REPORT TEMPLATE FOR ESD

General Components we used here:

STM32 nucleo board, Lux meter, Light intensity sensor (LDR), multi meter etc.

Description of the components:

STM32 nucleo board- The STM32 Nucleo board gives developers the easiest possible access to the STM32 MCU portfolio and is compatible with Arduino™ and ST Morpho. Access to all MCU pins through ST Morpho connectors. Flexible power supply via USB or exernal source (3.3V, 5V or 7V - 12V) On-board ST-LINK/V2-1 debugger/programmer.

Lux meter- Lighting professionals use a light meter (also called an illuminance meter or lux meter) to measure the amount of light in a space/on a particular work surface. The light meter has a sensor that measures the light falling on it and provides the user with a measurable illuminance reading.

One lux is equal to one lumen per square metre: 1 lx = 1 lm/m2 = 1 cd·sr/m2. A flux of 1000 lumens, spread uniformly over an area of 1 square metre, lights up that square metre with an illuminance of 1000 lux.

Light intensity sensor (LDR)- Photoresistors, also known as light dependent resistors (LDR), are light sensitive devices most often used to indicate the presence or absence of light, or to measure the light intensity.

LDR is a light-dependent resistor that changes its resistance when different amounts of light fall on it. They work on the principle of photo conductivity where it gives less resistance in high light intensity and high resistance in low light intensity.

Multi meter- A multimeter is a measuring instrument that can measure multiple electrical properties. A typical multimeter can measure voltage, resistance, and current, in which case it is also known as a volt-ohm-milli ammeter (VOM), as the unit is equipped with voltmeter, ammeter, and ohmmeter functionality, or volt-ohmmeter for short. Some feature the measurement of additional properties such as temperature and capacitance.

Analog multimeters use a microammeter with a moving pointer to display readings. Digital multimeters (DMM, DVOM) have numeric displays and have made analogue multimeters virtually obsolete as they are cheaper, more precise, and more physically robust than analogue mustimeters.

Block hardware and sensor code design:

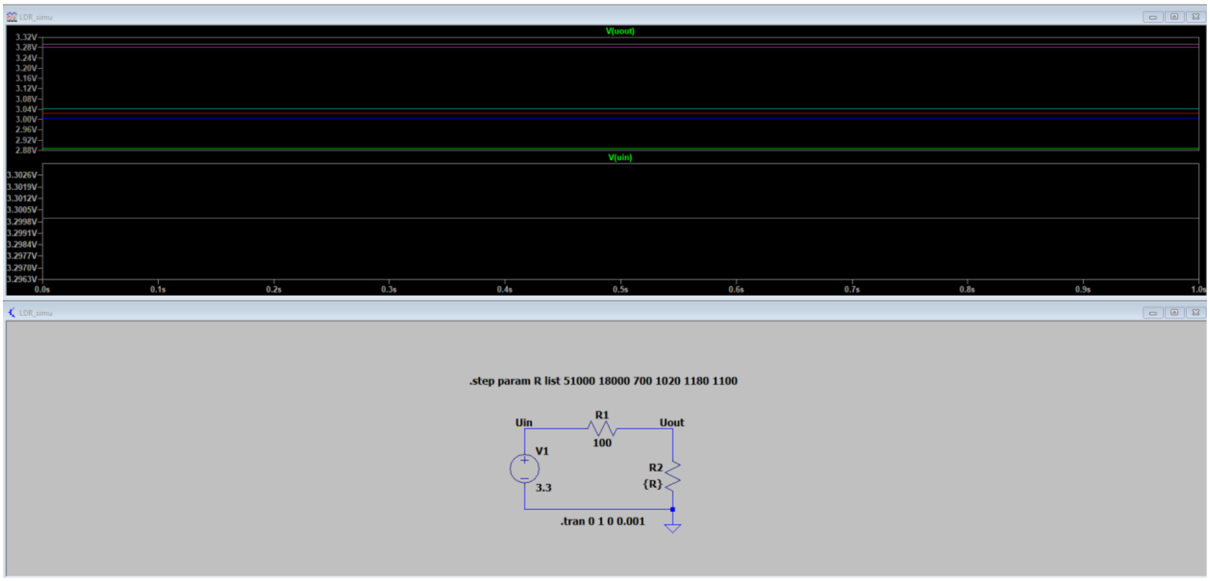
Assignment 1 (prototype testing on breadboard):

Design your own block and test it in the breadboard. The power source for your device is a +12 V DC. Your electronics must be fitted to Nucleo152RE development board inside Arduino connectors.

Choose other connectors so that you can use +12 V DC. Nucleo board can convert +12 V to 3.3V (how?). Serach your sensor datasheet from www.farnell.com.

Simulations with LTspice

In this simulation we checked the functionality of the LDR, by using {R} with script of different values.



We have designed a circuit with input source is 3.3V and output 3.3V. later we can add a circuit which can receive 12 V as an input and feed the LDR with 3.3 V

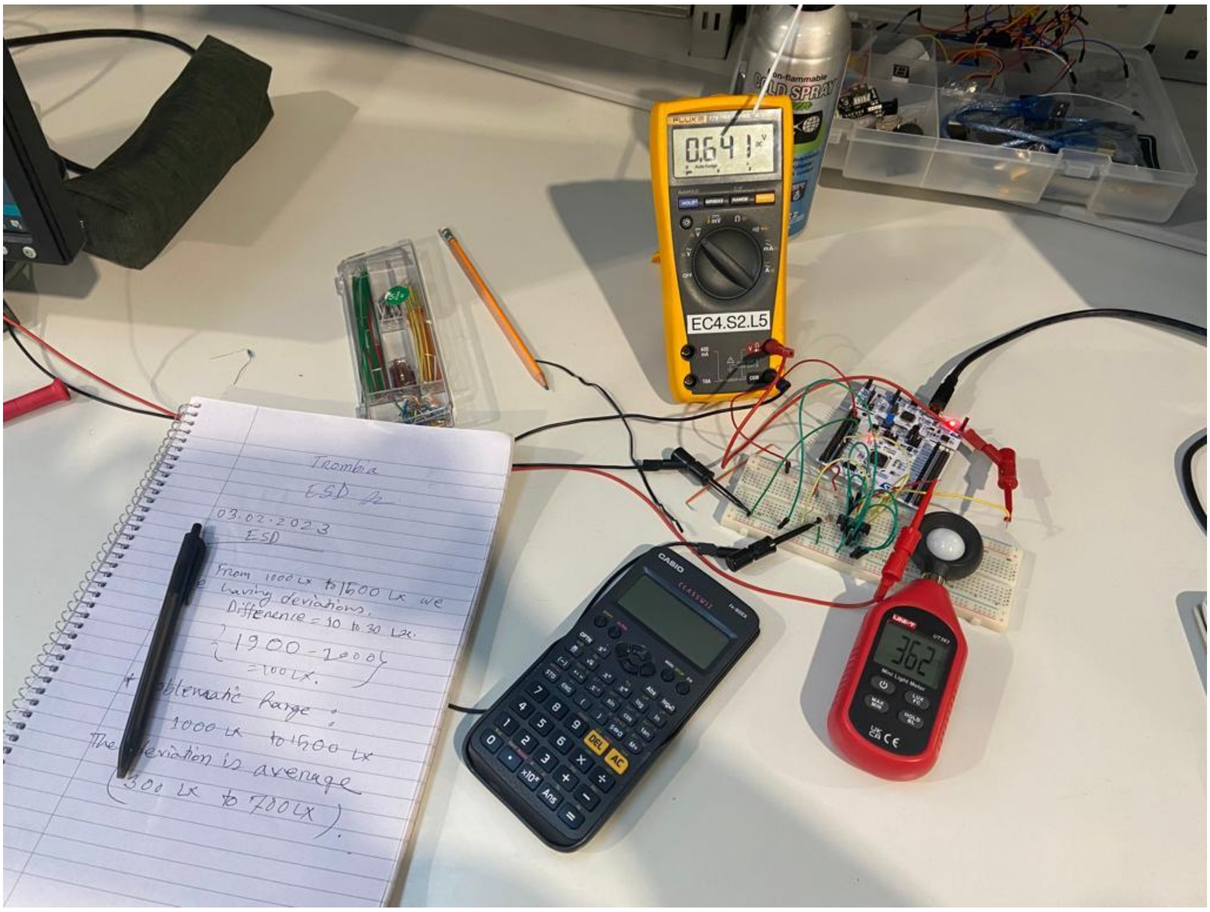


Table bellow has record multiple values for each case. The chosen case has marked with blue color.

LUX	Sensor resistance ohms	Ouput voltage using 5K resistor	Ouput voltage using 10k(3.3v)	Ouput voltage using 100k	
2	37	2.907142857	0.70212766	2.408759124	
20	22	2.688888889	1.03125	2.704918033	
50	16	2.514285714	1.269230769	2.844827586	
100	6.5	1.865217391	2	3.098591549	
150	5.1	1.666336634	2.185430464	3.139866794	
300	2.3	1.039726027	2.682926829	3.225806452	
560	2	0.942857143	2.75	3.235294118	

Lux	Output Voltage(10K)	ADC vref 3.3
7	2.7	3351.27
50	1.8	2234.18
80	1.43	1730
100	1.2	1489.45
300	0.9	1117.09
526	0.66	819.2
700	0.5	620.6
1000	0.4	372.36
1500	0.322	399.67
2000	0.2	248.24
3000	0.18	223.41

Provide some calculation which can support the previous values (Voltage to ADC to LUX) based on ADC formula and the Lux formula which we get it from the Excel curve.

Calculation of Lux value- One lux is equal to one lumen per square metre:

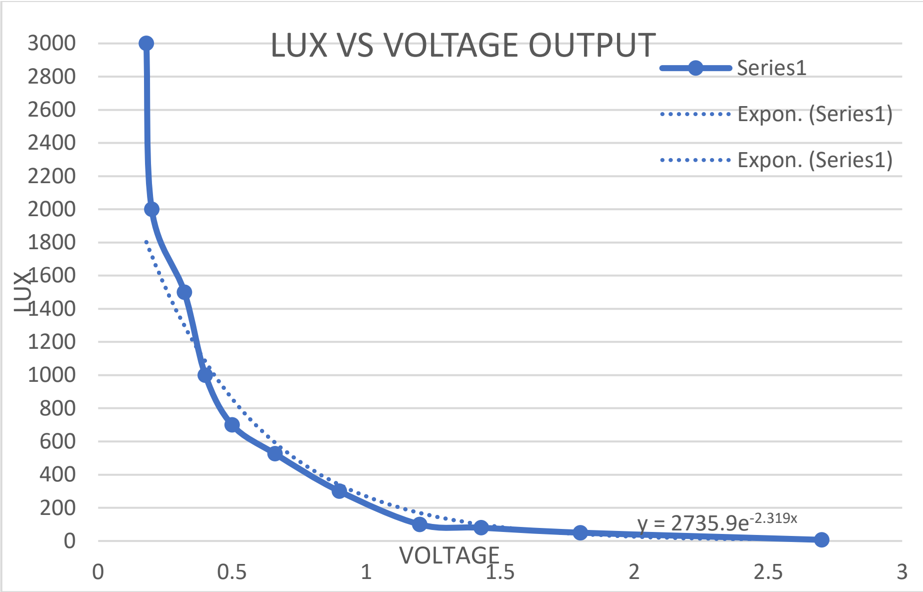
1 lx = 1 lm/m2 = 1 cd·sr/m2.

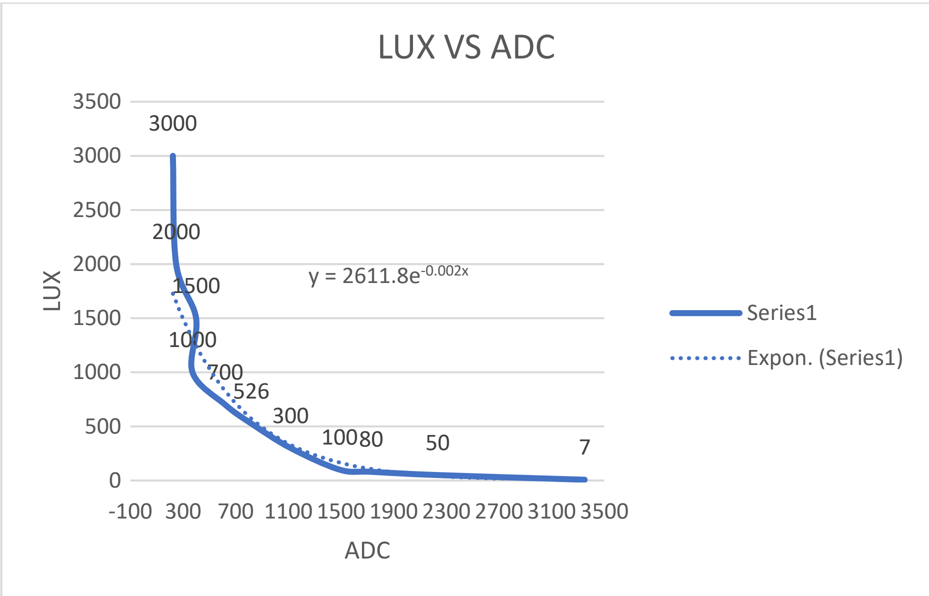
The formula: Illuminance is calculated with the following formula:

Lux [lx] = luminous flux [lm] / area [m^2].

Unit of lux: 1 lx = lumen/m^2

The illuminance is 1 lux if a luminous flux of 1 lumen falls uniformly on an area of 1 m^2. The further away the area is from the light source, the lower the illuminance.



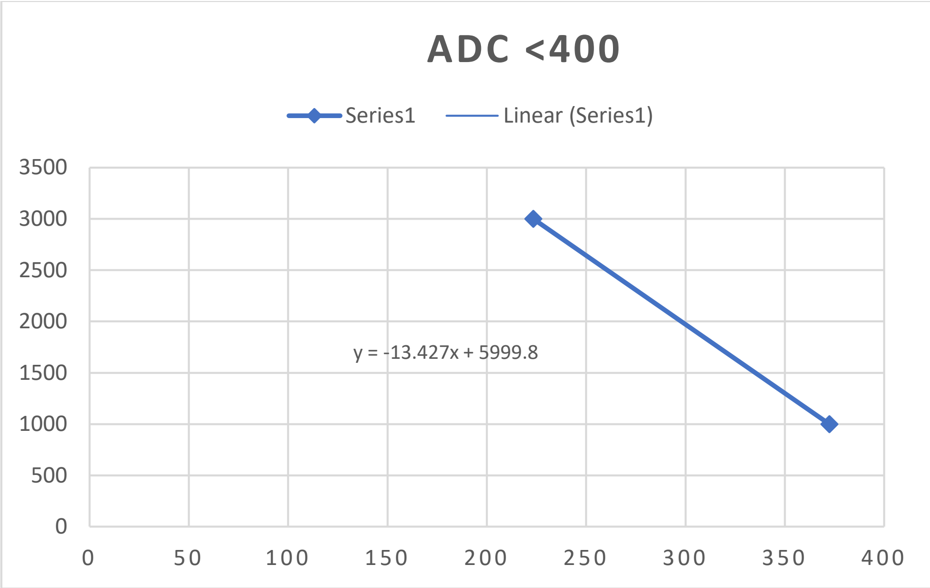


In the curves above show the relation between LUX & VOLTAGE, LUX VS ADC

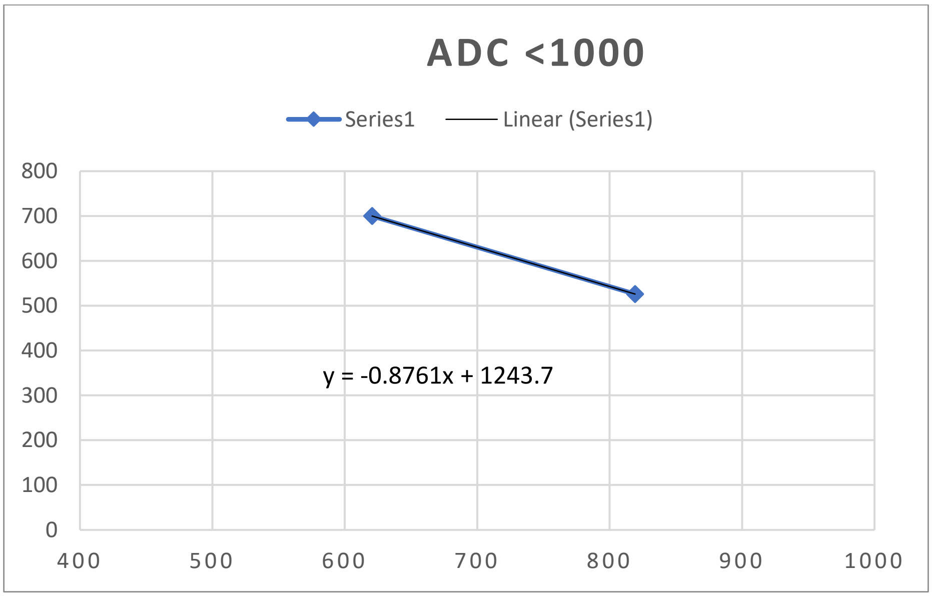
Based on the chosen components and measurements from the previous table, the voltage range we got from 0.18 V to 2.7V. Which is the best reading we could get.

Between 1000 lux and 1600 lux we had estimation we will get some inaccurate values as it can seen from the craves there is a gap between from tanged line and the curve itself.

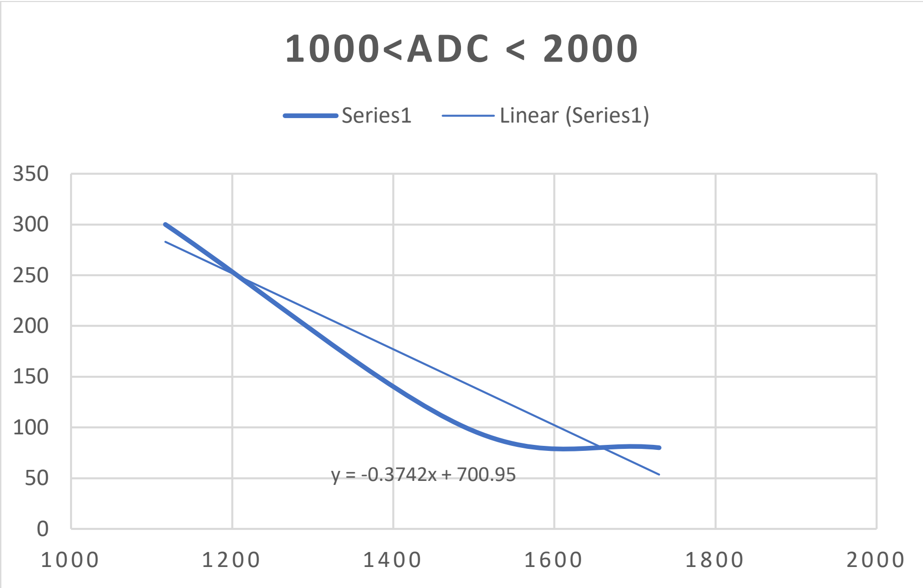
So we decide to divide the exponential curve into 4 linear functions which can make the error ratio smallest as possible.

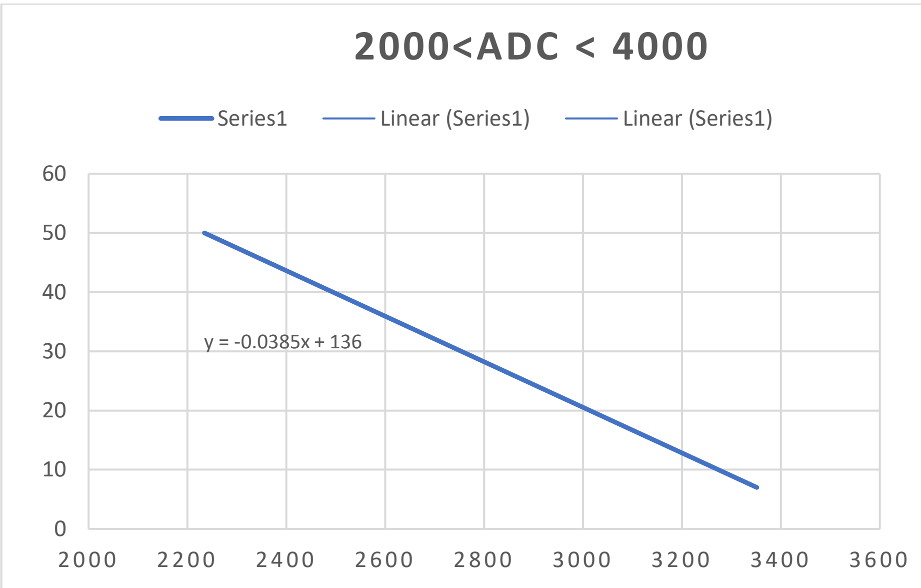


ADC <400
Series1
Linear (Series1)
y = -13.427x + 5999.8
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300
350
400



ADC <1000
Series1
Linear (Series1)
y = -0.8761x + 1243.7
800
700
600
500
400
300
200
100
0
400
500
600
700
800
900
1000


1000<ADC < 2000
Series1
Linear (Series1)
350
300
250
200
150
100
50
0
1000
1200
1400
1600
1800
2000
y = -0.3742x + 700.95


2000<ADC < 4000
Series1
Linear (Series1)
Linear (Series1)
60
50
40
30
20
10
0
2000
2200
2400
2600
2800
3000
3200
3400
3600
y = -0.0385x + 136

Assignment 2 (development of sensor code):

Explanation: Here we have added four formulas. The lux meter was giving different values in different level of intensities. So we had taken several readings; put it in the excel spread sheet and developed graph representation. Then we took 04 certain cases which take us to the closest to the exact values with least error.

Special Observation: The lux meter behaves in an unusual way from 1000 lx to 1500 lx (apprx) range. At this range the lux value deviation than the theoretical value is 300 lx to 1000 lx. But normally the deviation cannot be more than 40 lx.

```
float read_LUX(void)
{
    //char buf[100];

    ADC1->SQR5=0;              //conversion sequence starts at ch0
    ADC1->CR2|=1;              //bit 0, ADC on/off (1=on, 0=off)
    ADC1->CR2|=0x40000000;     //start conversion

    int result = 0;
    float lux = 0;

    while(!(ADC1->SR & 2)){}   //wait for conversion complete
    result=ADC1->DR;//read conversion result


    if((result >0)&&(result < 400)){

            lux = (-13.427*result) + 5999.8;
            }
    else if((result >400)&&(result < 1000)){

            lux = (-0.8761*result) + 1243.7;

        }
    else if((result > 1000) && (result <2000)){

            lux = (-0.5371*result) + 900.01;

        }
    else if ((result >2000)&&(result<4000)){

            lux = (-0.0385*result) + 136;

        }
    delay_Ms(100);

    //ADC1->CR2&=~1;     //bit 0, ADC on/off (1=on, 0=off)
    return lux ;

}
```

Figure- Commented Section of sensor code.

Assignment 3 (Drawing of a circuit diagram for electronics)-

The schematic is not drawn yet.

Significance of components:

We have used the following components for PADS schematic for RS-485 and +12 V DC:

TM_BLOCK/3P, TM_BLOCK/2P, MAX3485CPA+, PTC resistor as a fuse MF-MSMF050-2 - PPTC Resettable Fuse, etc.

TM_BLOCK/3P:

TM_BLOCK/2P:

MAX3485CPA+:

PTC resistor: The PTC (Positive Temperature Coefficient) resistor is also referred to as the PTC thermistor and is a thermistor with a positive temperature coefficient. Its resistance value also increases as temperature increases.

PPTC Resettable Fuse:

a) Radial Leaded Devices
b) Cured, flame retardant epoxy polymer insulating material meets UL 94V-0 requirements
c) RoHS compliant* and halogen free.

Lab 7:

Modbus RTU frame development:

Assignment 1 (Frame development; USART2:

Description:

Modbus -communication protocol is based on the master-slave architecture. It uses RS-485, RS-422, RS-232 interfaces, as well as Ethernet TCP / IP networks (Modbus TCP protocol) for data transfer.

The Modbus RTU message consists of the address of the SlaveID device, the function code, the special data, depending on the function code and the CRC of the checksum.

SlaveID	Function code	Special data	CRC

If you discard the SlaveID address and the CRC checksum, you get the PDU, Protocol Data Unit.

SlaveID is the address of the device, it can take a value from 0 to 247, addresses from 248 to 255 are reserved.

Data in the module is stored in 4 tables.

Two tables are read-only and two are read-write.

9999 values are placed in each table.

Snippets of codes-

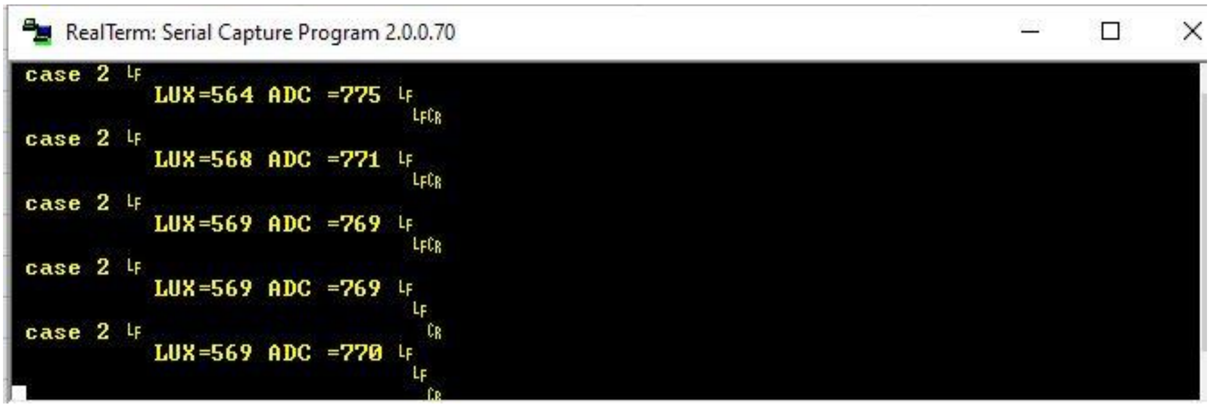


Figure- Realterm Serial Capture

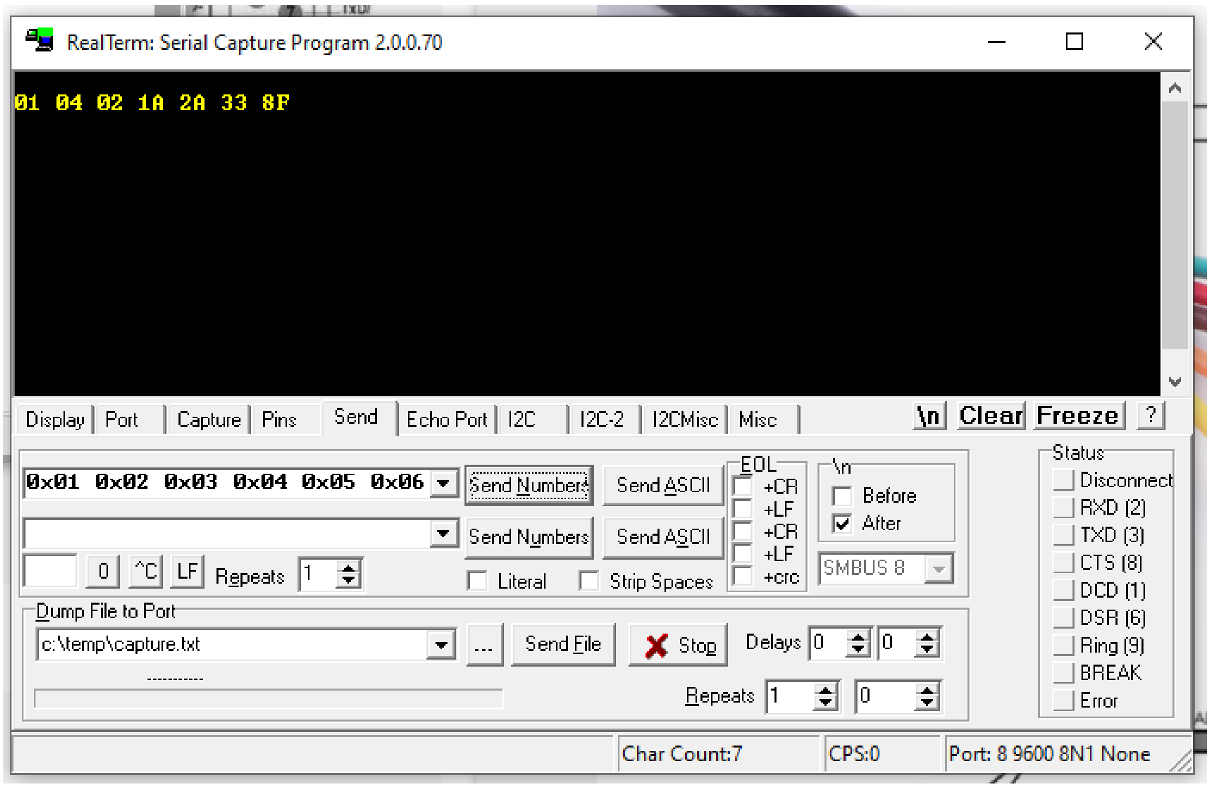


Figure- Master rep

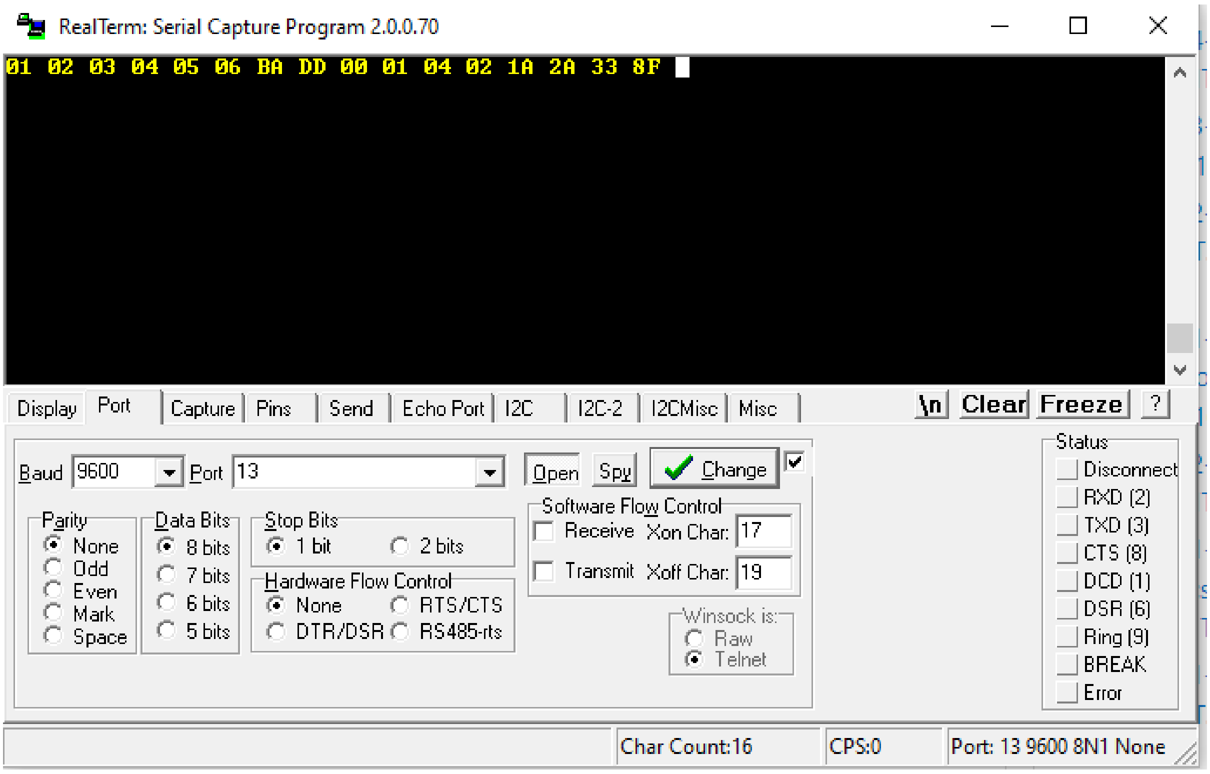
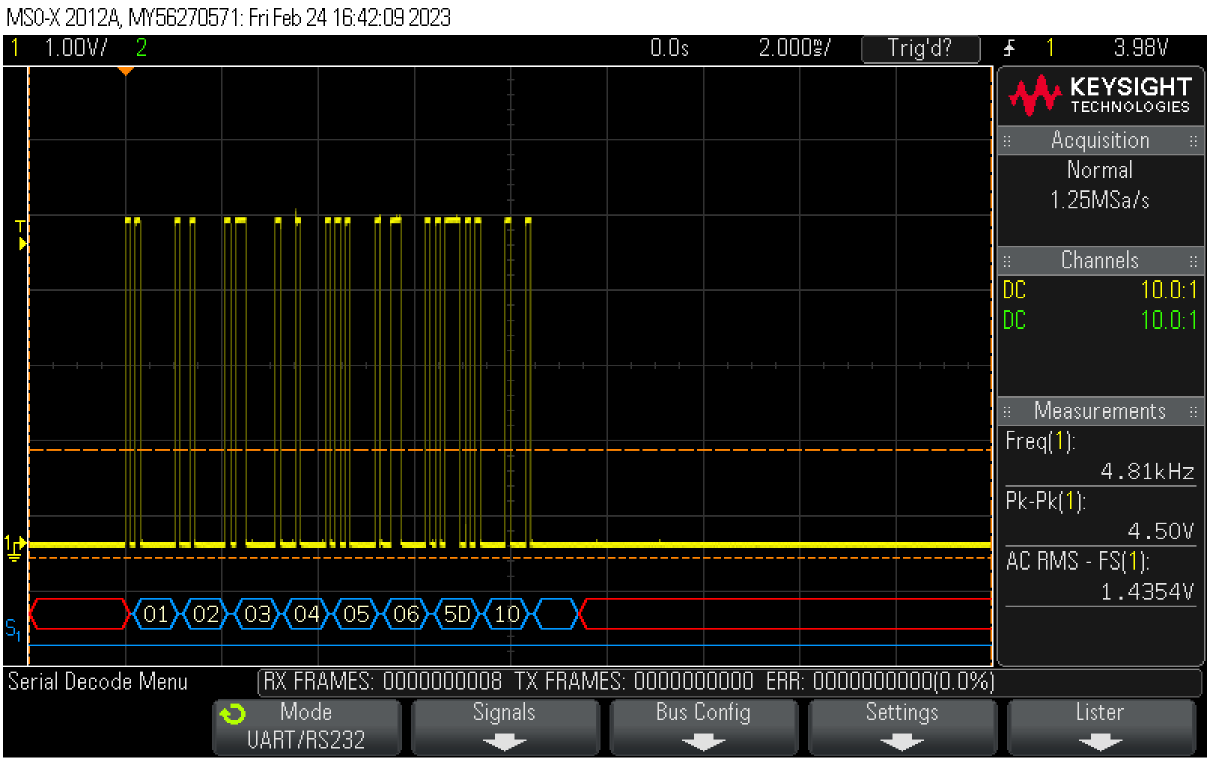
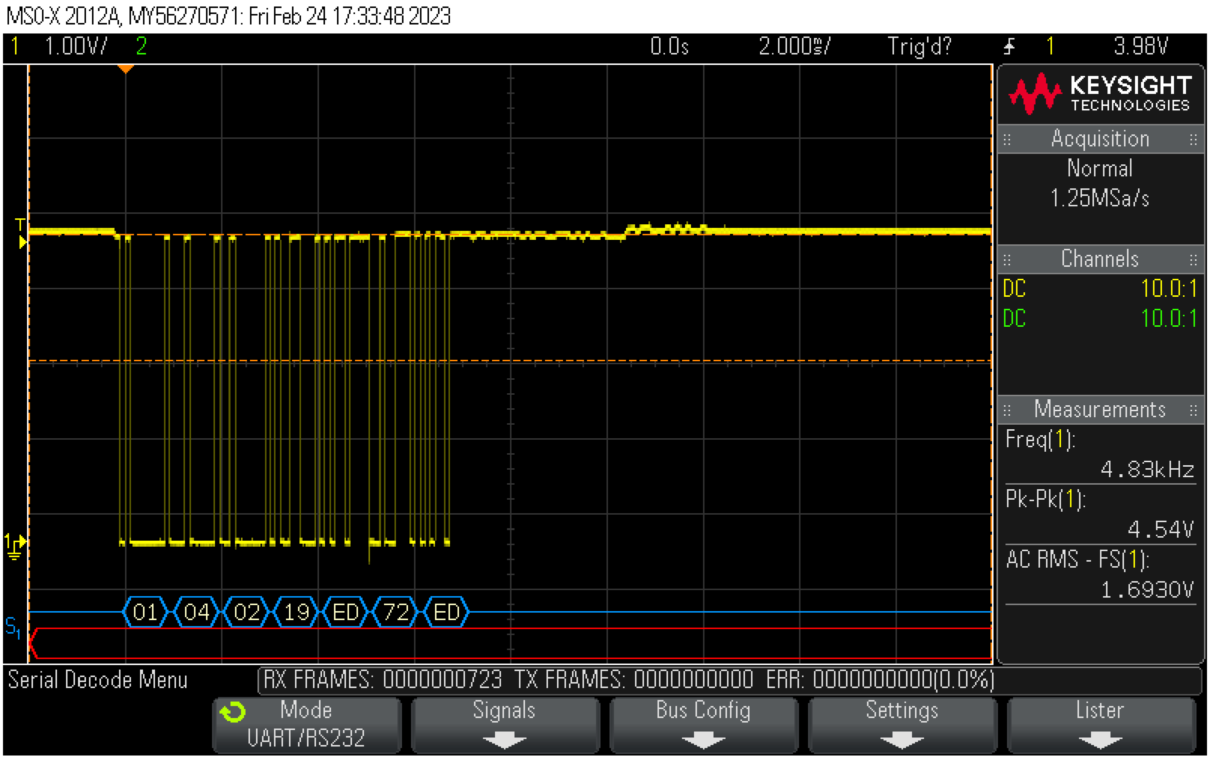


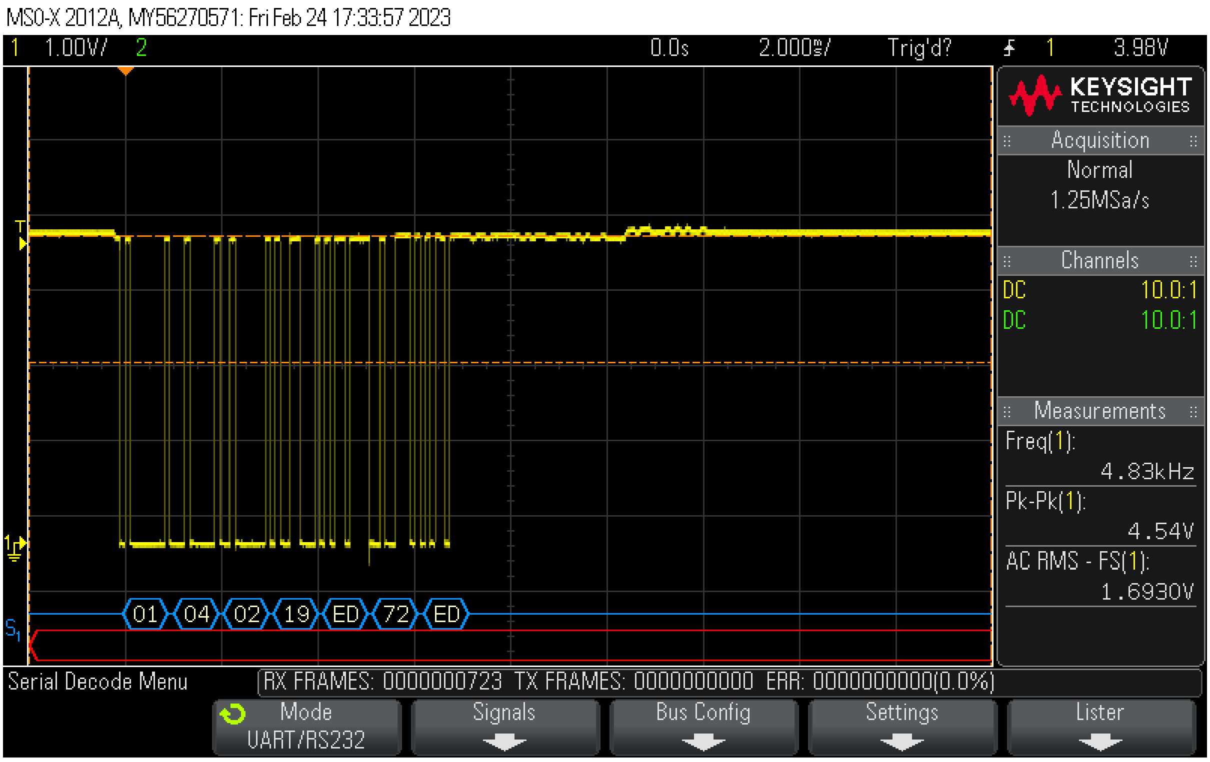
Figure- Master req, rep



Oscilloscope Capture 1 –



Oscilloscope Capture 2 –



Oscilloscope Capture 3 –

Assignment 2 (change frame code for USART1 with MAX3485CPA+):