we can use a rule called **Spiral Outward AntiClock (SOAC)**So **Ptr\_funcToBeCalled is a pointer to function that takes (void) as an argument and returns void.**
* A useful use for this principle is in the ISR (Interrupt Surface Routine) where a piece of code is executed when interrupt is fired.  
    
  As known, in the **Static Layered Architecture** the call direction must be in this way ( APP --> HAL --> MCAL) Thus, we can’t include APP files in The MCAL Layer, and yet we have to write an Application type code in the ISR.  
    
  We can use the concept of callback function to solve this problem.
* **Callback Implementation in Static Layered Architecture:** I will insert an example of implementing a simple Timer0 driver using the concept of a callback function.