Practical No: 01

Title: Design a function generator using an AVR microcontroller.

Objective:

To design a simple function generator using an Arduino Uno board and an R-2R ladder digital to analog converter (DAC).

Introduction:

Function generators are a very useful electronic instrument for testing electronics and electrical systems. These generators can produce repetitive waveforms with common shapes such as sine, square, pulse, triangular, sawtooth, etc. In addition to the shape of the waveform, a function generator may be able to vary the characteristics of the waveform like frequency/ period, Duty cycle, DC offset, etc.

Exercise 1: Create an 8-bit R-2R ladder DAC in the Autodesk TinkerCAD.

- Create a TinkerCAD account and sign in.
- Create a new circuit in the TinkerCAD and draw an 8-bit R-2R ladder DAC.
- Add an Arduino Uno board to the circuit and connect the DAC to port D.

Exercise 2: Generate a sawtooth waveform using software delays.

- Write a C/C++ code for a free-running counter targeting an ATmega328P microcontroller in the Arduino Uno board. The code should have an option to change the counting speed through the serial interface. You may use the software delay function to control the speed of the counter.
- Assign the counter value to port D, where the DAC is connected. Run/simulate the program in the TinkerCAD and observe the output of the DAC using the pre-built oscilloscope of the TinkerCAD.
- Vary the delay time and measure the period of the waveform.

Exercise 3: Generate a sawtooth waveform using the timer/ counter1 interrupt.

- Write a C/C++ code to configure the 16-bit timer/ counter1 module of the ATmega328p microcontroller run at 100 kHz in interrupt mode.

- Create a 100-element array in the code and fill it with 100 consecutive points of a sawtooth waveform.
- Calling the array inside the interrupt service routine, generate a sawtooth analog signal using the above DAC.
- What is the frequency of the generated sawtooth signal?
- How do you change the frequency of the signal in real-time?

Exercise 4: Generate a sine signal using the timer/ counter1 interrupt.

- Similar to Exercise 2, create a 100-element array that holds samples of a sine signal.
- Modify the above C/C++ code to integrate the sine signal generation capabilities into the system.

Exercise 5: Integrate waveform type and frequency value selection options.

- The signal type and the frequency of the function generator need to be set by sending the frequency value and signal type through the serial communication port. Extend the above code to fulfill this requirement.

****