

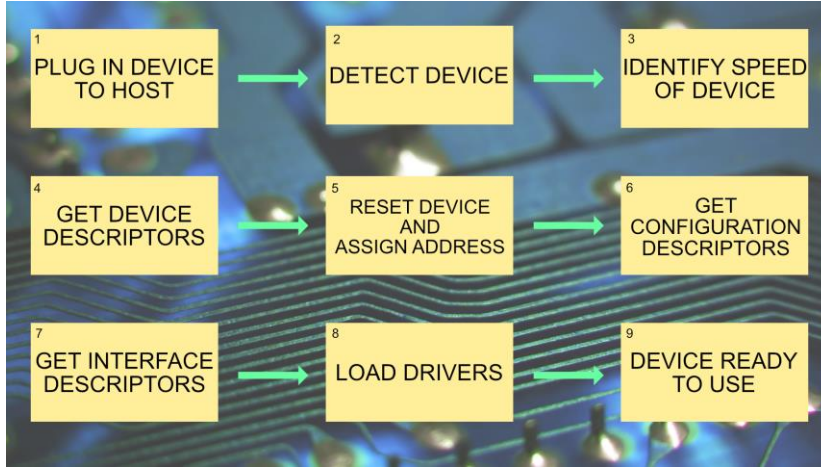
1-)Proje Tanımı

Bu projede STM32F4 Discovery geliştirme kartı USB üzerinden host cihaza(PC) mikrofon olarak tanıtıldı ve kart içerisinde kayıtlı sinüs sinyali USB üzerinden bilgisayara gönderildi. Aktarılan sinyal Audacity programında görüntülendi.

2-)Proje Adımları

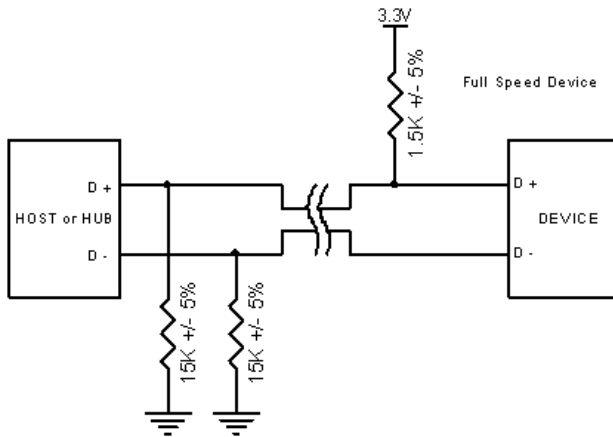
2.1-)Kartın Bilgisayara Mikrofon Olarak Tanıtılması

Bir USB device host cihaza bağlandığında ilk olarak enumeration işlemi gerçekleşir. Enumeration, host cihazın kendisine bağlanan device' ı tanıyıp bu cihazla etkileşiminde kullanacağı uygun driver' ı seçmesi işlemidir.



Görsel-1 USB Enumeration Adımları

Host cihaz D+ ve D- pinlerinin durumunu izleyerek kendisine bir device bağlandığını ve bağlanan device' ın haberleşme hızını tespit edebilir. Host cihazın D+ ve D- pinlerine 15k pull down dirençleri bağlıdır.



Görsel-2 USB FS Bağlantı Şeması

Host cihaz, device' ın D- pininde pull-up direnci bağlıysa kendisine USB LS device(1.5 Mbps) bağlandığını, device' ın D+ pininde pull-up direnci bağlıysa kendisine USB FS device(12Mbps) veya USB HS device bağlandığını tespit eder.

Cihazın bağlandığı tespit edilip haberleşme hızı belirlendikten sonra host cihaz artık device descriptor bilgisini sorabilir.

```
71 __ALIGN_BEGIN uint8_t hUSBDDeviceDesc[USB_LEN_DEV_DESC] __ALIGN_END =
72 {
73     0x12,                /* bLength */
74     USB_DESC_TYPE_DEVICE, /* bDescriptorType (which descriptor) */
75     0x00,                /* bcdUSB (USB Specification: USB 2.0) */
76     0x02,
77     0x00,                /* bDeviceClass (0x00:class info in Interface Descriptor) */
78     0x00,                /* bDeviceSubClass (in Interface Descriptor) */
79     0x00,                /* bDeviceProtocol (in Interface Descriptor) */
80     USB_MAX_EP0_SIZE,    /* bMaxPacketSize (for endpoint 0) */
81     LOBYTE(USBD_VID),    /* idVendor (vendor id) */
82     HIBYTE(USBD_VID),    /* idVendor */
83     LOBYTE(USBD_PID),    /* idVendor (product id) */
84     HIBYTE(USBD_PID),    /* idVendor */
85     0x00,                /* bcdDevice rel. 2.00 (USB Specification: USB 2.0) */
86     0x02,
87     USBD_IDX_MFC_STR,     /* Index of manufacturer string */
88     USBD_IDX_PRODUCT_STR, /* Index of product string */
89     USBD_IDX_SERIAL_STR,  /* Index of serial number string */
90     USBD_MAX_NUM_CONFIGURATION /* bNumConfigurations */
91 }; /* USB_DeviceDescriptor */
```

Görsel-3 USB Device Descriptor

Device descriptor bilgisinin ilk 8 byte' ı alındıktan sonra host tarafından device, D+ ve D- pinleri low' a çekilerek resetlenir.

Reset işleminden sonra device' a host tarafından adres ataması gerçekleştirilir.

Bir device host cihaza ilk bağlandığında otomatik olarak 0 adresini alır ve daha sonra host tarafından eşsiz(unique) bir adres atanır. Host cihaz bu eşsiz adresi hangi device ile haberleşeceğini seçerken kullanır.

Host cihaz daha sonra device descriptor(bu sefer tamamını), configuration descriptor ve interface descriptor bilgilerini sorar ve bu bilgiler ışığında uygun driver' ı yükler.

```
USBD_AUDIO_CfgDesc[0] = 0x09; /* bLength */
USBD_AUDIO_CfgDesc[1] = 0x02; /* bDescriptorType */
USBD_AUDIO_CfgDesc[2] = ((USB_AUDIO_CONFIG_DESC_SIZ+Channels-1)&0xff); /* wTotalLength */
USBD_AUDIO_CfgDesc[3] = ((USB_AUDIO_CONFIG_DESC_SIZ+Channels-1)>>8);
USBD_AUDIO_CfgDesc[4] = 0x02; /* bNumInterfaces (control and streaming) */
USBD_AUDIO_CfgDesc[5] = 0x01; /* bConfigurationValue */
USBD_AUDIO_CfgDesc[6] = 0x00; /* iConfiguration */
USBD_AUDIO_CfgDesc[7] = 0x80; /* bmAttributes BUS Powered */
USBD_AUDIO_CfgDesc[8] = 0x32; /* bMaxPower = 100 mA */
```

Görsel-4 Configuration Descriptor

USB haberleşmede, haberleşmeyi her zaman host cihaz başlatır. Host cihaz setup paket(request) göndererek device' ı configure edebilir(adres atama gibi) ve descriptorlarını sorgulayabilir.

```

typedef struct usb_setup_req
{
    uint8_t    bmRequest;
    uint8_t    bRequest;
    uint16_t   wValue;
    uint16_t   wIndex;
    uint16_t   wLength;
} USBD_SetupReqTypeDef;

```

Görsel-5 Token-Setup(request) Paket Formatı

bmRequestType ile ne üzerinde işlem yapılacağı seçilir. bRequest ile hangi işlemin yapılacağı seçilir. wValue ile request için bir değer gönderilebilir (sorgulanan desriptor tipi örneğin). Aşağıda host tarafından gönderilen request paketin değerlendirilme aşaması gösterilmiştir.

```

USB_StatusTypeDef USB_LL_SetupStage(USB_HandleTypeDef *pdev, uint8_t *psetup)
{
    USB_StatusTypeDef ret;

    USB_ParseSetupRequest(&pdev->request, psetup);

    pdev->ep0_state = USBD_EP0_SETUP;

    pdev->ep0_data_len = pdev->request.wLength;

    switch (pdev->request.bmRequest & 0x1FU)
    {
        case USB_REQ_RECIPIENT_DEVICE:
            ret = USBD_StdDevReq(pdev, &pdev->request);
            break;

        case USB_REQ_RECIPIENT_INTERFACE:
            ret = USBD_StdItfReq(pdev, &pdev->request);
            break;

        case USB_REQ_RECIPIENT_ENDPOINT:
            ret = USBD_StdEPReq(pdev, &pdev->request);
            break;

        default:
            ret = USBD_LL_StallEP(pdev, (pdev->request.bmRequest & 0x80U));
            break;
    }

    return ret;
}

```

Görsel - 6 bmRequest ile İşlem Seçim

```

USB_StatusTypeDef USBD_StdDevReq(USB_HandleTypeDef *pdev, USB_SetupReqTypeDef *req)
{
    USB_StatusTypeDef ret = USBD_OK;

    switch (req->bmRequest & USB_REQ_TYPE_MASK)
    {
        case USB_REQ_TYPE_CLASS:
        case USB_REQ_TYPE_VENDOR:
            ret = (USB_StatusTypeDef)pdev->pClass[pdev->classId]->Setup(pdev, req);
            break;

        case USB_REQ_TYPE_STANDARD:
            switch (req->bRequest)
            {
                case USB_REQ_GET_DESCRIPTOR:
                    USBD_GetDescriptor(pdev, req);
                    break;

                case USB_REQ_SET_ADDRESS:
                    USBD_SetAddress(pdev, req);
                    break;

                case USB_REQ_SET_CONFIGURATION:
                    ret = USBD_SetConfig(pdev, req);
                    break;

                case USB_REQ_GET_CONFIGURATION:
                    USBD_GetConfig(pdev, req);
                    break;
            }
    }
}

```

Görsel -7 bRequest ile Spesifik İşlem Seçimi

2.2-)Sinüs Sinyalinin Oluşturulması

Sinüs sinyali oluşturmak için look-up tablosu kullanıldı. Aşağıdaki fonksiyon host cihazın ses verisi için istekte bulunmasından sonra çağırılır ve gönderilecek ses verisi oluşturulur.

```

//number of channels
#define AUDIO_IN_CHANNELS 2
//sampling frequency in Hz
#define AUDIO_IN_SAMPLING_FREQUENCY 48000
//polling period in ms
#define HOST_POLL_REQUEST_PERIOD 1
//time between two consecutive samples in sec
#define TIME_BW_CONSECUTIVE_SAMPLES (double)1 / (AUDIO_IN_SAMPLING_FREQUENCY / HOST_POLL_REQUEST_PERIOD)
#define SINE_TABLE_SIZE 20000

void generateSineForUsbAudioClass(int16_t* sineArr, uint32_t sinFreq)
{
    static double t = 0;
    for(uint16_t i = 0; i < (AUDIO_IN_SAMPLING_FREQUENCY / (1000 / HOST_POLL_REQUEST_PERIOD)); i++)
    {
        t += TIME_BW_CONSECUTIVE_SAMPLES;

        //prevent overflow
        if(t >= (double)1 / sinFreq)
        {
            t -= (double)1 / sinFreq;
        }

        //channel-1
        sineArr[i * AUDIO_IN_CHANNELS] = sineLookupTable[(uint64_t)(t * SINE_TABLE_SIZE * sinFreq)];

        //channel-2 same signal will appear but with 180 degree phase difference
        sineArr[i * AUDIO_IN_CHANNELS + 1] = sineLookupTable[(uint64_t)(t * SINE_TABLE_SIZE * sinFreq + (SINE_TABLE_SIZE/2)) % SINE_TABLE_SIZE];
    }
}

```

Görsel - 8 Host Request Ardından Sinüs Sinyalinin Oluşturulduğu Fonksiyon

2.3-)Sinüs Sinyalinin Bilgisayara Gönderilmesi

USB haberleşmede control transfer, interrupt transfer, isochronous transfer, bulk transfer gibi farklı transfer tipleri kullanılır. Cihaz konfigürasyonu gibi işlemler için control transfer kullanılır. Ses transferi gibi gerçek zamanlı streaming işlemlerinde ise isochronous transfer kullanılır.

Isochronous transferde bant genişliği garanti altına alınmıştır ve ses gecikme olmadan sabit periyotlarla gönderilir. Bu periyot FS cihazlarda genellikle 1ms dir ve değiştirilebilir.

Aşağıdaki Format Interface descriptoru ses verisinin örnekleme frekansını, verinin kaç kanala ait olduğunu ve bir veri örneğinin(sample) kaç bitten oluştuğu gibi bilgileri içerir.

Endpoint-1 Standard Descriptor ses verisinin transferi için kullanılan endpointe ait descriptor'dür ve maximum paket boyutu, transfer tipi(isochronous) gibi bilgileri içerir.

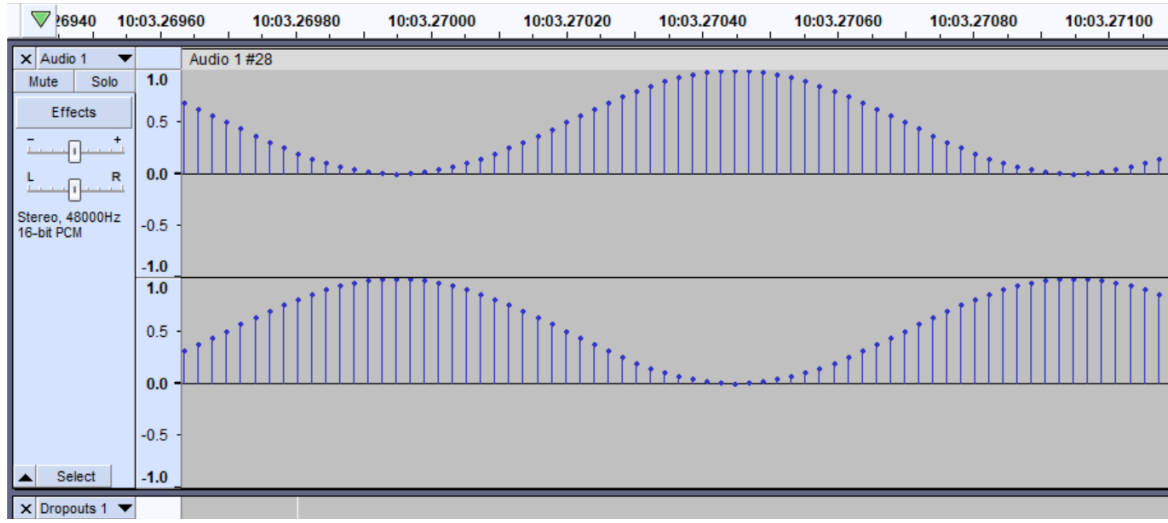
```
/* USB Microphone Audio Type I Format Interface Descriptor */
USBD_AUDIO_CfgDesc[index++] = 0x0B; /* bLength */
USBD_AUDIO_CfgDesc[index++] = AUDIO_INTERFACE_DESCRIPTOR_TYPE; /* bDescriptorType */
USBD_AUDIO_CfgDesc[index++] = AUDIO_STREAMING_FORMAT_TYPE; /* bDescriptorSubtype */
USBD_AUDIO_CfgDesc[index++] = AUDIO_FORMAT_TYPE_I; /* bFormatType */
USBD_AUDIO_CfgDesc[index++] = Channels; /* bNrChannels */
USBD_AUDIO_CfgDesc[index++] = 0x02; /* bSubFrameSize */
USBD_AUDIO_CfgDesc[index++] = 16; /* bBitResolution */
USBD_AUDIO_CfgDesc[index++] = 0x01; /* bSamFreqType */
USBD_AUDIO_CfgDesc[index++] = samplingFrequency&0xff; /* tSamFreq */
USBD_AUDIO_CfgDesc[index++] = (samplingFrequency>>8)&0xff;
USBD_AUDIO_CfgDesc[index++] = samplingFrequency>>16;
/* Endpoint 1 - Standard Descriptor */
USBD_AUDIO_CfgDesc[index++] = AUDIO_STANDARD_ENDPOINT_DESC_SIZE; /* bLength */
USBD_AUDIO_CfgDesc[index++] = 0x05; /* bDescriptorType */
USBD_AUDIO_CfgDesc[index++] = AUDIO_IN_EP; /* bEndpointAddress 1 in endpoint */
USBD_AUDIO_CfgDesc[index++] = 0x05; /* bmAttributes */
USBD_AUDIO_CfgDesc[index++] = ((samplingFrequency/1000+2)*Channels*2)&0xFF; /* wMaxPacketSize */
USBD_AUDIO_CfgDesc[index++] = ((samplingFrequency/1000+2)*Channels*2)>>8;
USBD_AUDIO_CfgDesc[index++] = 0x01; /* bInterval(polling period) */
USBD_AUDIO_CfgDesc[index++] = 0x00; /* bRefresh */
USBD_AUDIO_CfgDesc[index++] = 0x00; /* bSynchAddress */
```

Görsel 9 – Streaming format ve Endpoint Descriptroları

Host cihaz ses verisini istemek için 1ms periyotlarla IN token gönderir ve device cevap olarak ses verisini gönderir.

2.4-)Audacity Programında Sinyalin Görüntülenmesi

Aşağıdaki görselde 1kHz frekanslı sinüs sinyalinin Audacity programındaki çıktısı verilmiştir.



Görsel - 10 Audacity 1kHz Sinüs Sinyali