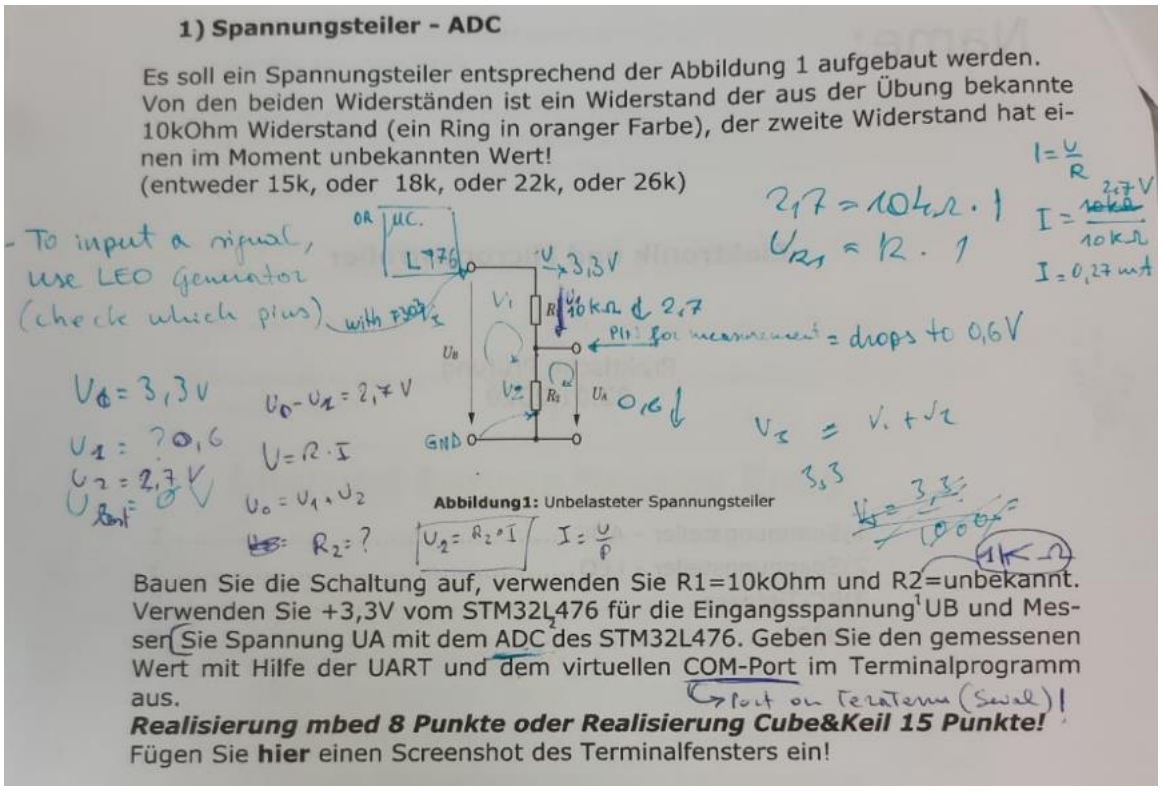
1) 1Spannungsteiler - ADC



1) Spannungsteiler - ADC

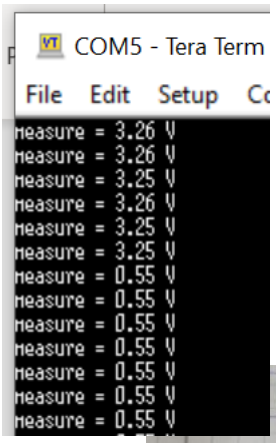
Es soll ein Spannungsteiler entsprechend der Abbildung 1 aufgebaut werden. Von den beiden Widerständen ist ein Widerstand der aus der Übung bekannte 10kOhm Widerstand (ein Ring in oranger Farbe), der zweite Widerstand hat einen im Moment unbekannten Wert!
(entweder 15k, oder 18k, oder 22k, oder 26k)

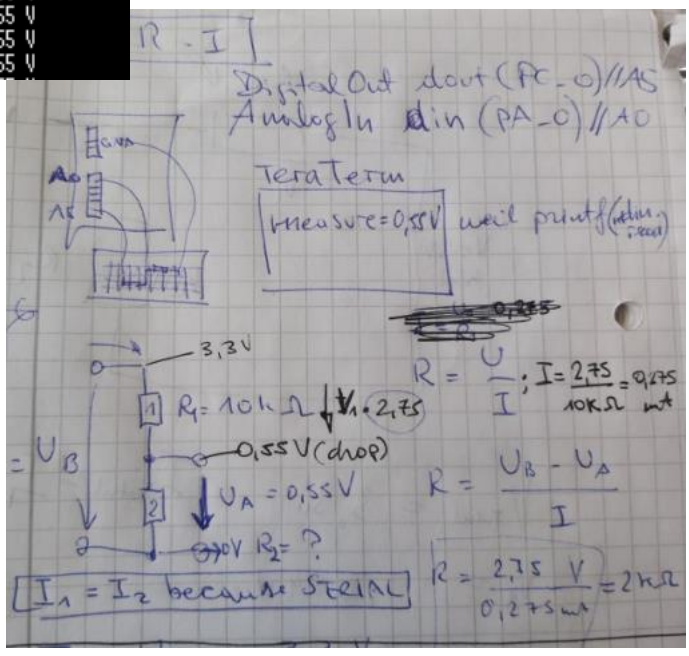
Abbildung1: Unbelasteter Spannungsteiler

Bauen Sie die Schaltung auf, verwenden Sie R1=10kOhm und R2=unbekannt. Verwenden Sie +3,3V vom STM32L476 für die Eingangsspannung UB und Messen Sie Spannung UA mit dem ADC des STM32L476. Geben Sie den gemessenen Wert mit Hilfe der UART und dem virtuellen COM-Port im Terminalprogramm aus.
Port on TeraTerm (Serial)!
Realisierung mbed 8 Punkte oder Realisierung Cube&Keil 15 Punkte!
Fügen Sie **hier** einen Screenshot des Terminalfensters ein!

Code in Mbed: ADC_findR

```
#include "mbed.h"

DigitalOut dout(PC_0);

AnalogIn din(PA_0);

Serial pc(SERIAL_TX,SERIAL_RX);


int main() {

   dout = 1;
   float meas;
   while(1)
   {

      meas = din.read();
      meas = (meas * 3300)/1000;

      pc.printf("measure = %0.2f V\r\n",meas);
      wait_ms(500);  }

}
```



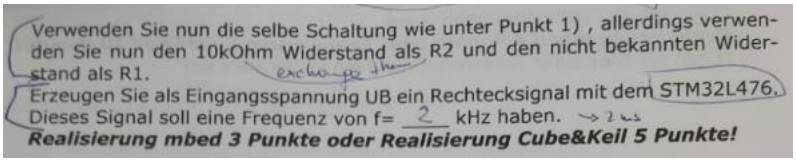


3,3 v - 0,55 v = 2,75 v (Voltage drop)

Intensity is the same everywhere in a serial circuit!
I = 2,75/10k = 0,275 milliAmpere

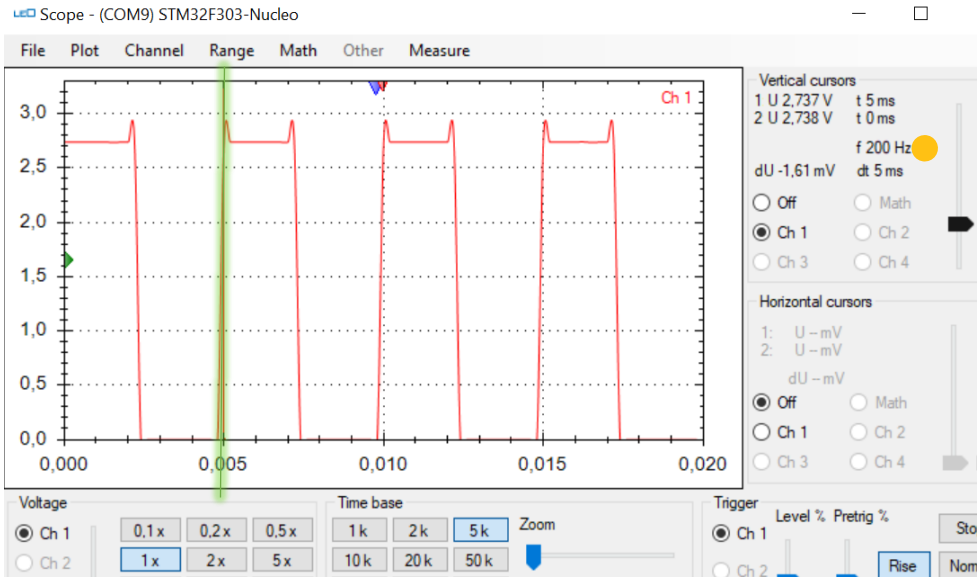
R2 = 2,75/0,275 mA = 2kOhm

2) Spannungsteiler – LEO



Code in Mbed: Output_Rectangle_LEO

```
#include "mbed.h"

DigitalOut rectangle(A5);

int main()
{
   float Hz = 200; // should have been 2 000, but anyway
   float mwait = 0.5/Hz; // Periodendauer berechnen
                         //0.5 für halbe Dauer up/down

   while(1) {

      rectangle = 1;
      wait(mwait);
      rectangle = 0;
      wait(mwait);
   }
}
```



Set also Ch1 on the right side. Bring vertical line to the end of the first cycle so you can check the frequency, which it confirms what you inputted via code- 200Hz

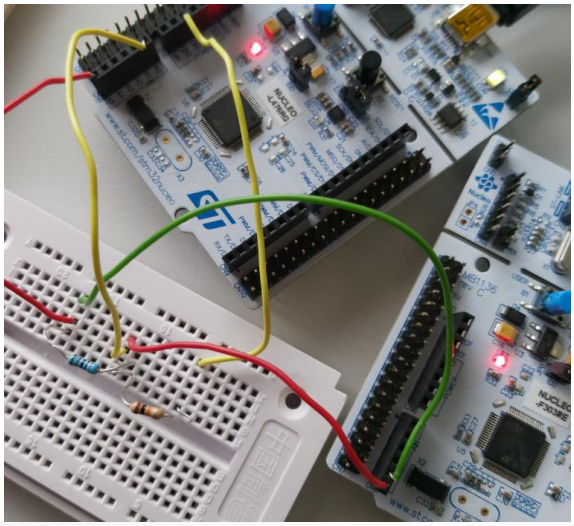
Messen Sie die Ausgangsspannungen UA mit Hilfe des Little Embedded Oszilloskop – LEO am STM32F303.

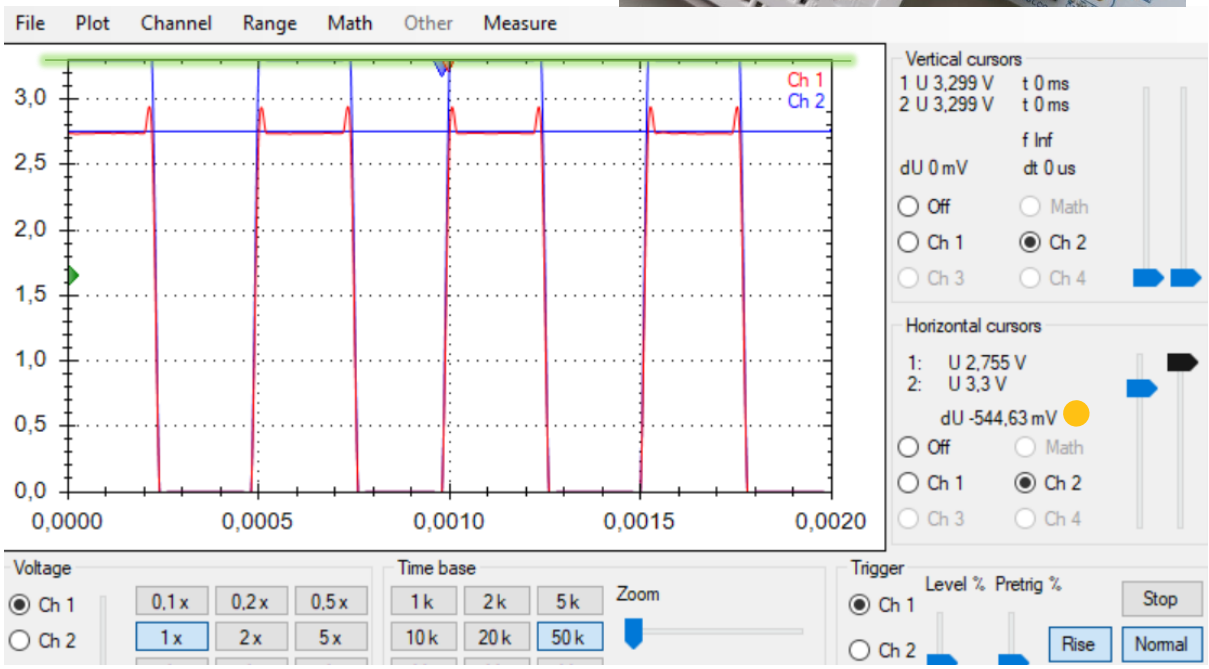
Connect middle pin to AnalogIn pin A5 on F303

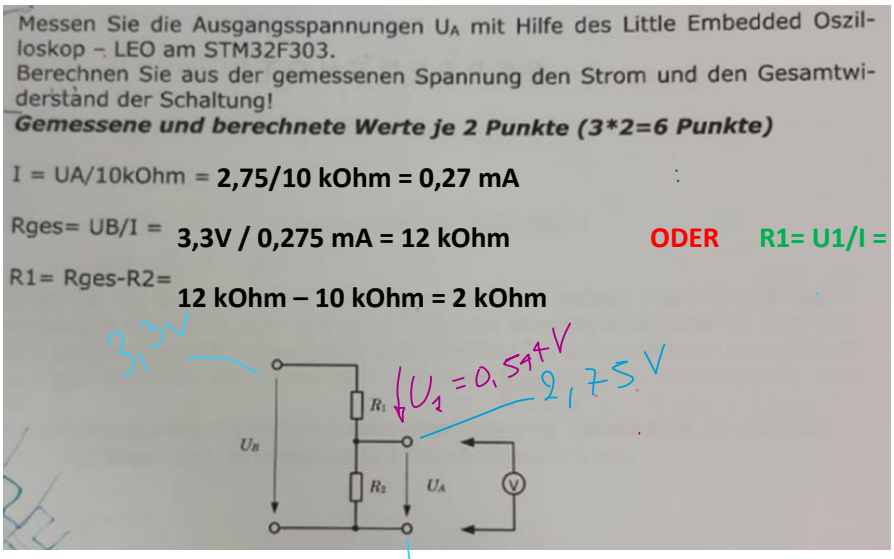
And a second green cable from the initial voltage 3,3 hole to A4 to clearly display it on LEO .

Set 2 Channels. Set the 2 horizontal lines and on the side it should give you the voltage difference

dU (delta U) = 544 mV = 0,544 V



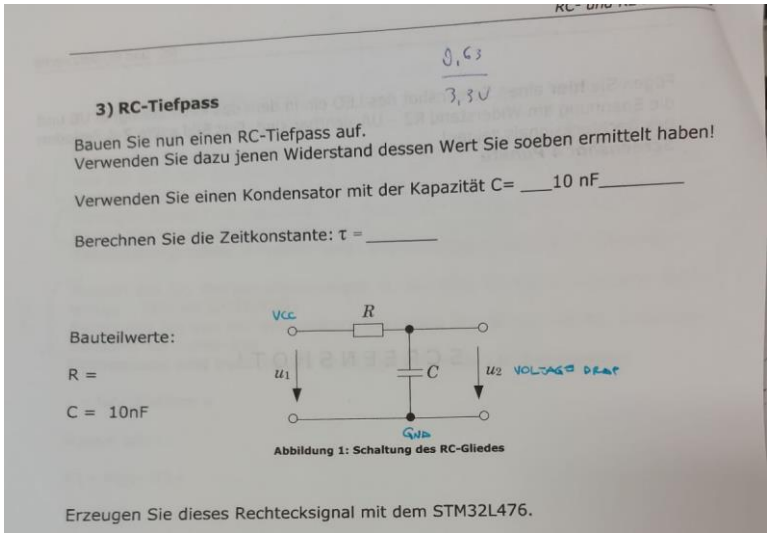




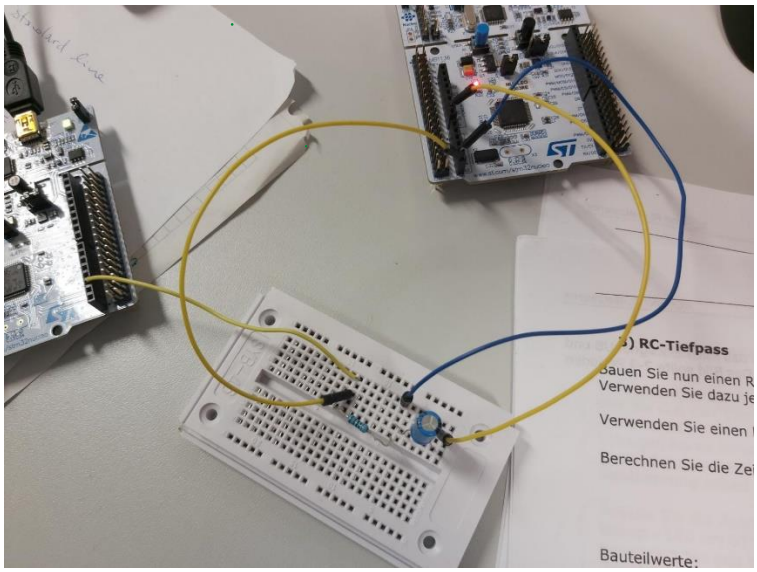
ODER R1= U1/I = 0,544 V / 0,275 mV = 2 kOhm

3) RC- Tiefpass

Instead we have: R= 2KOhm ; C= 100microFar , after calculations -> dt (tow) should be 0,2 s = 200 ms



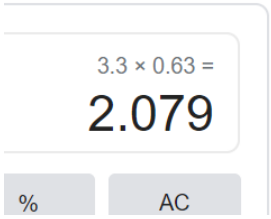
Circuit



CODE in Mbed (/Rectangl_Signal/main.cpp)

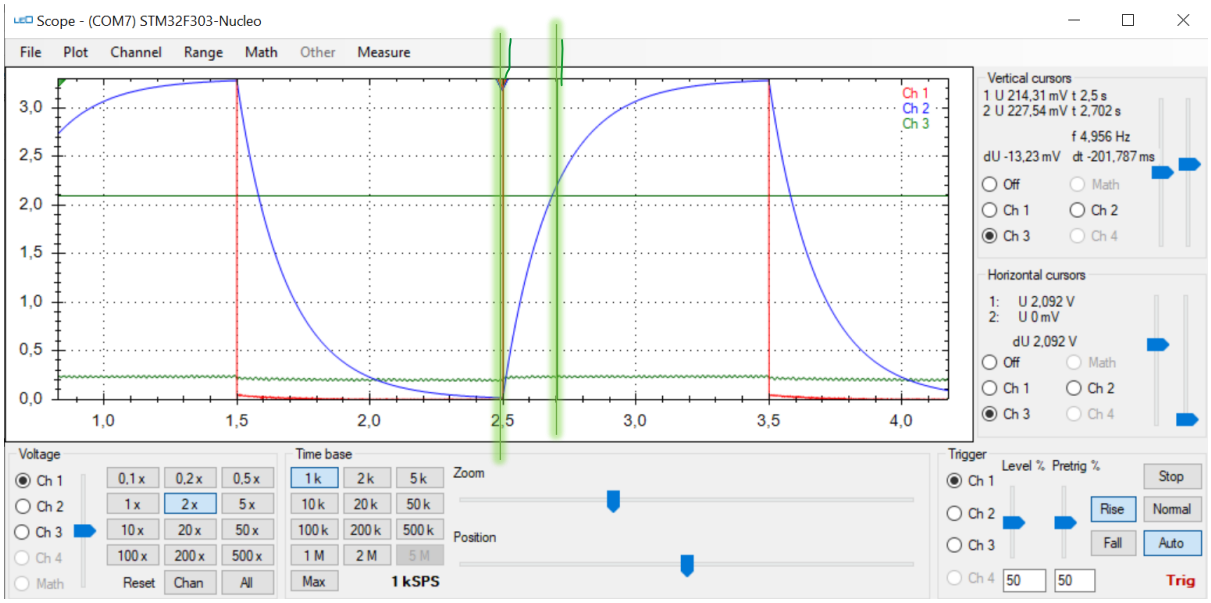
main.cpp

```
#include "mbed.h"

DigitalOut myled(D2);//outputs a digital signal

int main() {
    while(1) {
        myled = 1; // LED is ON
        wait(1); // 200 ms
        myled = 0; // LED is OFF
        wait(1.0); // 1 sec
    }
}
```

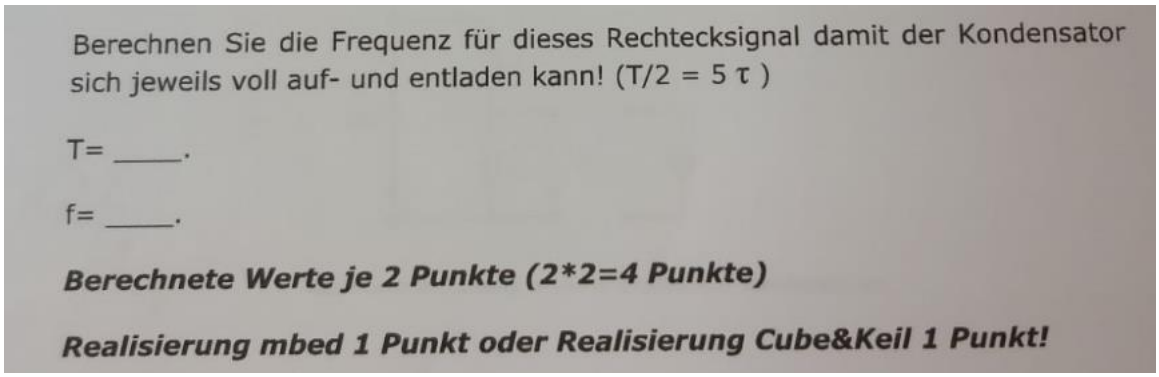


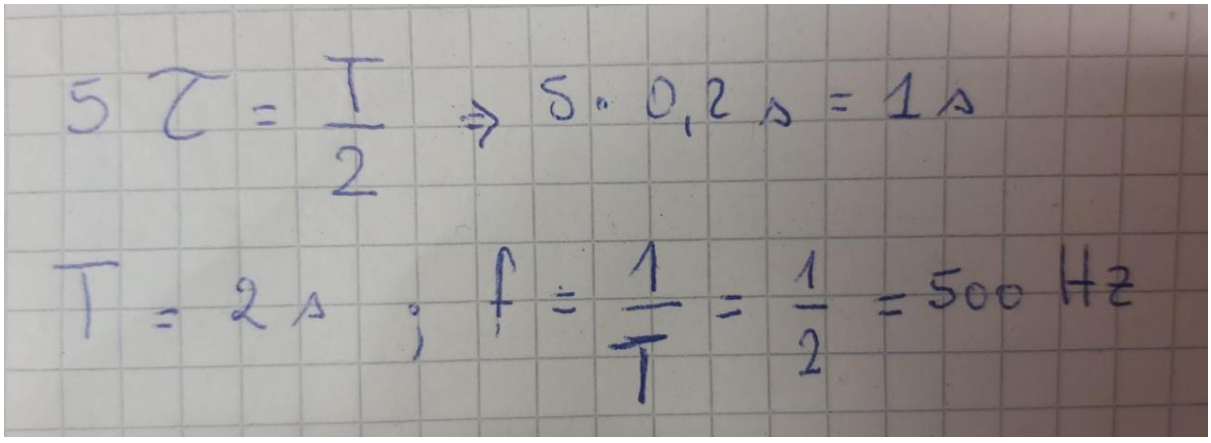
Remember!! You have to multiplicate the max Voltage 3,3 V with 0,63 (63% = (1-e^-1)) and you will get the value where exactly to set the horizontal line in the Oscilloscope -> 2,1 V circa

You should first do some settings for Data Length, Channels, Time base and Zooming



You now see that it outputs dt being 201,787 ms which correspond to our original calculation->0,2 s





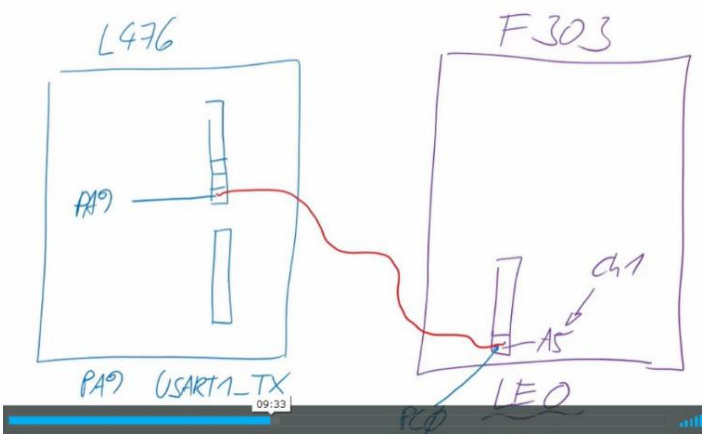
4) UART : Erzeugen Sie mit der UART des STM32L476 ein Signal.

Senden Sie das Zeichen _a_____ Mit einer Datenrate: __9600_____

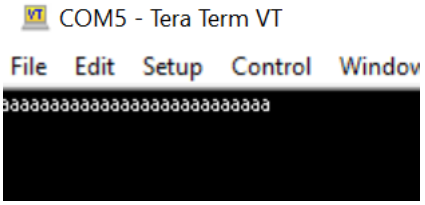
Realisierung mbed 3 Punkte oder Realisierung Cube&Keil 6 Punkte!

CODE: UART/main.cpp

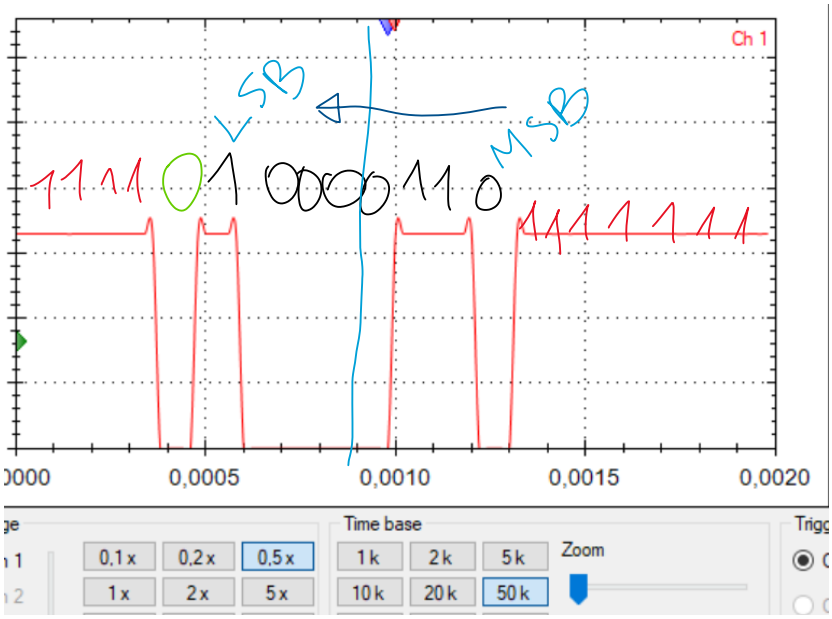
```
#include "mbed.h"

Serial pc(SERIAL_TX, SERIAL_RX, 9600); //UART
//Serial out(PC_4, PC_5, 9600);

int main(){

      while(1) {
      pc.printf("a");
     // out.printf("a");
      wait_ms(250);
   }
}
```




```
7
8  Serial pc(SERIAL_TX, SERIAL_RX, 9600); //UART
9  //Serial out(PC_4, PC_5, 9600);
10
11 int main()
12 {
13         while(1) {
14         pc.printf("a");
15        // out.printf("a");
16         wait_ms(250);
17
18     }
19 }
```



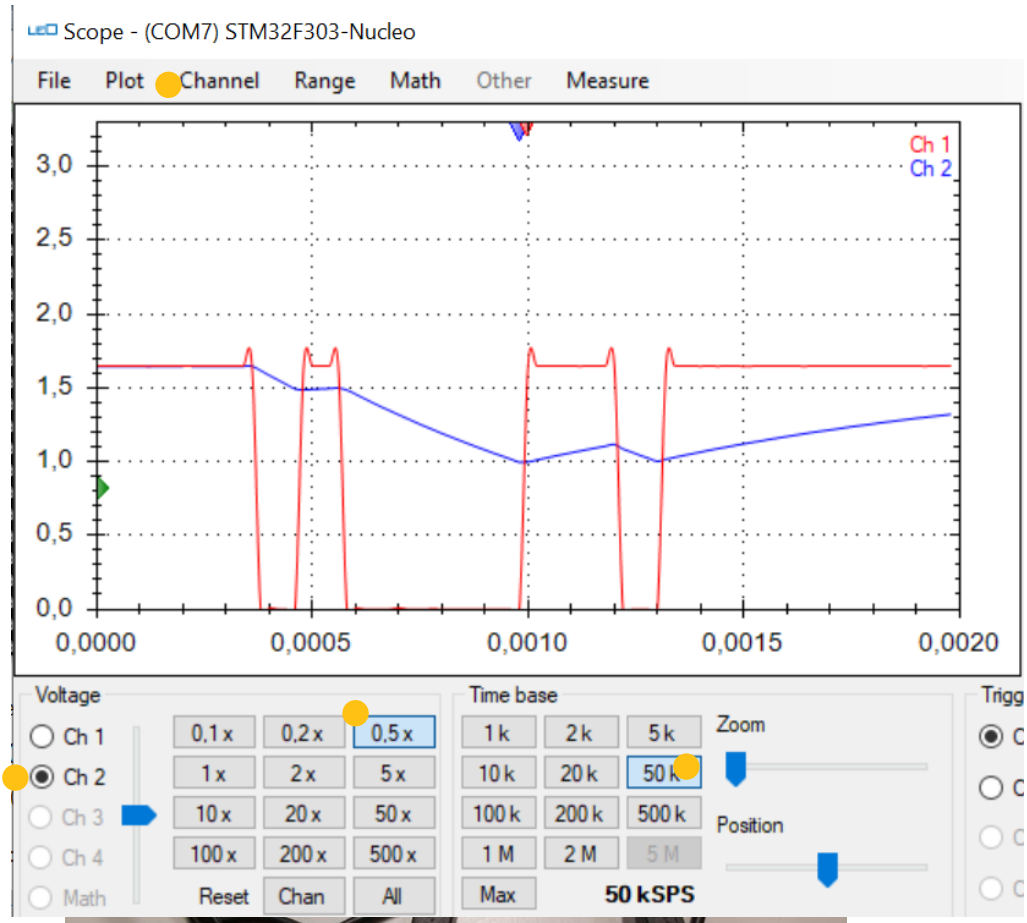
Fügen Sie hier einen Screenshot des LEO ein in der das Bitmuster für das gesendete Zeichen am Eingang U1 und am Ausgang U2 (am Kondensator C) sichtbar ist.

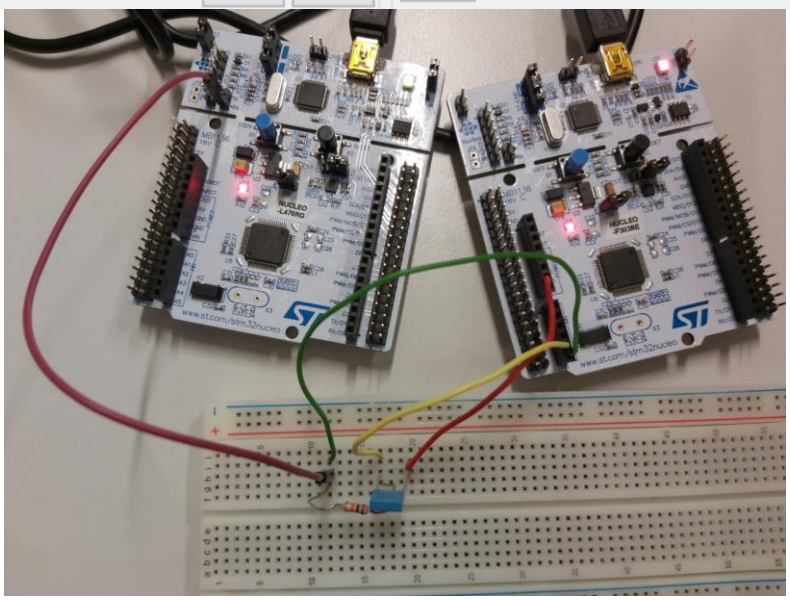


Calculate bit-muster, also pay attention on Least significant bit and most significant one: displayed bits 1111010000110111111 LSB ->MSB

read backwards

Das gesendete Bitmuster in hexadezimaler Schreibweise (ASCII) lautet: 0110 0001 b = 6 1 Hex___**1 Punk**





After adding a capacitor to your circuit, LEO will display the signal below, if you set

-Channel to 2 channels,

-Time base on 50k,

-Voltage to CH 2 and 0.5x

The blue signal is the Capacitor, so we need to bring it to same input as the one originated from the code.

ADDITIONAL EXERCISE:

Konfigurieren Sie nun einen digitalen Input Pin und einen digitalen Output Pin. Der digitale Input Pin soll das Signal nach dem RC-Tiefpass bekommen also an den Ausgang des Tiefpasses angeschlossen werden. Der digitale Output Pin soll den logischen Zustand ausgeben der am Input Pin erfasst/erkannt wird. Fügen Sie hier einen Screenshot des LEO ein in der das Bitmuster für das gesendete Zeichen am Eingang U1 und am Ausgang U2 (am Kondensator C) zeigt. Weiters soll der dritten Kanal das Signal am digitalen Output Pin zeigen.

Realisierung mbed 7 Punkte oder Realisierung Cube&Keil 12 Punkte!

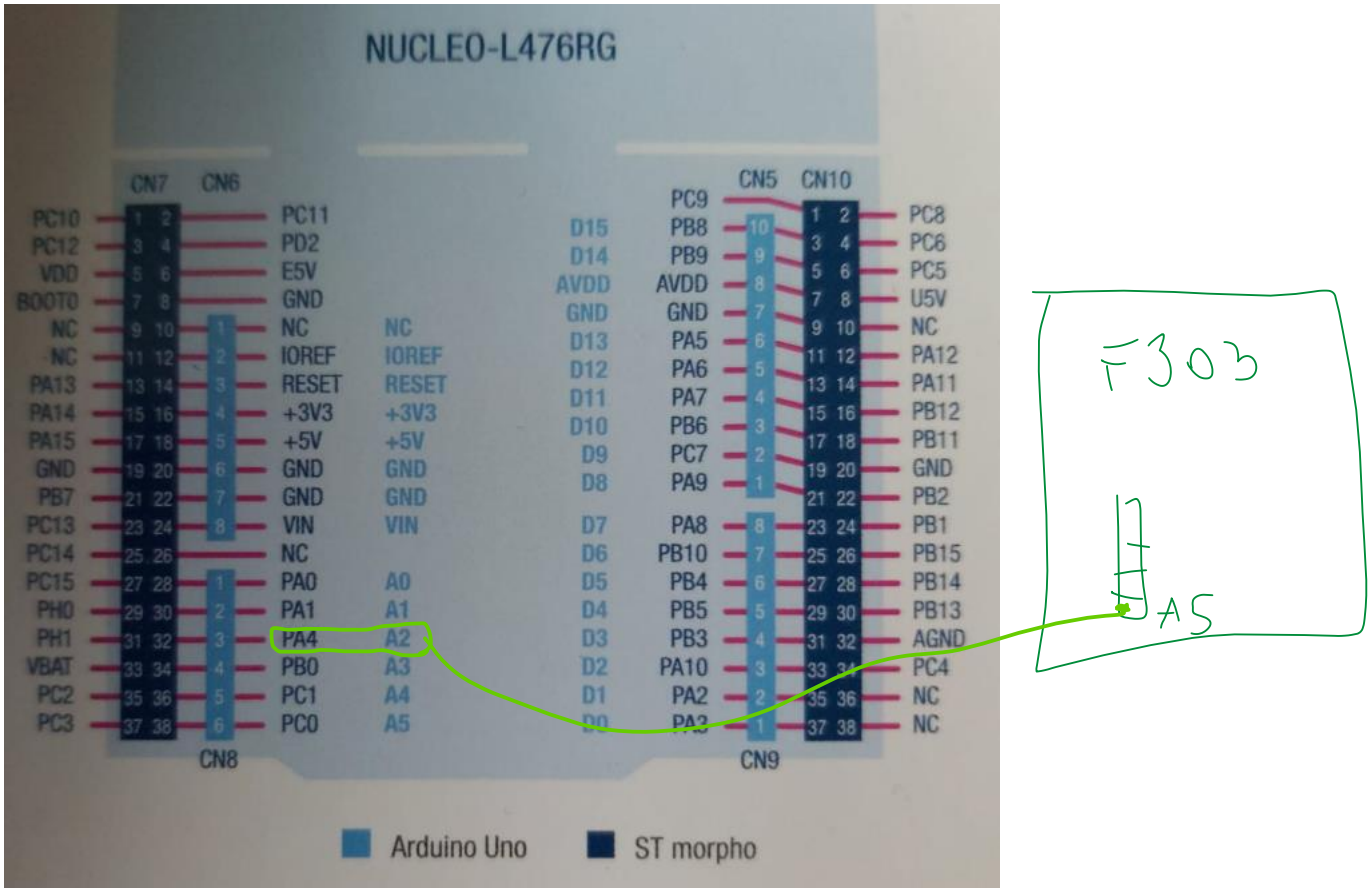
Das am digitalen Output Pin sichtbare Bitmuster in hexadezimaler Schreibweise lautet: _____ 1 Punkt

5) DAC

Erzeugen Sie mit dem DAC des STM32L476 ein Dreiecksignal. ***Realisierung mbed 10 Punkte oder Realisierung Cube&Keil 15 Punkte!***

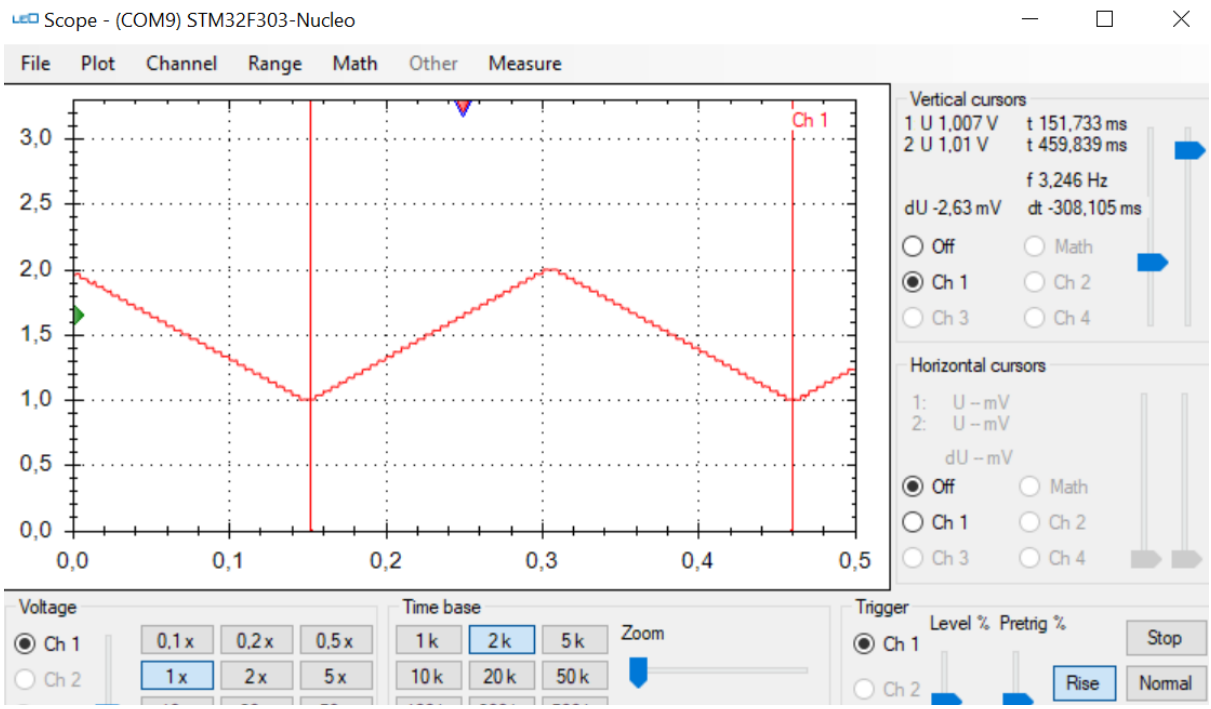
Dieses Dreieck soll aus ___30____ Stufen bestehen.

Und zwischen den Spannungswerten Umin = _____ und Umax = _____ verlaufen.



Code mbed: Triangle_DAC

```
#include "mbed.h"

AnalogOut aout(PA_4); //A2 pin

 // Analoge Werte: 65000 -> 3,3 V; 39393 ->  2 V; 19696  -> 1 V

int main() {

   const double anfangswert = 19696; //umin = 1 V

   const double endwert =  39393; //umax = 2 V

   double abstufung = 30; //die abstufung von der angabe verwenden

   double offset = 300; //wenn die amplitude nicht genau umax trifft , erhöht man einfach die spannungswerte mit dem offset(ausgleich)

   //r = 100000;  //10 kohm

   //7c = 0.000010; //10 mikrofarad

   //tau = r*c;

   //periode = 3 * tau //die formel in der angabe benutzen

   double periode = 5; //in ms , erhöht die dauer einer stufe

   double zaehlwert =((endwert-anfangswert)/abstufung);  //formel um den Schrittwert auszurechnen

   uint16_t sample = 0; //wir brauchen 16 bit um die analoge Spannung darzustellen

    while(1) {

        //für die steigende Sägezahnspannung, anfangswert wird immer um den zählwert erhöht

        for(double i = anfangswert; i < endwert;i = i + zaehlwert) {

            sample = (uint16_t)(i+offset);

            aout.write_u16(sample);

            wait_ms(periode);  }

        //für die sinkende Sägezahnspannung, auskommentieren um nur steigende Sägezahnspannung zu haben

        for(double i = endwert; i > anfangswert;i = i - zaehlwert) {

            sample = (uint16_t)(i+offset);

            aout.write_u16(sample);

            wait_ms(periode);

     }    // zur probe ob die Ausgabe überhaupt funktioniert aout.write_u16(19696);

    }

}
```



t/2= 5*tow -> T = 10*tow -> tow=t/10

and Freq=1/T

Erstellen Sie nun 3 Screenshots für unterschiedliche Frequenzen des Dreiecksignals. Diese 3 Frequenzen ergeben sich aus der Dauer für eine Stufe:

a) T= 3 * T f= _____

b) T = T f= _____

c) T = T / 3 f= _____

Punkt a) ist die „Standardaufgabe“

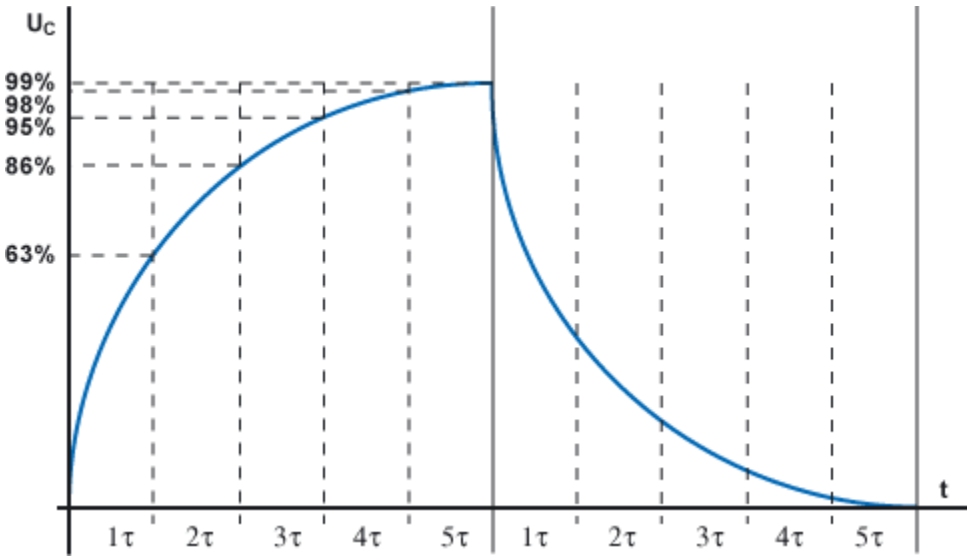
Punkt b) und c) sind alternativen falls Sie die UART nicht realisieren!

Fügen Sie **hier** einen Screenshot des LEO ein in dem das Dreiecksignal am Eingang U1 und die Spannung am Kondensator C – U2 sichtbar sind. Das Bild sollte 1-2 Perioden des Dreiecksignals zeigen! ***Screenshot a) 4 Punkte b) 3 Punkte c) 2 Punkte***

S C R E E N S H O T !

From ONUR RC Tiefpass: in Mbed RC_tiefpass

```
#include "mbed.h"

DigitalOut dout(PC_0);


//Widerstand 5k Zeitkonstante: T = R*C = 5k * 10 mikrofarad = 0,05 s = 50 ms dauert 1 tau ,
// wir wollen aber 5 tau haben , also 5*50 = 250 ms
//T(Periode) = (250*2) 500 ms
// f = 1/500ms = 2 Hz



int main()
{

  float hoch = 250;



  while(1)
  {


    dout = !dout;
    wait_ms(hoch);

    //3,3 - 0,07 = 3,23 , 3,23/10k = 323 mikroampere , r2 = 0,07/ 323 mikroampere

  }


}
//bsp frequenz = 8 khz , Periodendauer = 1/8*10^3 = 12,5 mikros , 12,5 mikros /2 = 62,5 mikro s, wait_us(62,5), wait_ms(0,0625)
```



T(Periode) = (250*2) 500 ms ; f = 1/500ms = 2 Hz

Beispiel 1)

Es soll ein Spannungsteiler entsprechend der Abbildung 1 aufgebaut werden. Von den beiden Widerständen ist ein Widerstand der aus der Übung bekannte 10kOhm Widerstand (ein Ring in oranger Farbe), der zweite Widerstand hat einen im Moment unbekannten Wert!
(entweder 15k, oder 18k, oder 22k, oder 26k)

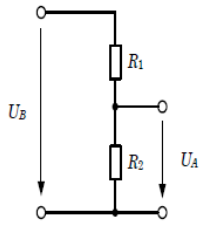


Abbildung1: Unbelasteter Spannungsteiler

Bauen Sie die Schaltung auf, verwenden Sie R1=10kOhm und R2=unbekannt. Verwenden Sie +3,3V vom STM32L476 für die Eingangsspannung UB und Messen Sie Spannung UA mit dem ADC des STM32L476. Geben Sie den gemessenen Wert mit Hilfe der UART und dem virtuellen COM-Port im Terminalprogramm aus.
Realisierung mbed 8 Punkte oder Realisierung Cube&Keil 15 Punkte!
Fügen Sie **hier** einen Screenshot des Terminalfensters ein!

Rechnen Sie den vom ADC gelieferten Zahlenwert in eine Spannung im Bereich 0-3,3V um und geben Sie diesen Wert mit Hilfe der UART und dem virtuellen COM-Port im Terminalprogramm aus.
Realisierung mbed 4 Punkte oder Realisierung Cube&Keil 10 Punkte!
Fügen Sie **hier** einen Screenshot des Terminalfensters ein!

Beim ersten Beispiel baut ihr nur die Schaltung auf und gebt folgenden code (ohne formatierung auf die spannung mit 3300) aus und macht ein screenshot. (Einfach ein screenshot vom putty so wie sie ist machen)

Beim zweiten Beispiel gebt ihr exakt den folgenden Code ein:

```
#include "mbed.h"
```

```
DigitalOut dout(PC_0);

AnalogIn din(PA_0);

Serial pc(SERIAL_TX,SERIAL_RX);


int main()
{

    dout = 1;
    float meas;
    while(1)
    {

        meas = din.read();
        meas = (meas * 3300)/1000;

        pc.printf("measure = %0.2f V\n",meas);
        wait_ms(500);

        //3,3 - 0,07 = 3,23 , 3,23/10k = 323 mikroampere , r2 = 0,07/ 323 mikroampere

    }


}
```

2) Spannungsteiler - LEO

Verwenden Sie nun die selbe Schaltung wie unter Punkt 1) , allerdings verwenden Sie nun den 10kOhm Widerstand als R2 und den nicht bekannten Widerstand als R1.
Erzeugen Sie als Eingangsspannung UB ein Rechtecksignal mit dem STM32L476. Dieses Signal soll eine Frequenz von f= _____ kHz haben.
Realisierung mbed 3 Punkte oder Realisierung Cube&Keil 5 Punkte!

Messen Sie die Ausgangsspannungen U_A mit Hilfe des Little Embedded Oszilloskop – LEO am STM32F303.
Berechnen Sie aus der gemessenen Spannung den Strom und den Gesamtwiderstand der Schaltung!
Gemessene und berechnete Werte je 2 Punkte (3*2=6 Punkte)

I = UA/10kOhm =

Rges= UB/I =

R1= Rges-R2=

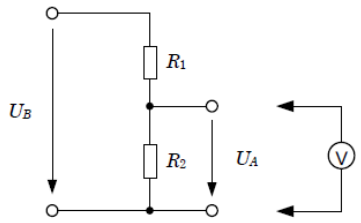
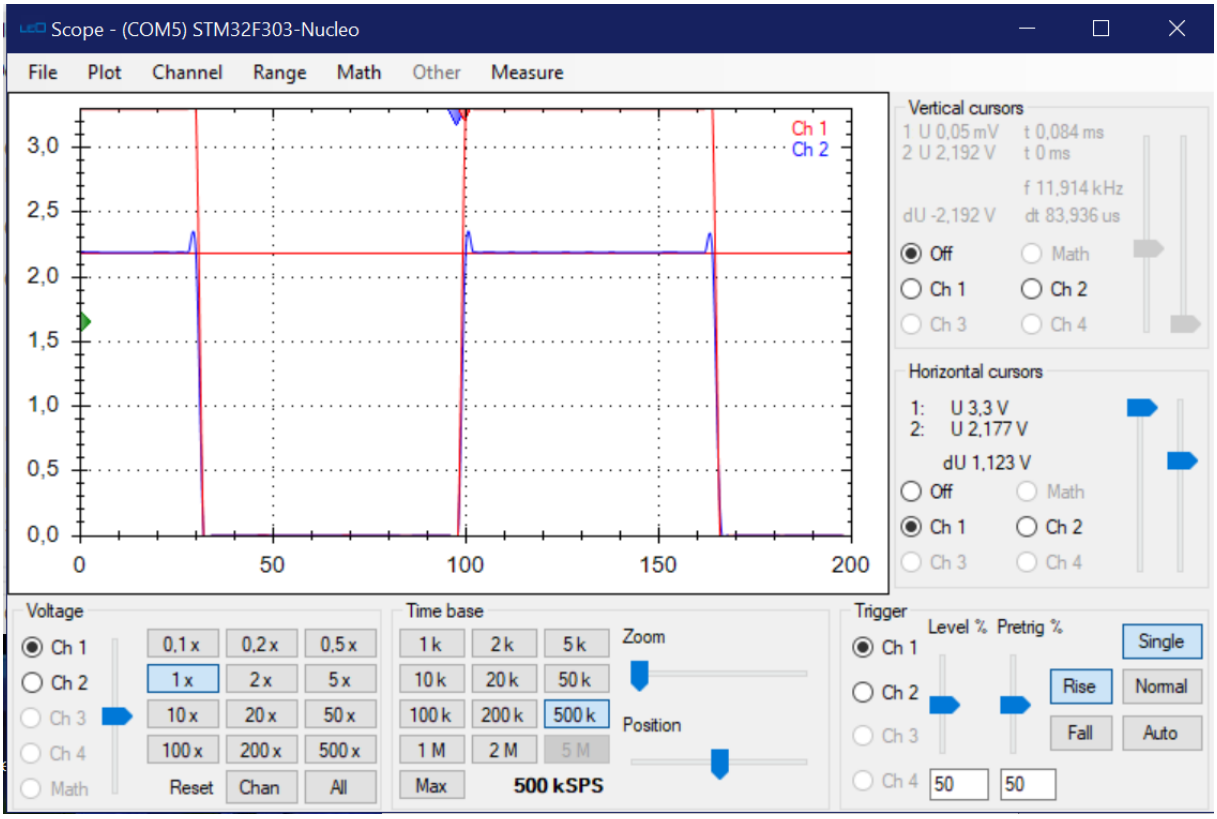


Abbildung 2: Messung an einem unbelasteten Spannungsteiler

Bei diesem Beispiel gebt ihr folgenden Code ein:

```
#include "mbed.h"

DigitalOut dout(PC_0);

int main()
{
   float frequenz = 8000;
   float hoch = (1/frequenz)/2;


   float meas;
   while(1)
   {


      dout = !dout;
      wait(hoch);

      //3,3 - 0,07 = 3,23 , 3,23/10k = 323 mikroampere , r2 = 0,07/ 323 mikroampere
```

```
    }


}
//bsp frequenz = 8 khz , Periodendauer = 1/8*10^3 = 12,5 mikros , 12,5 mikros /2 = 62,5 mikro s, wait_us(62,5), wait_ms(0,0625)
```



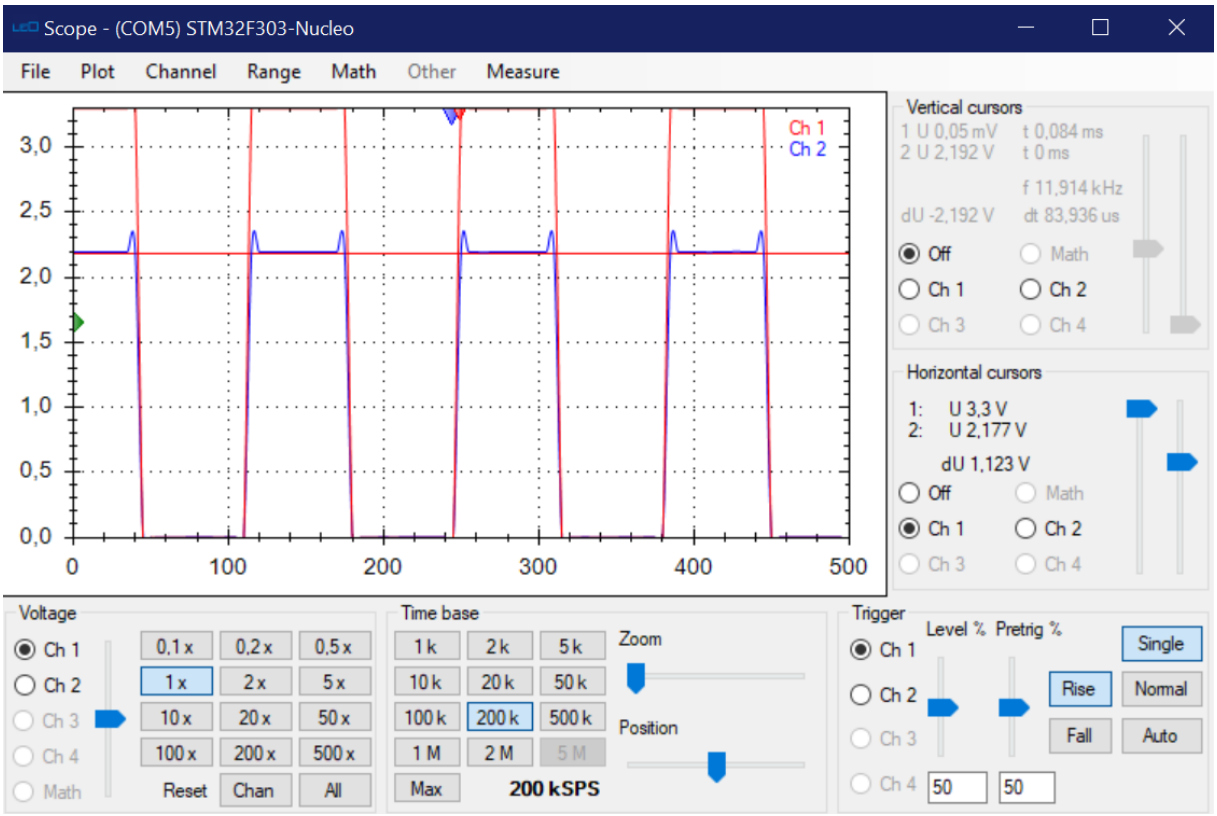
Übungg 2 Berechnungen (siehe bild bei zweitem u 2,1) (r1 = 5100 ohm , r2 = 10k)
U1 = 1,1
U2 = 2,1
Ug = 3,3
Frequenz 8 khz

I = 2,1 / 10 k = 110 mikroampere
Rges = 3,3/ 210 mikroampere = 15,7 k

R1 =15,7– 10 = 5,7 k

Fügen Sie **hier** einen Screenshot des LEO ein in dem das Rechtecksignal UB und die Spannung am Widerstand R2 – UA sichtbar sind. Das Bild sollte 2-4 Perioden des Rechtecksignals zeigen!

Screenshot 4 Punkte



3)

3) RC-Tiefpass

Bauen Sie nun einen RC-Tiefpass auf.
Verwenden Sie dazu jenen Widerstand dessen Wert Sie soeben ermittelt haben!

Verwenden Sie einen Kondensator mit der Kapazität C= ___10 nF_____

Berechnen Sie die Zeitkonstante: τ = _____

Bauteilwerte:

R =

C = 10nF

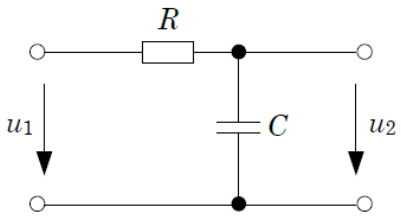


Abbildung 1: Schaltung des RC-Gliedes

Erzeugen Sie dieses Rechtecksignal mit dem STM32L476.

Berechnen Sie die Frequenz für dieses Rechtecksignal damit der Kondensator sich jeweils voll auf- und entladen kann! (T/2 = 5 τ)

T= _____.

f= _____.

Berechnete Werte je 2 Punkte (2*2=4 Punkte)

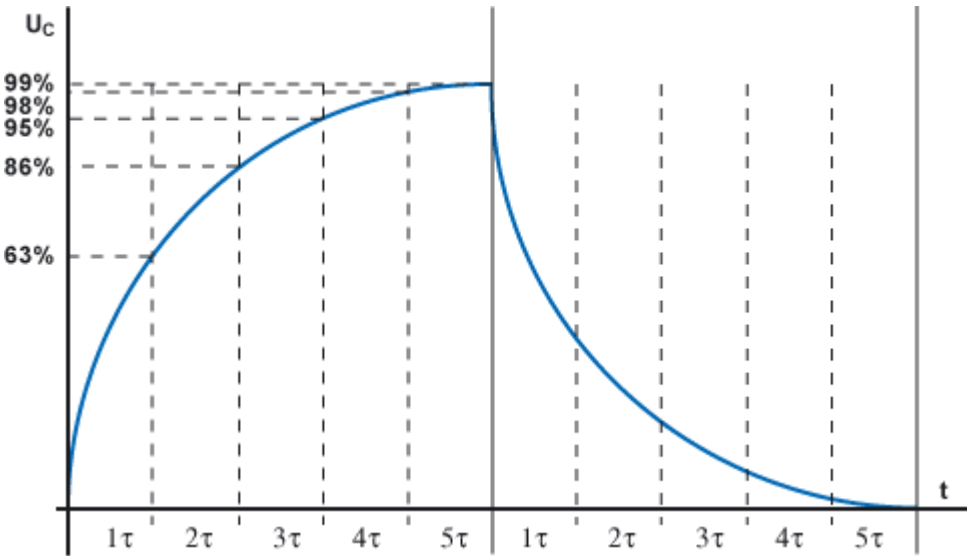
Realisierung mbed 1 Punkt oder Realisierung Cube&Keil 1 Punkt!

Kondensator : 10 mikrofarad
Widerstand 5k
Zeitkonstante: T = R*C = 5k * 10 mikrofarad = 0,05 s = 50 ms dauert 1 tau , wir wollen aber 5 tau haben , also 5*50 = 250 ms

Code:

```
#include "mbed.h"

DigitalOut dout(PC_0);


//Widerstand 5k Zeitkonstante: T = R*C = 5k * 10 mikrofarad = 0,05 s = 50 ms dauert 1 tau ,
// wir wollen aber 5 tau haben , also 5*50 = 250 ms
//T(Periode) = (250*2) 500 ms
// f = 1/500ms = 2 Hz


int main()
```

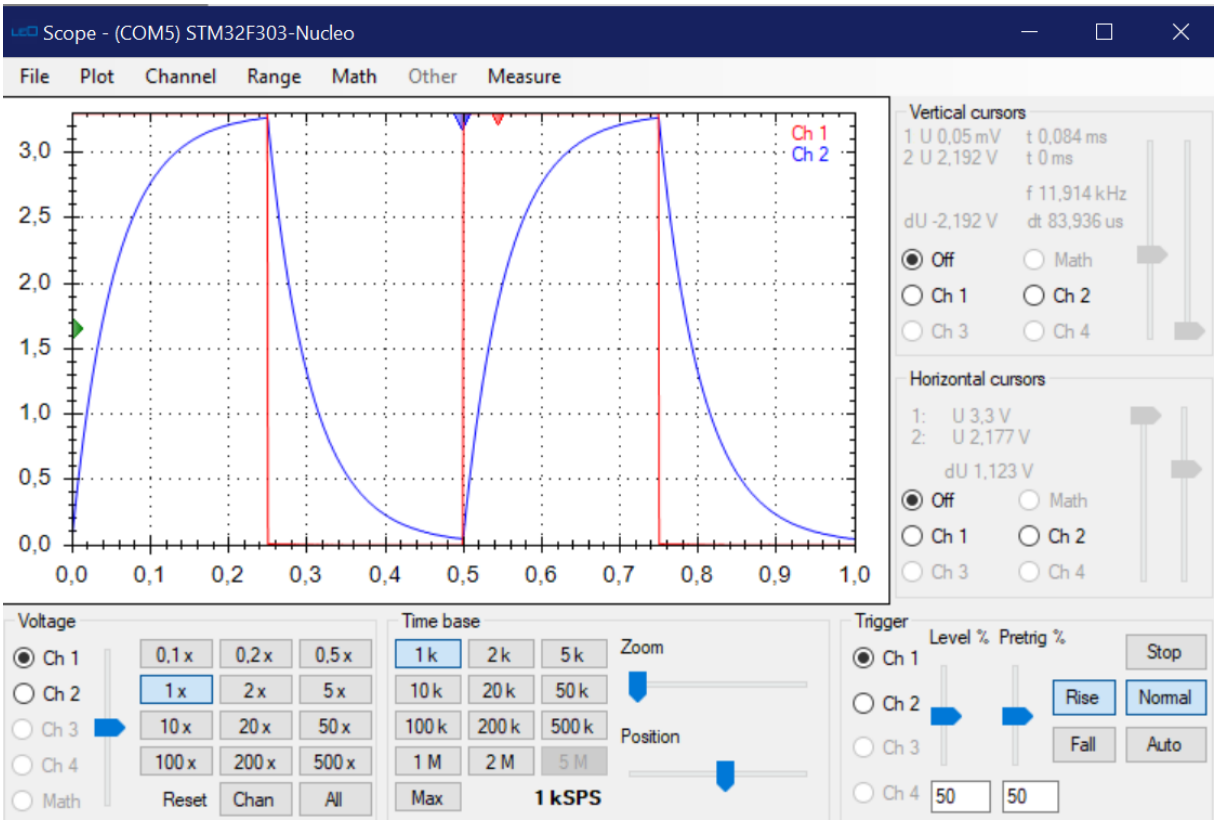
```
{
    float hoch = 250;



    while(1)
    {

        dout = !dout;
        wait_ms(hoch);

        //3,3 - 0,07 = 3,23 , 3,23/10k = 323 mikroampere , r2 = 0,07/ 323 mikroampere

    }


}
//bsp frequenz = 8 khz , Periodendauer = 1/8*10^3 = 12,5 mikros , 12,5 mikros /2 = 62,5
mikro s, wait_us(62,5), wait_ms(0,0625)
```



T(Periode) = (250*2) 500 ms
f = 1/500ms = 2 Hz

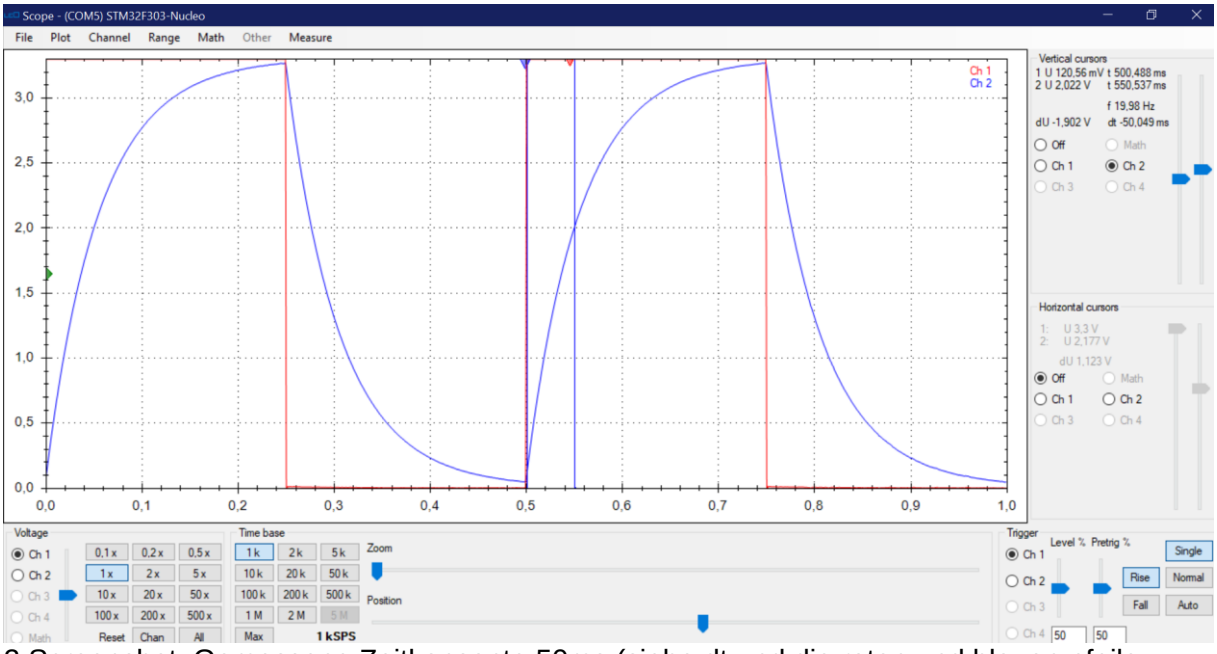
Fügen Sie **hier** einen Screenshot des LEO ein in dem das Rechtecksignal am Eingang U1 und die Spannung am Kondensator C – U2 sichtbar sind. Das Bild sollte 1-2 Perioden des Rechtecksignals zeigen!
Screenshot 4 Punkte

erster screenshot



Fügen Sie **hier** einen Screenshot des LEO ein in dem das Rechtecksignal am Eingang U1 und die Spannung am Kondensator C – U2 sichtbar sind. Fügen Sie nun die Cursor hinzu und messen Sie die Zeitkonstante. Das Bild sollte nicht mehr als 1 Periode des Rechtecksignals zeigen!

Gemessenen Zeitkonstante: τ = _____
Screenshot 4 Punkte



2.Screenshot, Gemessene Zeitkonsante 50ms (siehe dt und die roten und blauen pfeile oben)

4)

4) UART

Erzeugen Sie mit der UART des STM32L476 ein Signal.

Senden Sie das Zeichen _____

Mit einer Datenrate: _____

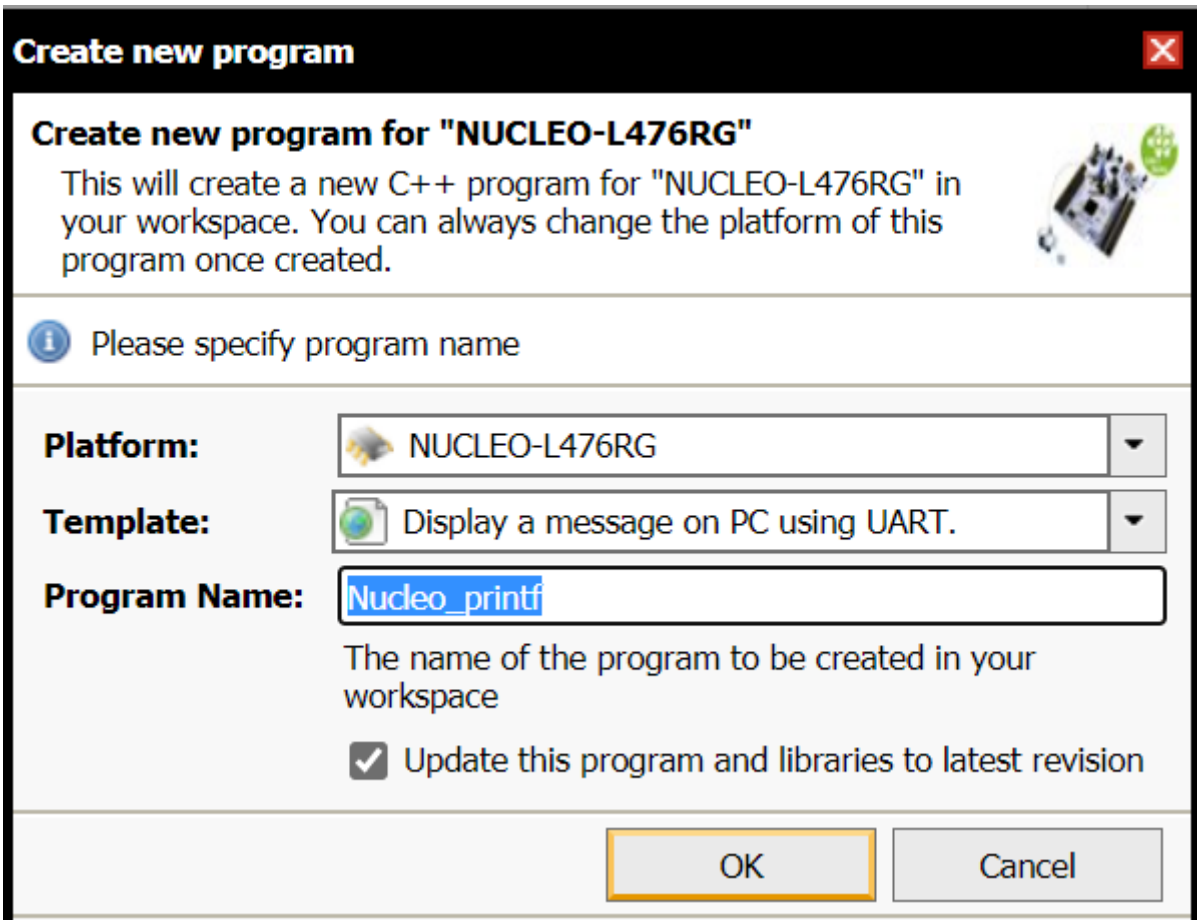
Realisierung mbed 3 Punkte oder Realisierung Cube&Keil 6 Punkte!

Fügen Sie **hier** einen Screenshot des LEO ein in der das Bitmuster für das gesendete Zeichen am Eingang U1 und am Ausgang U2 (am Kondensator C) sichtbar ist.

Das gesendete Bitmuster in hexadezimaler Schreibweise (ASCII) lautet:

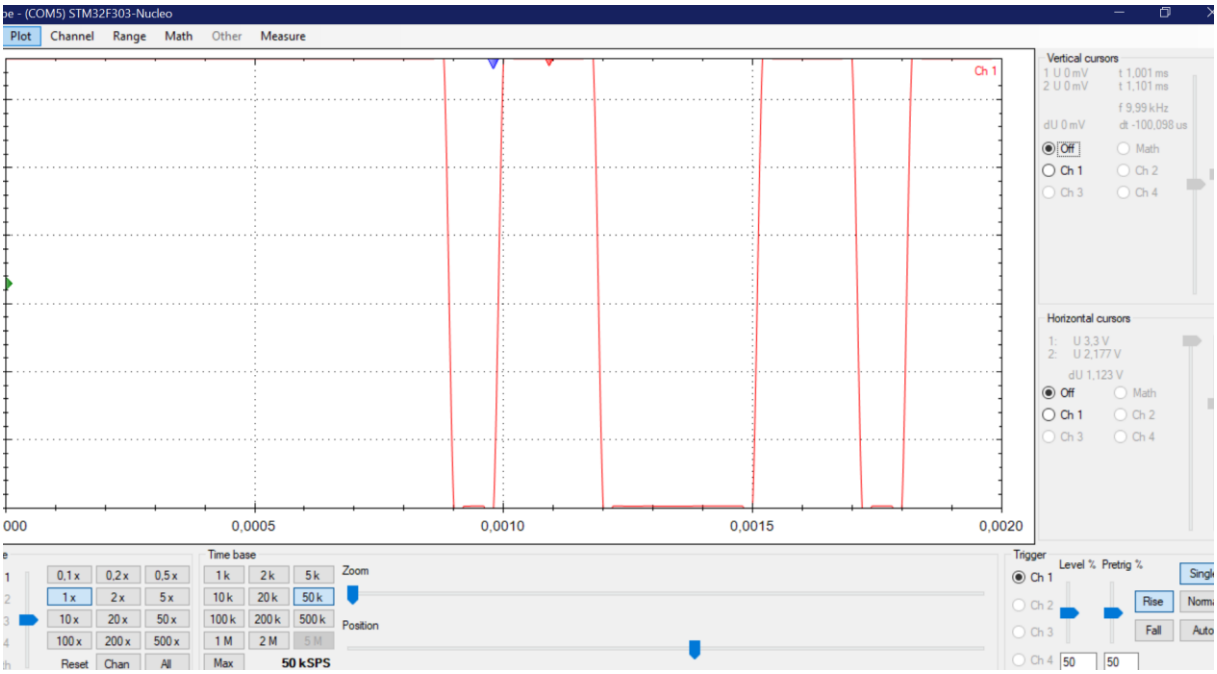
_____ ***1 Punkt***

Zeichen ausgeben



Zeichen: c
Datenrate: 9600
(Kondensatorschaltung beibehalten

```
#include "mbed.h"

//-----------------------------------
// Hyperterminal configuration
// 9600 bauds, 8-bit data, no parity
//-----------------------------------

Serial pc(SERIAL_TX, SERIAL_RX);
DigitalIn din(PC_0);
DigitalOut dout(PA_1);

int main()
{
    pc.baud(9600);
    while(1)
    {
        pc.printf("d");
        dout = din;
        wait_ms(250);


    }
}
```

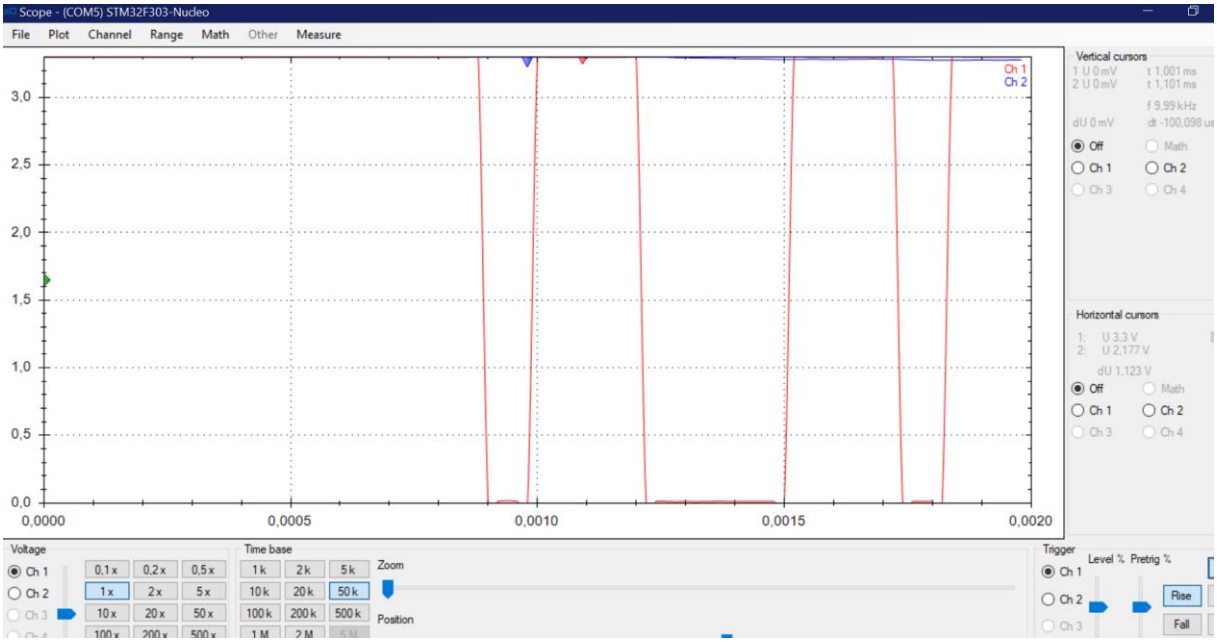


0 startbit

1100 0110

0110 0011

0x63



Da wir einen Kondensator haben der eine zu hohe kapazität hat , können wir channel 2 nicht sehen. Beim Test sieht man es besser.

Konfigurieren Sie nun einen digitalen Input Pin und einen digitalen Output Pin.
Der digitale Input Pin soll das Signal nach dem RC-Tiefpass bekommen also an den Ausgang des Tiefpasses angeschlossen werden.
Der digitale Output Pin soll den logischen Zustand ausgeben der am Input Pin erfasst/erkannt wird.
Fügen Sie **hier** einen Screenshot des LEO ein in der das Bitmuster für das gesendete Zeichen am Eingang U1 und am Ausgang U2 (am Kondensator C) zeigt.
Weiters soll der dritten Kanal das Signal am digitalen Output Pin zeigen.

Realisierung mbed 7 Punkte oder Realisierung Cube&Keil 12 Punkte!

Das am digitalen Output Pin sichtbare Bitmuster in hexadezimaler Schreibweise lautet:

_____ ***1 Punkt***

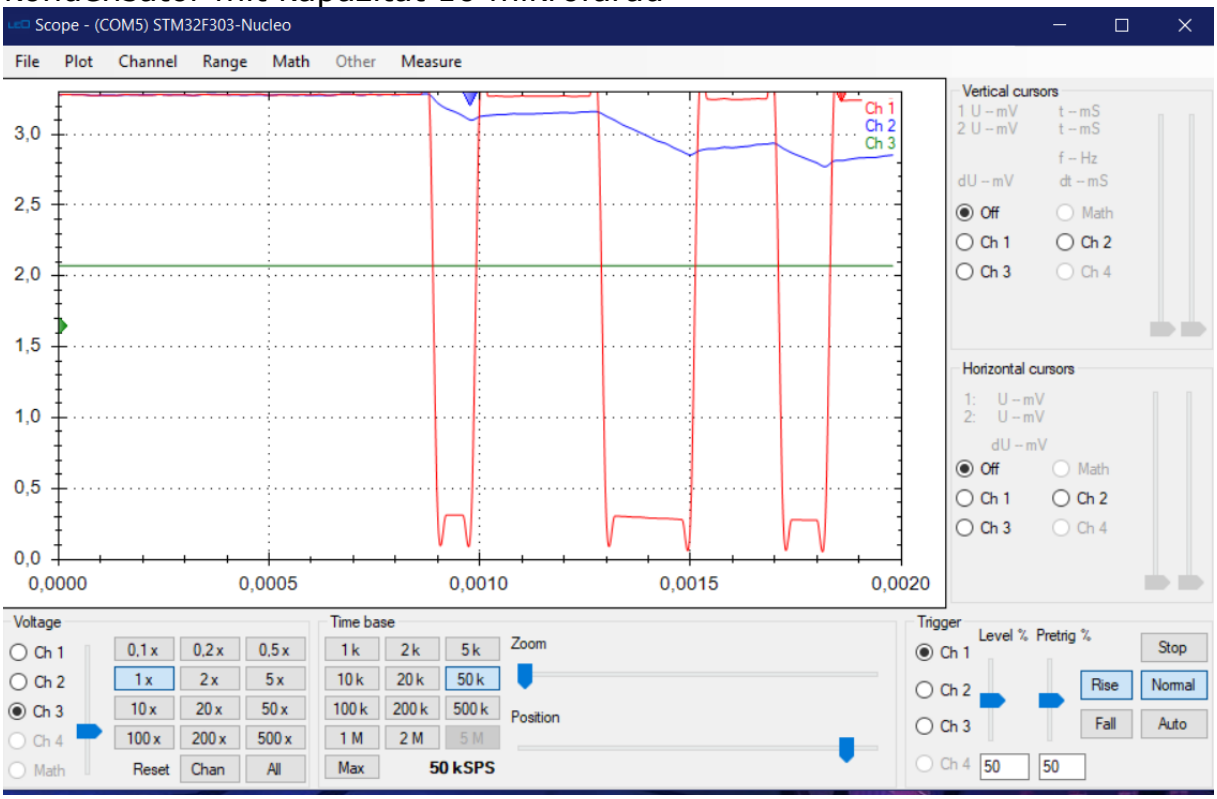
Rx → eingang → ch1

Input pin(bei mir pc_0) zwischen kondensator und widerstand → ch2

Output pin(bei mir pa_0) zu ch3

Zu allerletzt ground

Da hab ich einen widerstand von 220 ohm genommen und einen kondensator mit kapazität 10 mikrofarad



```
#include "mbed.h"

//---------------------------------
// Hyperterminal configuration
// 9600 bauds, 8-bit data, no parity
//---------------------------------

Serial pc(SERIAL_TX, SERIAL_RX);
DigitalIn din(PC_0);
DigitalOut dout(PA_0);
DigitalOut UserLED(PA_5);

int main()
{
   pc.baud(9600);
   pc.printf("started");
   while(1)
   {
     /* char d = pc.getc();
      pc.putc(d);
      if(d == 'a') {
          UserLED = !UserLED; //Sollte so passen
          //}
     */
          dout = din;



   }
}
```

Achtung ! Der dritte Channel zeigt konstant 3,3 v an, da der Kondensator zu hoch ist und sich nie so tief entlädt , dass er den Schmitt Trigger schaltet, beim Test müsst ihr ein Rechtecksignal sehen