

Hello,

I'm wondering about the transfer rate of USB FS and sample rate with ADC.

I have written a program which measures continuous two 50Hz sinus signals. Then it transfers them to the PC and shows the result with labview.

The problem is, that it seems to be under sampled. Now I am searching for the bottle neck.

Here are the key parts of the firmware:

```
// When a 'B' is received one sample (two bytes for each channel) is send
static int8_t CDC_Receive_FS (uint8_t* Buf, uint32_t *Len)
{
    /* USER CODE BEGIN 6 */

    switch(Buf[0]){
        case 'B':

            CDC_Transmit_FS(ADC_buf,4);
            break;
    }
    USBDCDC_ReceivePacket(&hUsbDeviceFS);

    return (USBD_OK);
    /* USER CODE END 6 */
}

// data is put in the buffer
void convert_ADC_samples(void){

    ADC_buf[0] = (Filter1 >> 8) & 0xFF;
    ADC_buf[1] = Filter1 & 0xFF;
    ADC_buf[2] = (Filter2 >> 8) & 0xFF;
    ADC_buf[3] = Filter2 & 0xFF;

}

// ADC configuration
void MX_ADC1_Init(void)
{
    ADC_ChannelConfTypeDef sConfig;

    /**Common config
    */
    hadc1.Instance = ADC1;
    hadc1.Init.ClockPrescaler = ADC_CLOCK_ASYNC;
    hadc1.Init.Resolution = ADC_RESOLUTION12b;
    hadc1.Init.ScanConvMode = ADC_SCAN_ENABLE;
    hadc1.Init.ContinuousConvMode = ENABLE;
    hadc1.Init.DiscontinuousConvMode = DISABLE;
    hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
    hadc1.Init.NbrOfConversion = 2;
    hadc1.Init.DMAContinuousRequests = ENABLE;
    hadc1.Init.EOCSelection = EOC_SINGLE_CONV;
    hadc1.Init.LowPowerAutoWait = DISABLE;
    hadc1.Init.Overrun = OVR_DATA_OVERWRITTEN;
    HAL_ADC_Init(&hadc1);
}
```

```

    /**Configure Regular Channel
    */
    sConfig.Channel = ADC_CHANNEL_2;
    sConfig.Rank = 1;
    sConfig.SingleDiff = ADC_SINGLE_ENDED;
    sConfig.SamplingTime = ADC_SAMPLETIME_1CYCLE_5;
    sConfig.OffsetNumber = ADC_OFFSET_NONE;
    sConfig.Offset = 0;
    HAL_ADC_ConfigChannel(&hadc1, &sConfig);

    /**Configure Regular Channel
    */
    sConfig.Channel = ADC_CHANNEL_3;
    sConfig.Rank = 2;
    HAL_ADC_ConfigChannel(&hadc1, &sConfig);
}

```

This gets me a result like this (as you can see it looks like some values are missing):
Please have a look at 1samples.png

I thought it might be a problem of small data packets, because I always need to wait for the 'B' and then answer with only one sample. So I changed the software to send packets with 5 samples:

```

// 5 samples (each is made of two bytes), two channels => 20 Bytes at all

static int8_t CDC_Receive_FS (uint8_t* Buf, uint32_t *Len)
{
    /* USER CODE BEGIN 6 */

    switch(Buf[0]){
        case 'B':

            CDC_Transmit_FS(ADC_buf,20);
            break;
    }
    USBD_CDC_ReceivePacket(&hUsbDeviceFS);

    return (USB_D_OK);
    /* USER CODE END 6 */
}

// this function is called non stop in main ()
void convert_ADC_samples(void){

    uint8_t array_margin = sample_count *4;

    ADC_buf[0 + array_margin] = (Filter1 >> 8) & 0xFF;
    ADC_buf[1 + array_margin] = Filter1 & 0xFF;
    ADC_buf[2 + array_margin] = (Filter2 >> 8) & 0xFF;
    ADC_buf[3 + array_margin] = Filter2 & 0xFF;

    sample_count++;
    if(sample_count == 5){
        sample_count = 0;
    }
}
}

```

the result is this:
Please have a look at 5samples.png

Ok, there might be a problem on the PC (labview) side, but it also could be the μ C program. The way I'm organizing the data could be optimized. Or what is your opinion?