

RTES Quiz 2

1.

```
#include <ad5933.h>
#include <stdio.h>

#define BUS_ADDR (0x0D)
#define CONTROL_REG (0x80)
#define STATUS_REG (0x8F)

#define REG1 (0x92)
#define REG2 (0x93)

#define CTRL_TMEASURE (0b10010000)
#define STATUS_TVALID (0x01)

float getTemperature()
{
    uint8_t *bufferArr, tempArr[2];
    int tempVal_int = 0;
    Start_I2C(); //starting I2C and initializes all the pins, enables clock etc.

    I2C_Send_Start_Condition(BUS_ADDR, 1); //addressing the AD5933 device with 1 for writing

    I2C_Send_Start_Condition(CONTROL_REG, 1); //addressing the control register
    I2C_Write_Byte(CTRL_TMEASURE); //Setting the Control Register to measure Temperature
    I2C_Send_Stop_Condition(); //stops the sequence for the writing

    I2C_Send_Start_Condition(STATUS_REG, 0); //addressing the status register to read(0)
    I2C_RequestRead(*bufferArr, 1); //Reading from status register

    if (*bufferArr == STATUS_TVALID)
    {
        I2C_Send_Start_Condition(0x90u, 1); //sending the command to measure the temperature
    }
}
```

```

        I2C_Send_Stop_Condition();          //stop sequence for the writin
g process

        I2C_Send_Start_Condition(BUS_ADDR, 0); //addressing the AD5933 dev
ice with 0 for reading

        I2C_Send_Start_Condition(REG1, 0); //addressing the internal regis
ter 1 where the temp. data is stored, with 0 for reading
        I2C_RequestRead(*tempArr[0], 1);    //Reading from temperature regi
ster 1
        I2C_Send_Start_Condition(REG2, 0); //addressing the internal regis
ter 2 where the temp. data is stored, with 0 for reading
        I2C_RequestRead(*tempArr[1], 1);    //Reading from temperature regi
ster 2
        I2C_Send_Stop_Condition();

        tempVal_int = (tempArr[0] << 8 | tempArr[1]) & 0x1FFF; // Convert
raw temp to integer

        // This conditional statement is used to check the sign of 5th bit
        if ((tempArr[0] & (1 << 5)) == 0)
        {
            return tempVal_int / 32.0;
        }
        else
        {
            return (tempVal_int - 16384) / 32.0;
        }
    }
    else{
        return -1000.0; //PLACEHOLDER value can be used
    }
}

int main(){
    // calls the getTemperature function to get the temperature of the AD5
933 chip
}

```

2.

```

void Timer0_0V_Handler();
void pinRising_Handler();
void pinFallingHandler();

```

```
uint32_t elapsed_time = 0, start_time = 0, end_time = 0, tval = 0, top_counter = 0;

void Timer0_0V_Handler()
{
    if (tval == 250)
    {
        tval = 0;
        top_counter++;
    }
}

void pinFalling_Handler()
{
    start_time = tval + top_counter * 250;
}

void pinRising_Handler()
{
    end_time = tval + top_counter * 250;
    elapsed_time = end_time - start_time;
}
```