Lab-9 Group -4 report

(210020050,210020007)

Objective

To configure and program the Tiva C Series TM4C123GH6PM microcontroller to communicate with the MCP4725 I2C DAC and generate an analog waveform, excluding square waves. This exercise aims to demonstrate DAC configuration using I2C communication for generating various waveforms.

- Configure the Tiva C Series microcontroller to communicate with the MCP4725 I2C DAC.
- Program the microcontroller to generate an analog waveform (e.g., triangular or sawtooth wave).

Approach

The program generates a ramp waveform by sending incrementally increasing values to the DAC, which resets after reaching a specified threshold.

Initialization: The I2C_Init() function sets up the I2C0 module for communication on Port B (PB2 and PB3) with open-drain configuration on SDA.

Data Handling: The DAC is addressed at 0x61 in write mode, and data is prepared in two bytes for transmission.

Waveform Generation:

```
while(1){

    I2C0_MDR_R = datasent[i%2];

    if (i ==0 && start==0) {
       I2C0_MCS_R |= 0x03;
       start=1;
    }
     else {
       I2C0_MCS_R = 0x01;
    }

    while (I2C0_MCS_R & 0x01);
    if (I2C0_MCS_R & 0x02) {
      I2C0_MCS_R = 0x04;
      break;
    }
    if(data[0]<4090&&i%3==2){
    data[0]=data[0]+4096/1000;
    }
    else if(data[0]>=4090)
    {
      data[0]=0x0000;
    }
    datasent[0] = (data[0] >> 8) & 0xFF;
    datasent[1] = data[0] & 0xFF;
    i=i+1;

  }
```

The program iteratively increases the DAC input by small increments. Once it reaches a threshold (around 4090), it resets the input value to zero to form a ramp-like waveform. The DAC outputs this as a varying voltage, effectively generating a sawtooth waveform.

Observations

This code generates a repeating sawtooth waveform by incrementing the data value sent to the DAC until it reaches a threshold, then resetting. This results in a smooth ramp-up followed by a reset, which repeats continuously. The output waveform amplitude and we confirmed by observing the DAC output on an oscilloscope.

Conclusion

This lab successfully demonstrates UART configuration and communication on the Tiva C Series microcontroller, as well as its use in controlling external peripherals (LEDs) based on data received serially. The code effectively processes incoming characters, controls GPIO outputs, and sends data back over UART. This forms the basis for serial communication and peripheral control in embedded applications.