Installing STM32CubeIDE in Linux Mint 21.1 and in Windows 10, and compiling and uploading the STM32F411 USB Audio DAC firmware.

Download STM32CubeIDE for Debian from here: https://www.st.com/en/development-tools/stm32cubeide.html - register or confirm with and email) – the file is named:
en.st-stm32cubeide_1.11.2_14494_20230119_0724.unsigned_amd64.deb_bundle.sh.zip

Extract the file in the Download folder and open a terminal and make it executable the run it:

```
sudo chmod +x ***.sh
sudo ./***.sh
```

Ignore the message about the sandbox etc, and reboot for the new udev rules to take effect.

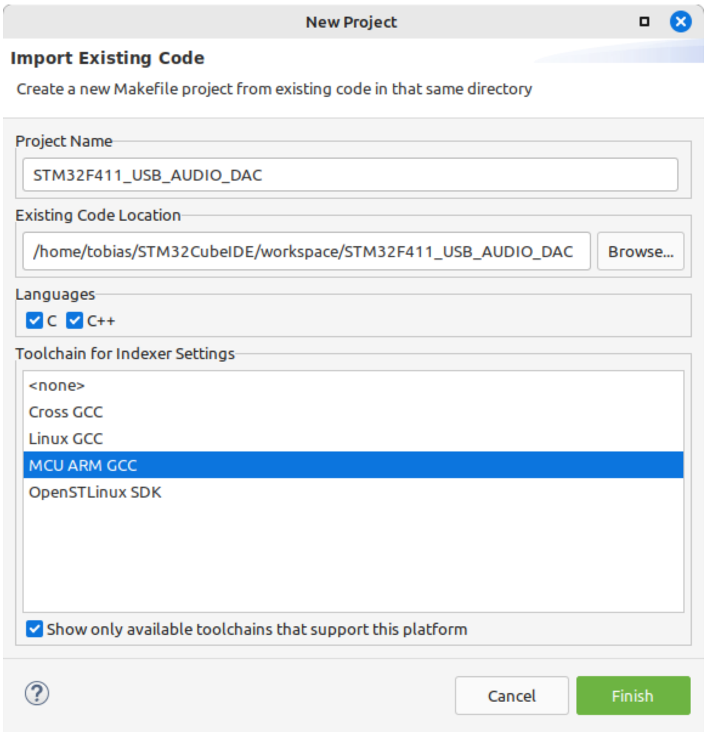
Then run STM32CubeIDE and select a default workspace such as /home/user/STM32CubeIDE/workspace/, and then select File -> New -> STM32Project. Wait a few minutes until the message about a target selector appears then select your STM32 MCU - in this case type STM32F411CEU6 in the top lefthand box then select it in the bottom table and press Next.

Wait until the STM32F411 definition files (and any updates) have been downloaded and installed.

You can then follow the instructions https://community.st.com/s/question/0D53W00000HNrEsSAL/how-to-install-stm32cubeide-on-ubuntu further for doing a quick blinky test, or just cancel the project and start with the new USB Audio DAC project below. Remember to connect the MCU and the ST-Link-USB adapter to the PC.

Download the zip file from here https://github.com/har-in-air/STM32F411_USB_AUDIO_DAC and extract it in /home/user/STM32CubeIDE/workspace/. Then remove the -main part of the extracted folder and open CubeIDE.

Choose File -> New -> Makefile Project with Existing Code, browse to the STM32F411_USB_AUDIO_DAC folder and select it. Choose MCU Arm GCC – see below, and press Finish:



New Project

Import Existing Code

Create a new Makefile project from existing code in that same directory

Project Name: STM32F411_USB_AUDIO_DAC

Existing Code Location: /home/tobias/STM32CubeIDE/workspace/STM32F411_USB_AUDIO_DAC

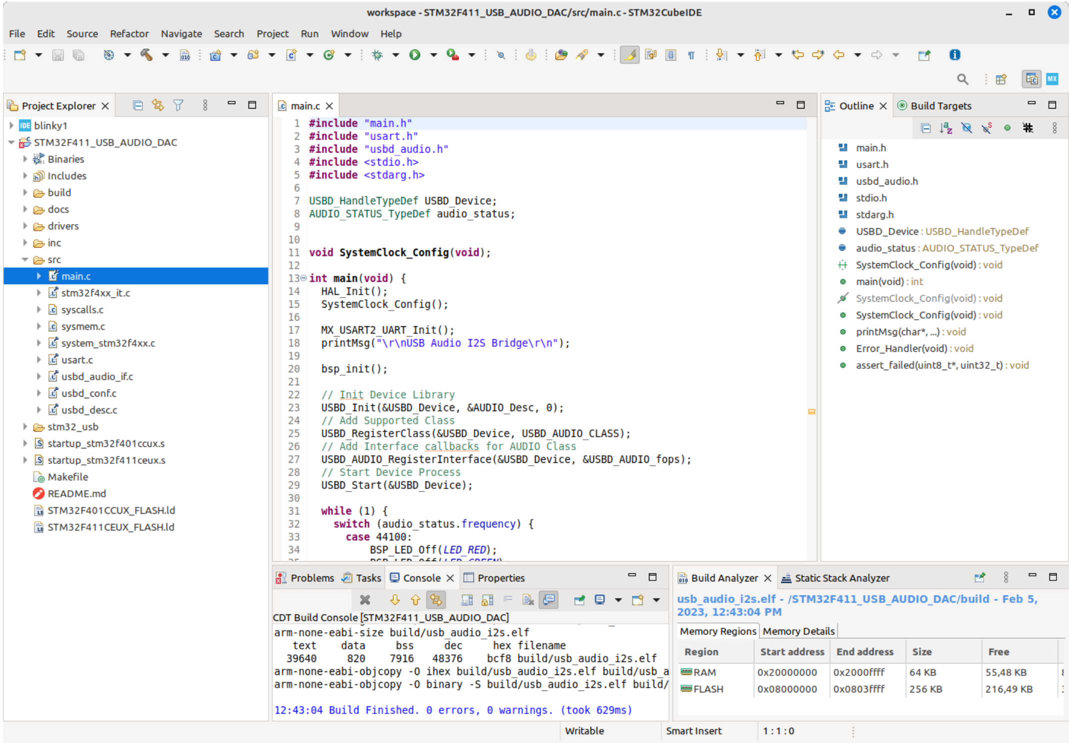
Languages: ☑ C ☑ C++

Toolchain for Indexer Settings:
- <none>
- Cross GCC
- Linux GCC
- MCU ARM GCC
- OpenSTLinux SDK

☑ Show only available toolchains that support this platform

Press Project -> Build Project see below, then OK:

It should complete the build -see below:



```c
#include "main.h"
#include "usart.h"
#include "usbd_audio.h"
#include <stdio.h>
#include <stdarg.h>

USBD_HandleTypeDef USBD_Device;
AUDIO_STATUS_TypeDef audio_status;


void SystemClock_Config(void);

int main(void) {
  HAL_Init();
  SystemClock_Config();

  MX_USART2_UART_Init();
  printMsg("\r\nUSB Audio I2S Bridge\r\n");

  bsp_init();

  // Init Device Library
  USBD_Init(&USBD_Device, &AUDIO_Desc, 0);
  // Add Supported Class
  USBD_RegisterClass(&USBD_Device, USBD_AUDIO_CLASS);
  // Add Interface callbacks for AUDIO Class
  USBD_AUDIO_RegisterInterface(&USBD_Device, &USBD_AUDIO_fops);
  // Start Device Process
  USBD_Start(&USBD_Device);

  while (1) {
    switch (audio_status.frequency) {
      case 44100:
          BSP_LED_Off(LED_RED);
```

CDT Build Console [STM32F411_USB_AUDIO_DAC]

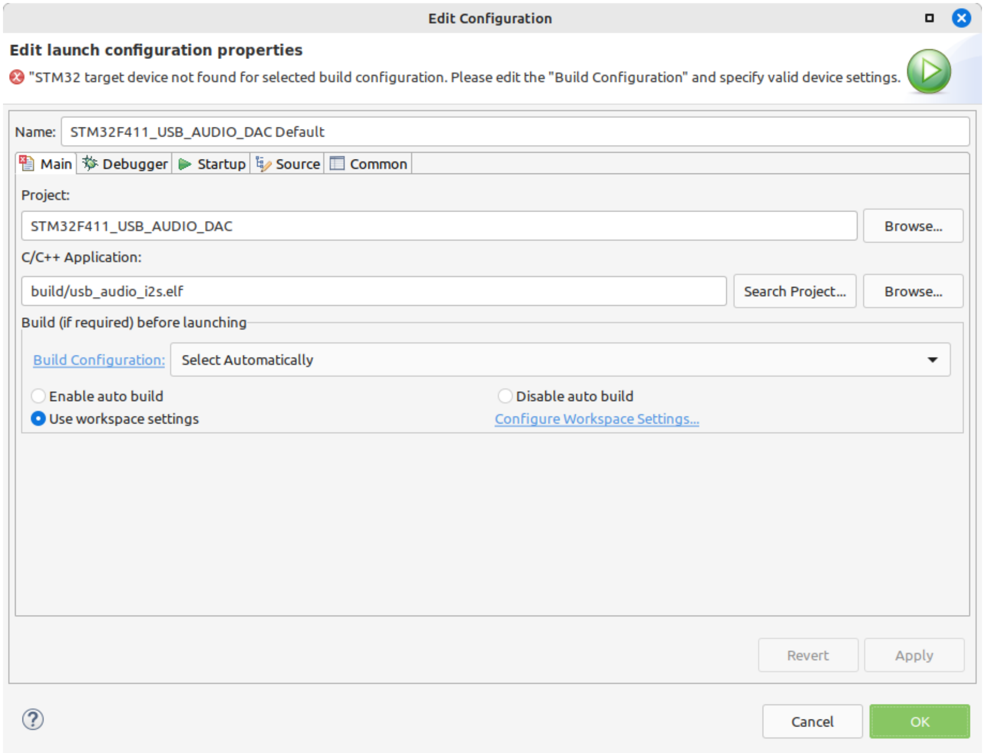
```
arm-none-eabi-size build/usb_audio_i2s.elf
   text    data     bss     dec     hex filename
  39640     820    7916   48376    bcf8 build/usb_audio_i2s.elf
arm-none-eabi-objcopy -O ihex build/usb_audio_i2s.elf build/usb_a
arm-none-eabi-objcopy -O binary -S build/usb_audio_i2s.elf build/

12:43:04 Build Finished. 0 errors, 0 warnings. (took 629ms)
```

usb_audio_i2s.elf - /STM32F411_USB_AUDIO_DAC/build - Feb 5, 2023, 12:43:04 PM

Region	Start address	End address	Size	Free
RAM	0x20000000	0x2000ffff	64 KB	55,48 KB
FLASH	0x08000000	0x0803ffff	256 KB	216,49 KB

You can check in the build subfolder for the binaries files .bin and .hex – which you can use to program the MCU using STM32CubeProgrammer or continue to use the IDE to upload the code.

Press the Run button see below, then OK:



Edit launch configuration properties

"STM32 target device not found for selected build configuration. Please edit the "Build Configuration" and specify valid device settings.

Name: STM32F411_USB_AUDIO_DAC Default

Project: STM32F411_USB_AUDIO_DAC

C/C++ Application: build/usb_audio_i2s.elf

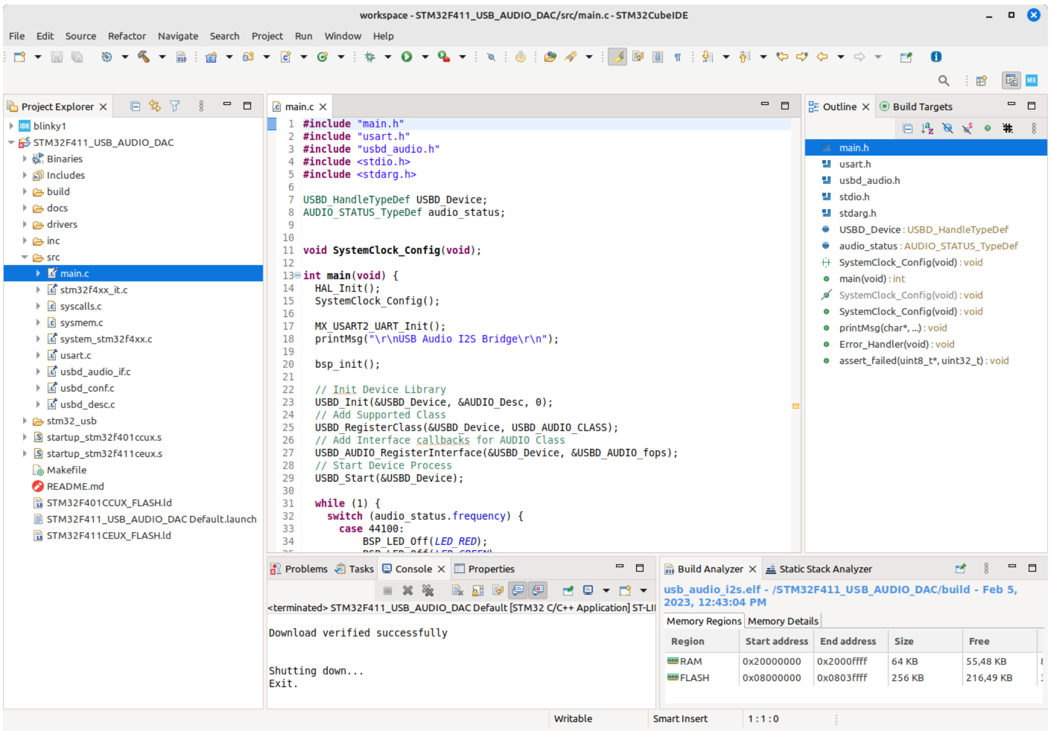
Build (if required) before launching

Build Configuration: Select Automatically

○ Enable auto build ○ Disable auto build

● Use workspace settings Configure Workspace Settings...

Then it will complete the upload:



```c
#include "main.h"
#include "usart.h"
#include "usbd_audio.h"
#include <stdio.h>
#include <stdarg.h>

USBD_HandleTypeDef USBD_Device;
AUDIO_STATUS_TypeDef audio_status;


void SystemClock_Config(void);

int main(void) {
  HAL_Init();
  SystemClock_Config();

  MX_USART2_UART_Init();
  printMsg("\r\nUSB Audio I2S Bridge\r\n");

  bsp_init();

  // Init Device Library
  USBD_Init(&USBD_Device, &AUDIO_Desc, 0);
  // Add Supported Class
  USBD_RegisterClass(&USBD_Device, USBD_AUDIO_CLASS);
  // Add Interface callbacks for AUDIO Class
  USBD_AUDIO_RegisterInterface(&USBD_Device, &USBD_AUDIO_fops);
  // Start Device Process
  USBD_Start(&USBD_Device);

  while (1) {
    switch (audio_status.frequency) {
      case 44100:
          BSP_LED_Off(LED_RED);
```

```
<terminated> STM32F411_USB_AUDIO_DAC Default [STM32 C/C++ Application] ST-LI

Download verified successfully


Shutting down...
Exit.
```

usb_audio_i2s.elf - /STM32F411_USB_AUDIO_DAC/build - Feb 5, 2023, 12:43:04 PM

Region	Start address	End address	Size	Free
RAM	0x20000000	0x2000ffff	64 KB	55,48 KB
FLASH	0x08000000	0x0803ffff	256 KB	216,49 KB

You can now close the IDE, unplug the ST-Link and plug the STM32F411 in with its own USB adapter – it should show up as a new USB Audio device using aplay -L

For windows 10 install the CubeIDE using the exe file and extract the Github files under the windows assigned workspace. Then follow the same instructions as for Linux above.