**Topic:** Interrupts

**Pre-requisite knowledge:** IO programming

**Modules covered from software manual:** Interrupts (Section 4)

**Components required:** Fire Bird V with 8051 adapter board

**Basic Concept:**

* **What is an Interrupt?**
  + Any signal that causes break in continuity of some ongoing process is an interrupt.
  + In microcontrollers, interrupt signal halts the execution of main program and dedicates processor to another task.
  + These can be software or hardware and external or internal interrupts.
* The program which is associated with the interrupt is called the **Interrupt Service Routine** (ISR) or **interrupt handler**.
* **ISR** is a piece of code that tells the processor or controller what to do when the interrupt occurs. After the execution of ISR, controller returns back to the instruction it has jumped from (before the interrupt was received).
* There are different types of interrupts, which we will be studying here:
  + Timer Overflow Interrupts
  + Timer Compare Interrupts
  + Serial Interrupts
  + External Hardware Interrupts
* **Working of Interrupts:**
  + Upon receiving the interrupt signal the microcontroller finishes current instruction and saves the PC on stack.
  + It jumps to a fixed location in memory depending on type of interrupt
  + Starts to execute the interrupt service routine until RETI (return from interrupt)
  + Upon executing the RETI the microcontroller returns to the place where it was interrupted.
  + Then it performs that service routine and returns back to main function after its completion.
* **Why interrupts are necessary?**

Interrupts are required to:

1. Perform many operations simultaneously.
2. Avoid busy waiting or polling (stop unwanted processes).
3. To interface application that requires synchronization.
   1. Timer Overflow Interrupts

Timer Overflow Interrupt:

Detailed explanation of concept

External hardware interrupts is used to control position. Robot’s left wheel position encoder is connected to Interrupt1 (INT1) pin of microcontroller. <Detailed knowledge of position encoder is covered in next section>

Every pulse from the encoder causes an interrupt to occur in the microcontroller. In ISR (interrupt service routine) of that interrupt an unsigned integer variable “left\_shaft\_count” is incremented and by comparing this count with the required/ desired count, position estimation is done.

* **The code for the above is given in the experiments folder.**

Hope you got the concept of interrupts.