USB microphone based on STM32F4-DISCOVERY

STM32

I previously wrote about outputting audio via USB using the STM32F4-DISCOVERY board. Implementing the microphone, however, proved more challenging. I couldn't find any similar projects online.
I tried building a USB microphone on this board myself several times, and finally got it working.

Features of using the STM32F4-DISCOVERY microphone

The board is equipped with an MP45DT02 digital microphone. It outputs data in PDM format . Data from the microphone to the controller is transmitted via the I2S interface, which is a type of SPI (in this case, SPI2). The microphone is clocked via the same I2S (in this mode, I2S independently requests data from the microphone). It's worth noting that I2S uses a clock frequency from its own PLL, allowing for precise acquisition of specific audio sampling rates. However, this lacks synchronization with the rest of the controller's peripherals (if you configure any timer and I2S to generate identical frequencies, their interrupts will drift relative to each other).

Working with the microphone on the ST is demonstrated in the "audio playback and record" example in the "waverecorder.c" file. The example is quite complex, as I2S is interrupt-based, so I significantly modified the original file.
I2S initialization occurs in the WaveRecorder_SPI_Init(uint32_t Freq) function. Data from the microphone is collected in 16-bit packets (the microphone itself doesn't transmit numeric data; it transmits a bitstream). The Freq value determines the packet transmission frequency (the clock frequency will be 16 times higher). The documentation specifies that the microphone clock frequency should be higher than 1 MHz, so I chose a packet frequency of 64 kHz.
Since the I2S controller module is designed to work only with a stereo stream, and the microphone is monophonic, when configuring I2S in this function, I have to specify a frequency half the required value.

Microphone data can't be used directly; it must be filtered and decimated. In their example, ST uses the precompiled libPDMFilter library, which contains the necessary filter. It seems to me that this filter doesn't work as described in the

Live broadcast

Comments Publications

lightstar1411 → "Strange" Wireless Microphone 47 → Circuit Design

Andre225 → AtMega1284p/644p and W5500 Ethernet 16 → AVR

anakost → SPD EEPROM Programmer for DDR3 and DDR4 Memory Modules 1 → Tool

sunjob → Printed circuit board for AtMega 8535/16/32/644/1284 49 → AVR

sunjob → W5500+STM32+SD web server 19 → Blog named after. GYUR22

sunjob → Connecting a microcontroller to a local network: HTTP and CGI 146 → Connecting hardware to a computer.

sunjob → Connecting the microcontroller to the local network: TCP client 84 → Communication between the hardware and the computer.

sunjob → Connecting a microcontroller to a local network: Conclusion 80 → Connecting the hardware to the computer.

sunjob → Connecting a microcontroller to a local network: performance tests and a brief description of the API stack 50 → Connecting the hardware to the computer.

sunjob → Connecting a microcontroller to a local network: HTTP and CGI (conclusion) 85 → Connecting hardware to a computer.

sunjob → Connecting a microcontroller to a local network: TCP and HTTP (continued) 46 → Connecting the hardware to the computer.

sunjob → Connecting a microcontroller to a local network: Broadcast messages and DHCP 37 → Communication between the hardware and the computer.

sunjob → Connecting the microcontroller to the local network: UDP client 45 → Communication between the hardware and the computer.

sunjob → Connecting the microcontroller to the local network: UDP server 106 → Communication between the hardware and the computer.

sunjob → Connecting a microcontroller to a local network: working with ENC28J60 112 → Connecting the hardware to the computer.

sunjob → Connecting the microcontroller to the local network 42 → Connecting the hardware

documentation (AN3998), but implementing my own filter would be quite problematic.
In my project, I configured the filter to output data at a rate of 16 KSPS (using a decimation factor of 64). However, for the filter to function correctly at this rate, it must be sent 64 16-bit words, and as a result, it will return 16 16-bit signed audio data values. The filter has an adjustable gain. Another feature of the filter is that input data must be passed to it with the bytes in the words swapped. This somewhat slows down the filtering process.

To accumulate 64 words, I used a DMA operating in double-buffering mode. The I2S module generates DMA data transfer requests. Once one of the buffers is full, the DMA generates an interrupt, and the filter is called in its handler. Two 16-word buffers are also allocated for storing audio data, which are used to store the generated PCM data.

In this project, I implemented a microphone output to the onboard audio DAC, allowing you to listen to the microphone's sound by connecting headphones to the board. Audio data is transferred to the audio DAC via another I2S module and another DMA channel. Again, since the audio DAC is stereo, each word of data must be duplicated before sending the microphone data (this is done in EVAL_AUDIO_TransferComplete_CallBack()).

Note that at maximum microphone gain (the documentation states a maximum of 64, although 90 worked for me), the microphone's sensitivity is very high.

Transferring data via USB

The ST V2.1.0 library is used for working with USB. During my experiments, I switched to the library taken from here .
This version has some commented-out sections of the code (search for "#ifdef original").
I tried descriptors from the above link, from a very useful article **by lleeloo** , and from AN295 from Silicon Labs.

For a long time, I couldn't get the library to work—the system saw the device as a microphone, but there were no Callback calls from the isochronous IN Endpoint, which is where data transfer needs to be initiated.
During my experiments, I discovered that for USB to work correctly, I need to perform a one-time audio data transfer from the SOF Callback (I do this when the system selects the active interface), after which IN Callbacks begin to be generated. However, there is a peculiarity to data transfer: before each transfer, the FIFO of the given endpoint needs to be cleared:

```
DCD_EP_Flush(pdev,AUDIO_IN_EP);
DCD_EP_Tx (pdev,AUDIO_IN_EP, (uint8_t*)(data), AUDIO_IN_PACKET);
```

There are no particular software complications with audio transfer—in the IN Callback, I simply call data transfer from the buffer that was filled during the last filter call.

Link to the project (for IAR 6.30):
github.com/iliasam/STM32F4_USB_MICROPHONE

USB , microphone , STM32F4-Discovery , microphone , MP45DT02 , USB Audio IN

+7 November 11, 2014, 10:38 PM **citizen** 1

Files in the topic: STM32F4_USB_MICROPHONE.zip

Comments (25)

RSS collapse / expand

Attach the project to the post. 0

to the computer.

trengtor → PIP Regulator 3 → Algorithms and Software Solutions

penzet → Sprint Layout in OS X 18 → Software for Electronics Engineers

VGA → EmBitz 6 → Software for electronics engineers

VGA → Rail-to-rail: ideal control unit or a clever marketing ploy? 1 → Theory, measurements, and calculations

Full broadcast | RSS

1-Wire Altera Android Arduino ARM Assembler AVR C++ compel DIY enc28j60 Ethernet FPGA gcc I2C IAR KEIL LaunchPad LCD led Linux LPCXpresso MSP430 nxp PCB PIC pinboard2 RS-485 RTOS STM32 STM8 STM8L TI UART USB algorithm assembler ADC library power unit detail display idea tool contest competition 2 LUT microcontrollers for beginners review Debug board soldering iron PCB pay FPGA crafts purchases programmer programming LED software scheme circuit design Technologies smart home photoresist freebie crap Watch humor

Blogs

Top

AVR	**38.98**
STM8	**37.92**
Garbage truck	**29.53**
STM32	**28.46**
Detail	**24.63**
Connection between hardware and computer.	**April 24**
Circuitry	**18.15**
Smart home	**17.75**
MSP430	**17.13**
LPC1xxx	**14.79**

All blogs

VGA
November 12, 2014, 2:38 AM

Added. 0

citizen
November 12, 2014, 9:32 AM

A similar topic of working with sound, but transmitting it wirelessly and meaningfully applying the results. 0
Speech recognition on STM32F4-Discovery http://habrahabr.ru/post/146501/
I'm amazed by the little box that controls women behind the wheel with its voice... *"Turn left..., turn right..."*
Maybe it could start doing something that talks and points around the house?
"Go to the kitchen..., your milk's boiled over..."

plcist
November 12, 2014, 10:00

I downloaded the project, a port for CooCox is just around the corner :-) 0

Romanets
November 12, 2014, 11:01

CooCox has one drawback: all paths in the project are absolute, so if you copy the project to another computer, it won't work. So I don't see much point in publishing such a project. 0

citizen
November 12, 2014, 12:07 PM

Everything is tolerated normally, you have incorrect information regarding coconut. +1

Romanets
November 12, 2014, 1:15 PM

Coocox has a flaw—a lack of understanding of what's going on. You check the boxes, but what's actually happening is a mystery. 0

Villain
November 12, 2014, 2:28 PM

At work we use it only as a medium (without lib) and there are no such problems ;) 0

xar
November 12, 2014, 9:09 PM

Please provide an example of the recorded audio. 0

Vitalik
November 12, 2014, 8:48 PM

dropmefiles.com/s5ttz 0

Romanets
November 13, 2014, 5:17 PM

Here are 2 versions of PDM_Filter: +1
* the original in assembler and without the processor checkback (rewritten to support hardfloat)
* and a FIR filter in C that does the same thing, but with a fixed set of coefficients
github.com/piratfm/codec2_m4f/tree/master/lib/PDM_filter

tipok
November 13, 2014, 8:45 PM

Thank you very much! Could you tell me what could be causing the barely noticeable delay in audio streaming? 0

angry kid

February 22, 2015, 2:25 AM

Hello! Could you please tell me if this project can be modified to allow reception via a microphone connected to the audio jack? 0

ravil
April 17, 2016, 3:06 PM

The audio connector on the STM32F4-Discovery is intended for audio output only. 0

citizen
April 17, 2016, 10:19 PM

Thank you 0

ravil
April 17, 2016, 10:31 PM

You can easily attach an external i2s codec to the Discovery 0

Romanets
April 20, 2016, 9:35 PM

By the way, there's no need to reset anything there. This is because the packet parity filter on isochronous endpoints is non-disabled.
The parity of the current frame must be captured in the IsoIncomplete interrupt, inverted (since the microphone receives isochronous CT data every frame, and at Full Speed, the parity alternates between DATA0 and DATA1, etc.), and sent to the IN endpoint in the SOF when the "desired" parity and the current frame's parity match.
Why in the SOF? Because when sending data to the CT, the current frame's parity (correct) will be written to its parity filter. Another trick is that we send as much data as we have (the IN endpoint must be asynchronous). +1

Romanets
April 20, 2016, 9:43 PM

Thanks for the comment, I'll have to try reworking the code. 0

citizen
April 20, 2016, 10:37 PM

I made this full duplex as part of my hobby project... 0

Romanets
April 20, 2016, 10:43 PM

Good evening! What if I use an ADC?
The goal is to connect a piezoelectric transistor or microphone, send the output signal to the computer, receive it in LabView, and plot the spectrum. 0

ravil
April 20, 2016, 10:47 PM

It doesn't matter where the digital data sent to the endpoint comes from. Similarly, you configure the ADC, DMA, and perhaps even a timer as the sampling frequency source. It will work. 0

Romanets
April 20, 2016, 10:53 PM

Why can't I use a regular sound card (USB/PCI/other)?
But if absolutely necessary, the microphone can be connected to the STM32 ADC— we.easyelectronics.ru/AVR/zapis-zvuka.html 0

citizen
April 20, 2016, 10:56 PM

My professor at university insists on this. I suggested a sound card... he insists on an STM32 so he can make a portable device in graduate school. Thanks, we'll learn. 0

ravil
April 20, 2016, 11:08 PM

The STM32 has one bug, but I won't mention it yet. I wonder if you'll stumble upon it or if I'm just stupid.

0

Romanets
April 20, 2016, 11:21 PM

I'm an energy guy. I just recently got involved with MK and I'm still pretty green and dumb :) I hope the joint doesn't catch me off guard.

0

ravil
April 20, 2016, 11:32 PM

Only registered and authorized users can leave comments.