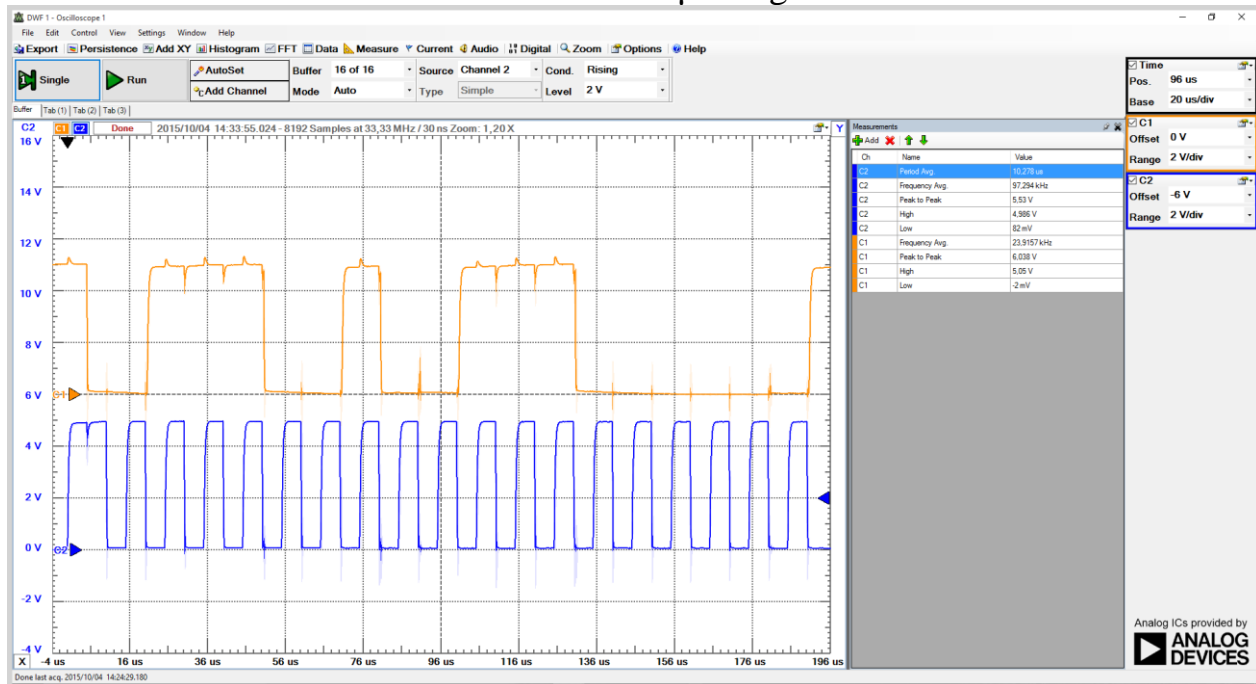




## Messungen am Bus mit dem Oszilloskop: Ein IIC Baustein PCF8574a wird von einem $\mu$ C angesteuert.



### Aufgaben:

Bestimmen sie aus dem dargestellten Oszilloskopbild:

1. Frequenz vom Taktsignal
2. Markieren sie im Datensignal
  - IIC Startbedingung
  - Adressbits (8bit)
  - Acknowledge
  - Datenbyte (8bit)
  - Acknowledge
  - Read/Write bit
3. Wie lautet die Adresse für den Baustein PCF8574a? Angabe binär und Hex
4. Wie lautet das übertragene Byte? Angabe binär und Hex
5. Wer sendet bei dieser Übertragung das Acknowledge?

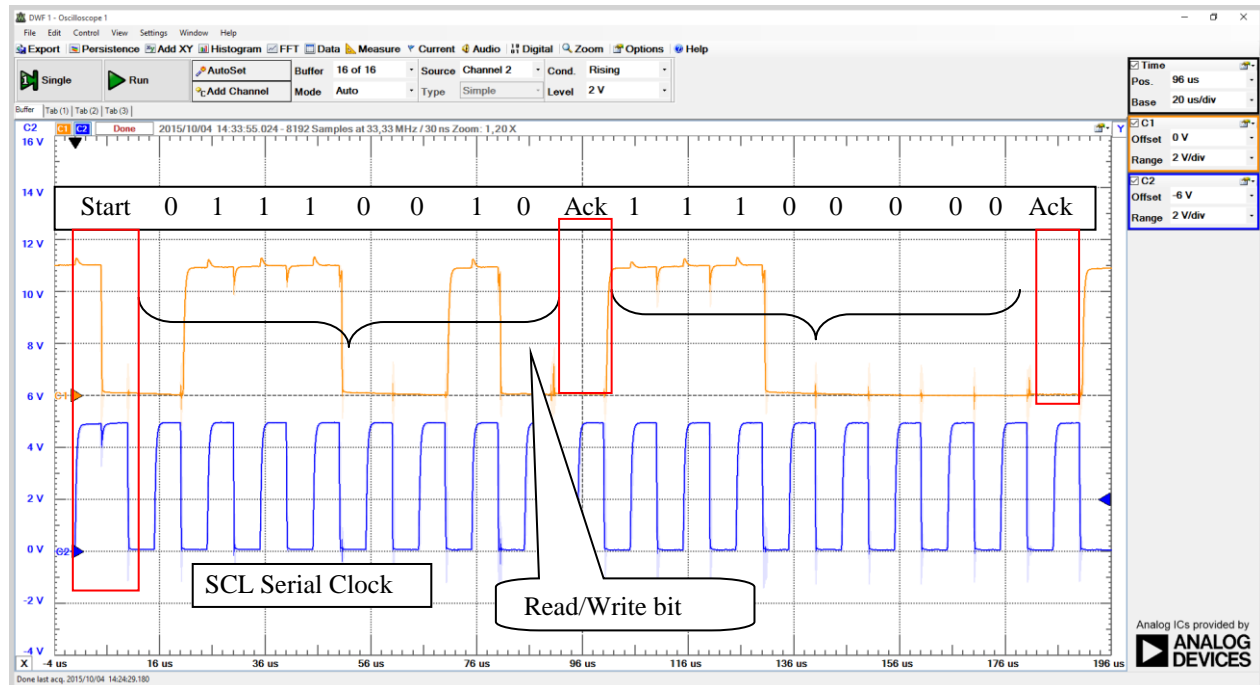
### Zusatzfragen: Datenblatt PCF8574/8574a notwendig!

6. Welche Pegel (0 /5V) haben die drei Adressbits am PCF8574a?
7. Wenn ein Baustein PCF8574 !ohne Zusatz a! verwendet werden würde, wie lautet dann die Adresse?
8. Nennen sie die 8 möglichen Adressen für den Baustein PCF8574a und PCF8574. Als letztes Adressbit wird immer eine 0 ausgegeben ->Der Baustein empfängt Daten vom Master, (Master sendet ->write)

## Lösungen

1. Frequenz Taktsignal: 100 kHz

2.



3. 0111 0010    0x72    letztes bit RD/WR = 0 somit write aktiv -> Master sendet

4. 1110 0000    0xE0

5. Das Acknowledge-bit (low aktiv) liefert der IIC Baustein PCF8574a



### Zusatz:

Bild mit Start, Adresse 0X72 (8bit) Ackn Datenbyte 0XFF Ackn

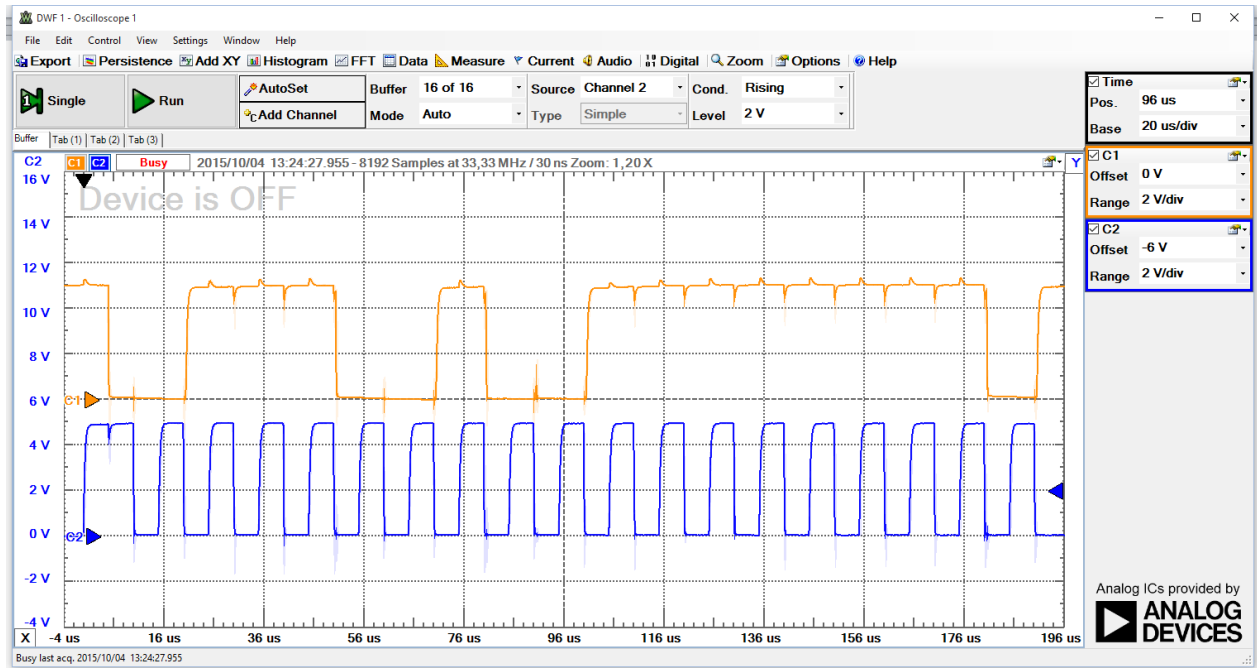
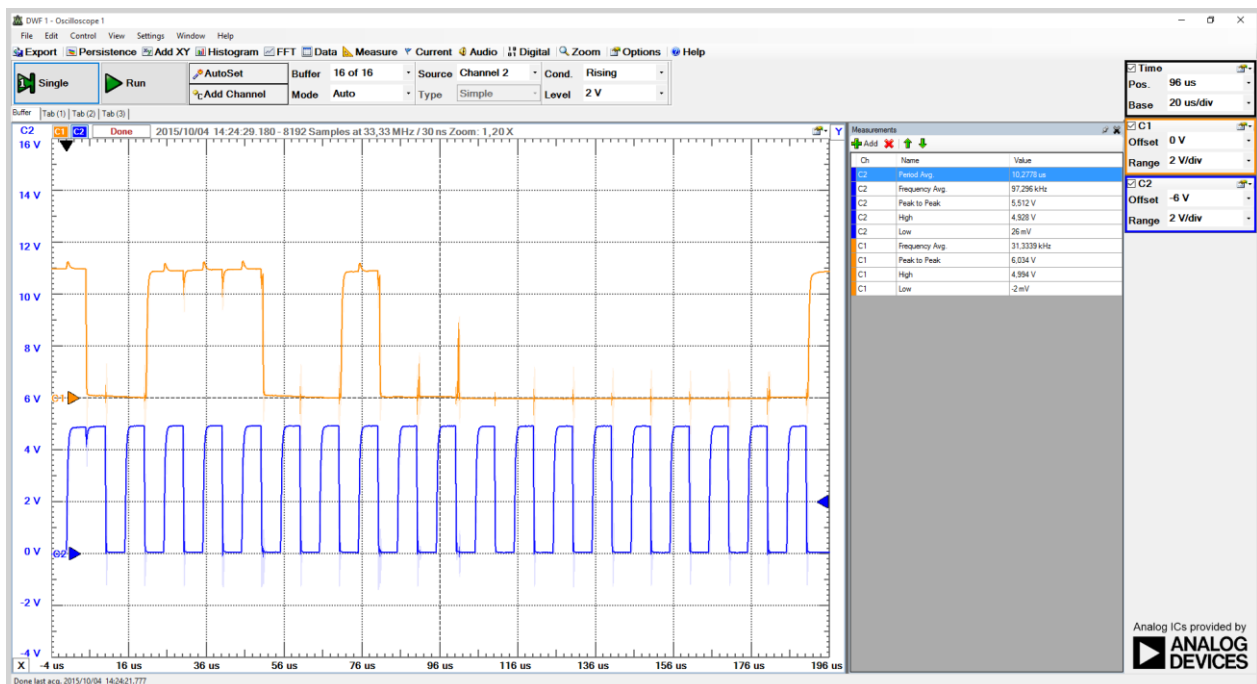


Bild mit Start, Adresse 0X72 (8bit) Ackn Datenbyte 0X00 Ackn





## Quellcode für ARM Nucleo F103RB

```
/*Data send to IIC-board with PCF8574a IC
Data coming from analog-value Pin A0 = PA_0 Nucleo-board
you see a 8bit light strip as a result
Aug. 2015 G. Neumaier Gewerblich-Technische Schule Offenburg Germany
*/

#include "mbed.h"
#define PCF8574a_ADDR 0x72 // PCF8574a address 0x72IIC-board Birk

I2C i2c(I2C_SDA, I2C_SCL); //IIC Pins SDA and SCL for ARDUINO pinheader Nucleo-board
//SCL and SPA pullup resistors 2k2Ohm to+5Volt
DigitalOut myled(LED1); //LED on Nucleo-board
AnalogIn analog_value0(A0); //same as Pin PA_0
Serial pc(SERIAL_TX, SERIAL_RX);

int main()
{
    float mess0;
    unsigned char mess1, x, LEDs8;
    char data_write[2]; //must be char!!
    // char data_read[2]; //read buffer

    unsigned char speicher[9] = {0xff, 0x7f, 0x3f, 0x1f, 0x0f, 0x07, 0x03, 0x01, 0x00}; //data for light strip
    data_write[0] = 0xf2; //LEDs low aktiv dummy
    int status = i2c.write(PCF8574a_ADDR, data_write, 1, 0);
    if (status != 0) // Error no acknowledge detected
    {
        while (1) //-> endless loop when error no IIC-IC detected
        {
            myled = !myled;
            wait(0.7);
        }
    }

    while (1) //endless loop
    {
        mess0 = analog_value0.read(); // Converts and read the analog input value (value from 0.0 to 1.0)
        mess0 = mess0 * 255; // Change the value to be in the 0 to 255 range -> 8bit
        mess1 = (char) mess0; //type converting from float to char(8bit)
        printf("Analogvalue0 = %.1f \r\n", mess0); //Rs232 output-> You will see this in terminalprogram on
PC
        //%.1f -> display 1 digit after comma
        //
        x= mess1/29; //Remainder of the division falls away, only whole numbers
        LEDs8= speicher[x]; //brings accordingly x values aus dem array "speicher"
        data_write[0] = ~LEDs8; //8bit analog-value ~ inverted leds lowaktiv!!
        i2c.write(PCF8574a_ADDR, data_write, 1, 1); // no stop
        //you see a digital 8bit led-strip at leds contacted to IC PCF8574a
    }
}
}
```