

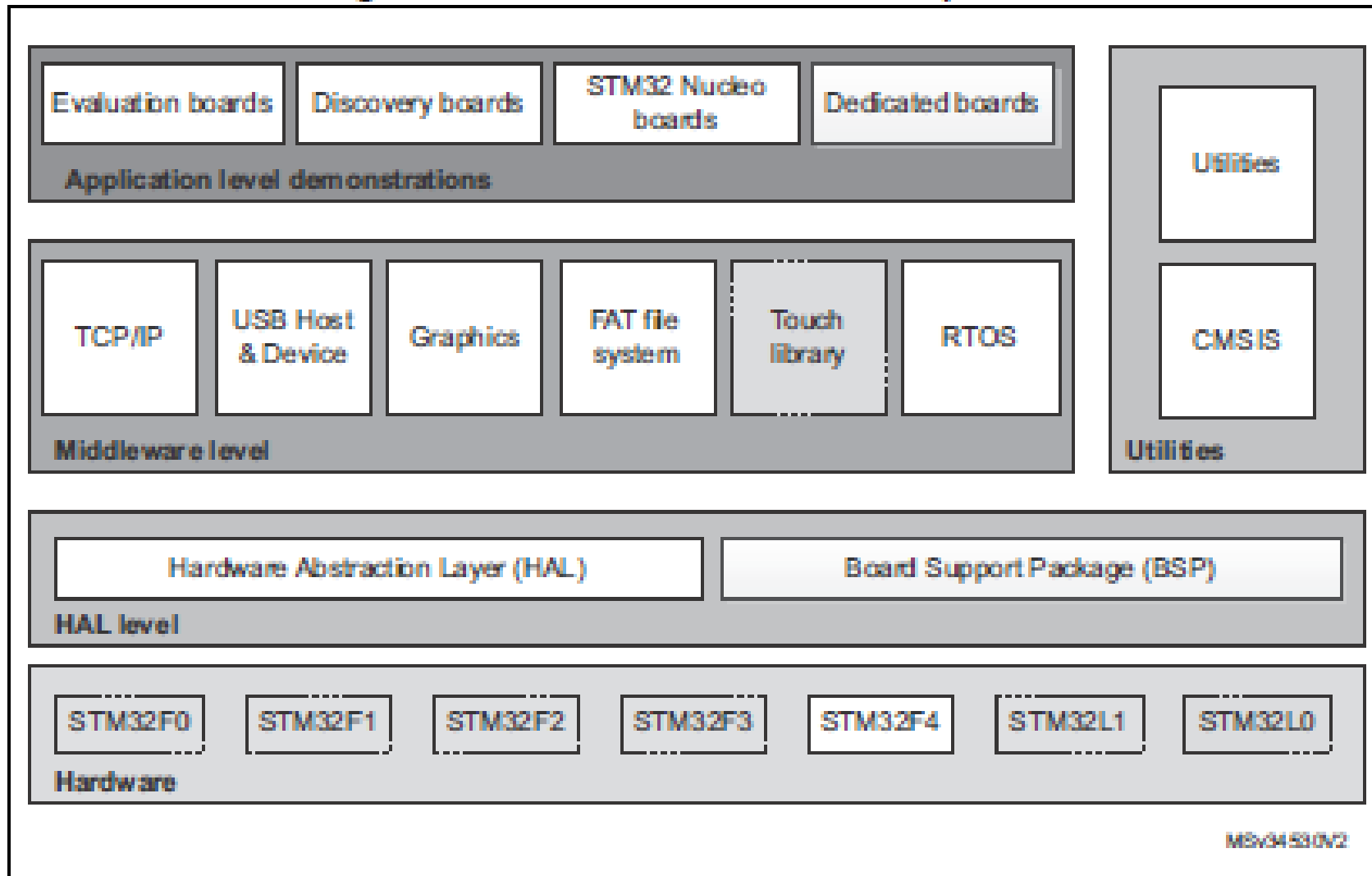
Software structure

To read a codec register:

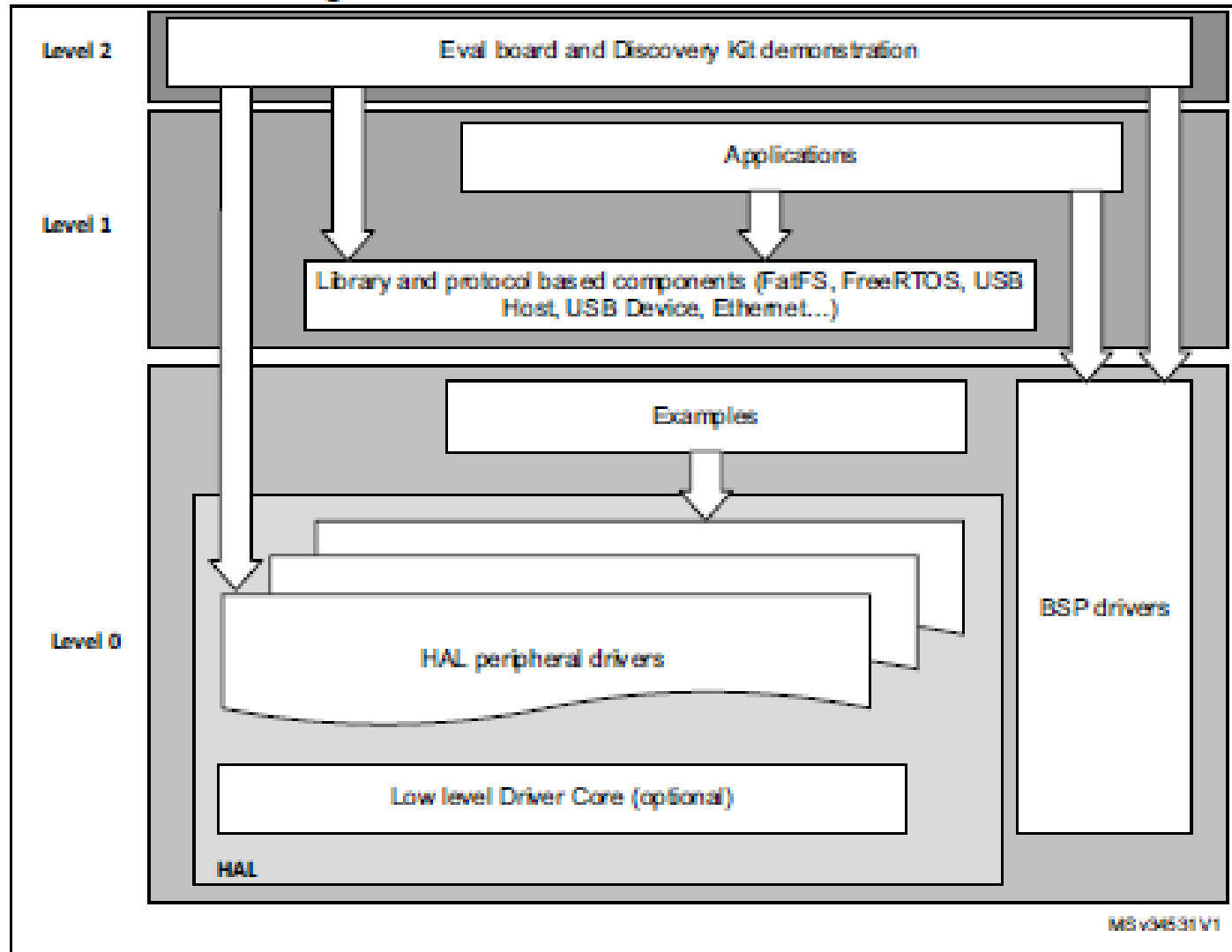
START - A7.0 - write to slave Reg codec reg #

START - A7.1 - read slave Value reg. value from slave ~~STOP~~

STM32F4 “Firmware”



STM32CubeF4 Architecture



Nelson Project

The screenshot shows the uVision IDE interface. The title bar indicates the project path: `C:\Users\nelsovp\Documents\Course Materials\ELEC5260_6260\ARM Files\STM32F4 Programs\AudioVPN\AudioVPN.uvprojx - uVision`. The menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains various icons for file operations and debugging, with a dropdown menu currently showing 'GPIO'. The 'Project' window on the left displays the following structure:

- Project: AudioVPN
 - Target 1
 - User
 - Audio.c
 - SystemClockConfig.c
 - BSP_Discovery
 - cs43l22.c
 - stm32f4_discovery.c
 - CMSIS
 - Device
 - RTE_Device.h (STM32Cube Framework:Classic)
 - startup_stm32f407xx.s (Startup)
 - stm32f4xx_hal_conf.h (STM32Cube Framework:Classic)
 - system_stm32f4xx.c (Startup)
 - stm32f4xx_hal.c (STM32Cube HAL:Common)
 - stm32f4xx_hal_cortex.c (STM32Cube HAL:Cortex)
 - stm32f4xx_hal_dma.c (STM32Cube HAL:DMA)
 - stm32f4xx_hal_dma_ex.c (STM32Cube HAL:DMA)
 - stm32f4xx_hal_gpio.c (STM32Cube HAL:GPIO)
 - stm32f4xx_hal_i2c.c (STM32Cube HAL:I2C)
 - stm32f4xx_hal_pwr.c (STM32Cube HAL:PWR)
 - stm32f4xx_hal_pwr_ex.c (STM32Cube HAL:PWR)
 - stm32f4xx_hal_rcc.c (STM32Cube HAL:RCC)
 - stm32f4xx_hal_rcc_ex.c (STM32Cube HAL:RCC)
 - stm32f4xx_hal_spi.c (STM32Cube HAL:SPI)

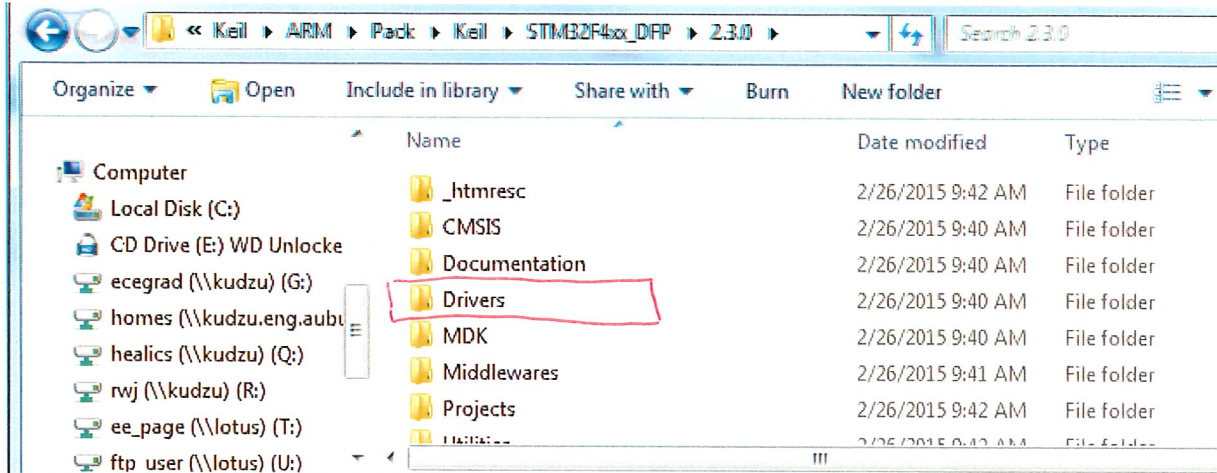
Handwritten blue annotations include:

- A bracket on the left side of the 'User' and 'BSP_Discovery' folders labeled "My files".
- Text next to the 'BSP_Discovery' folder: "Copied from".
- A long path written across the workspace: `C:\Keil\ARM\Pack\Keil\STM32F4xx_DFP\2.7.0\Drivers\BSP\Components\cs43l22\STM32F4-Discovery`.

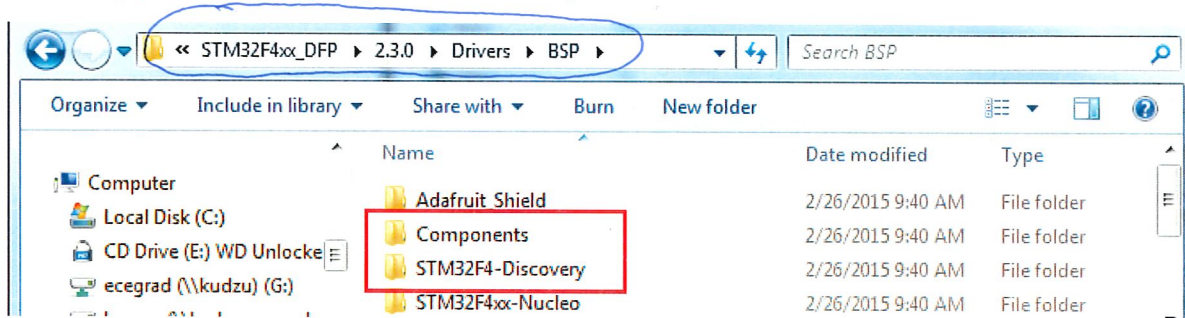
The 'Build Output' window at the bottom is currently empty. The 'ST-Link Debugger' status bar is visible at the bottom right.

2.7 also

STM32F4xx_DFP Version 2.3 Directory (under Keil installation on drive C:)

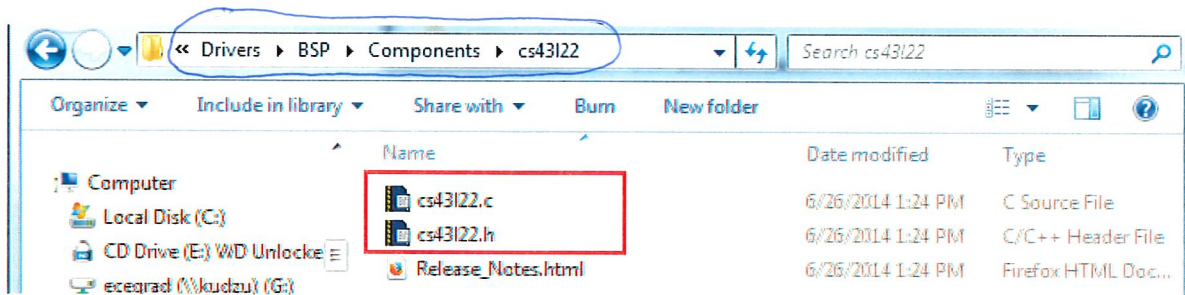


Board Support Package (BSP) drivers directory:

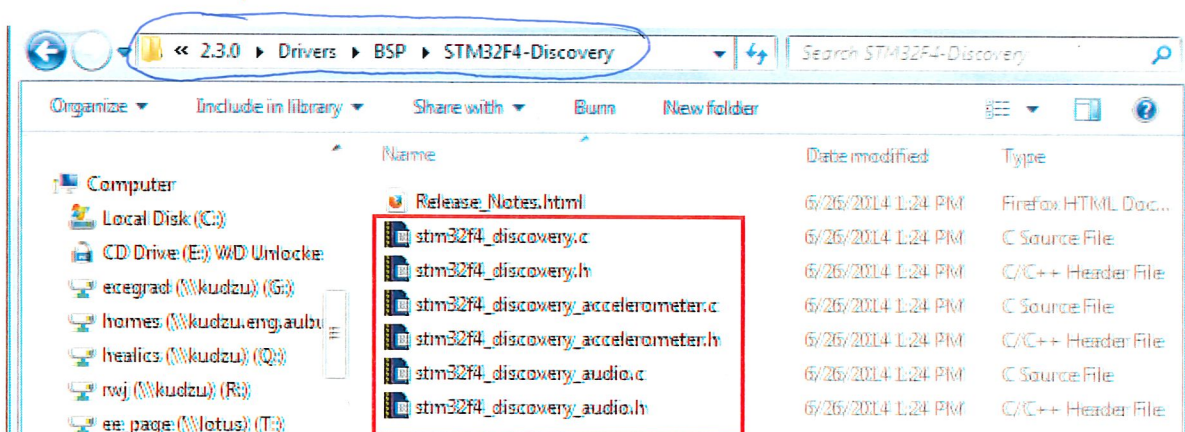


BSP/Component/
STM32F4-
Discovery

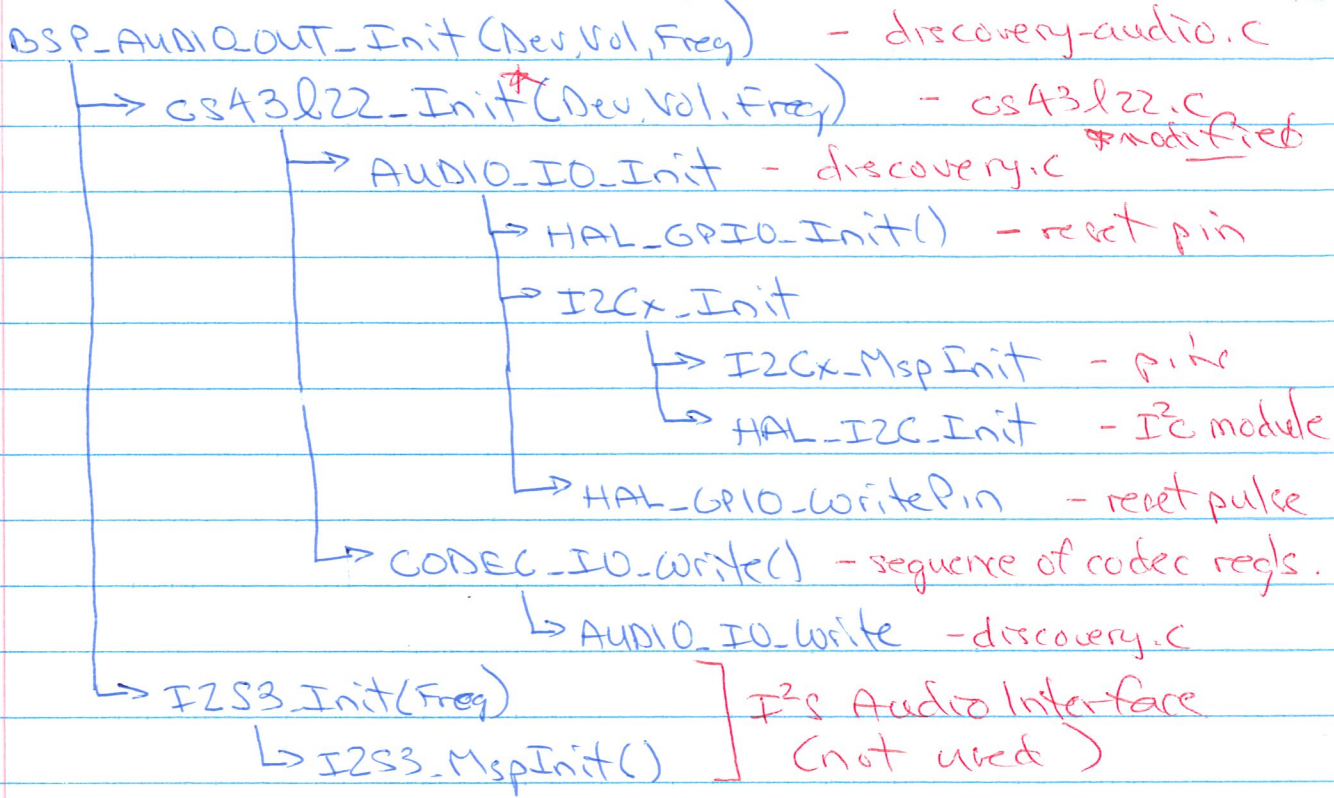
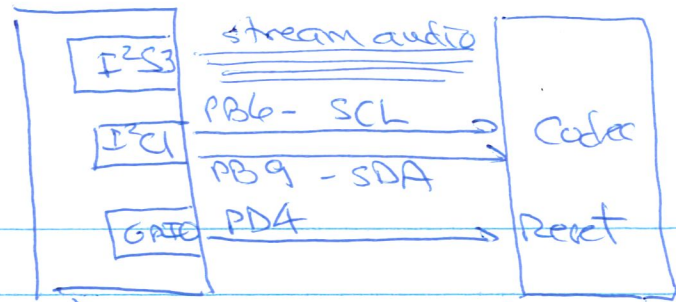
CS43L22 Audio Codec driver:



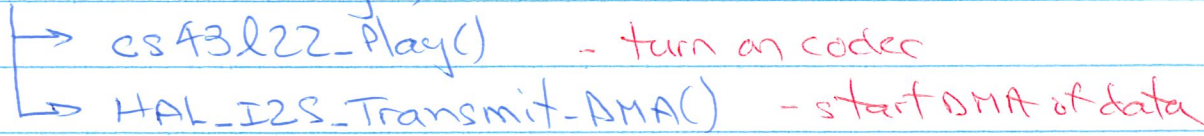
STM32F4-Discovery board drivers:



Hierarchy of functions



BSP_AUDIO_OUT_Play()



(Other AUDIO_OUT functions)

CODEC_IO_Write()

↳ AUDIO_IO_Write()

↳ I2Cx_WriteData()

↳ HAL_I2C_Mem_Write()

Pack > Keil > STM32F4xx_DFP > 2.7.0 > Drivers > BSP > STM32F4-Discovery

stm32f4-discovery.c

stm32f4-discovery-audio.c

stm32f4-discovery-accelerometer.c

BSP-AUDIO-IN-...

BSP-AUDIO-OUT-Init()

- Play()

- Stop()

- SetVolume()

⋮

stm32f4-discovery.c (oh) - lower level board setup

LED Functions: Led = LED3 | LED4 | LED5 | LED6

BSP_LED_Init(Led) → [LEDx-GPIO-CLK-ENABLE() / HAL-GPIO-Init()]

BSP_LED_On(Led) → HAL-GPIO_WritePin()

BSP_LED_Off(Led) →

BSP_LED_Toggle(Led) → HAL-GPIO_TogglePin()

Button Functions:

BUTTON-KEY

BSP_PB_Init(Button, Mode*) → [BUTTONx-GPIO-CLK-ENABLE() / HAL-GPIO-Init]

BSP_PB_GetState(Button) → HAL-GPIO_GetState()

* BUTTON-MODE-GPIO

BUTTON-MODE-EXTI - also calls NVIC funcs.

Audio Functions:

* AUDIO_IO_Init() I2C addr. - init GPIO clock, pin, I2C reg

* AUDIO_IO_Write(Addr, Reg, Value) → I2Cx_WriteData(A, R, V)

AUDIO_IO_Read(Addr, Reg) → I2Cx_ReadData(A, R)

I2C Functions:

I2Cx_Init() → HAL-I2C_Init()

I2Cx_WriteData(A, R, V) → HAL-I2C_Mem_Write(A, R, V)

I2Cx_ReadData(A, R) → HAL-I2C_Mem_Read(A, R)

I2Cx_MspInit() - config. GPIO pins for SCL/SDA

HAL-I2C Functions

I²C HAL driver functions used.

HAL-I2C-Init (I2C_HandleTypeDef *hi2c)

HAL-I2C-MspInit (hi2c) - µC pine, etc.

HAL-I2C-Mem-Write

HAL-I2C-Mem-Read

HAL-I2C-IsDeviceReady

Data Structures

I2C_HandleTypeDef

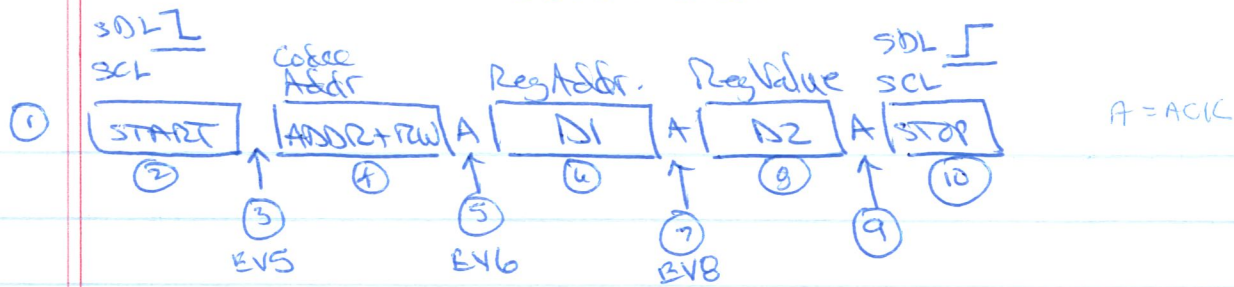
I2C_InitTypeDef

I2C_TypeDef

Init: clock speed, Duty Cycle (fast mode)
Own Address1 (7 or 10), Addressing Mode (7/10)
(Dual Address Mode, Own Address2)

Handle: *Instance
Init
*pBuffPtr
Xfer Size
Xfer Count
*hdma tx
*hdma rx
Lock
state
ErrorCode

I2C Low-Level Data Write



Codec Write Register (Register Addr, Register Value)
 \Downarrow `CODEC_IO_WRITE` \rightarrow `HAL_I2C_Mem_Write()`

- ① • `I2C_GetFlagStatus(, I2C_FLAG_BUSY)` - wait until bus not busy
- ② • `I2C_GenerateStart(, ENABLE)`
- ③ • `I2C_CheckEvent(, I2C_EVENT_MASTER_MODE_SELECTED)`
 • we are now Master: EV5 \Rightarrow BUSY, MSL, SB flags.
 (start generated) (start generated)

④ `I2C_Send7bitAddress(, 100101A0, I2C_DIRECTION_TRANSMIT)` $R/W = 0$

⑤ `I2C_CheckEvent(, I2C_EVENT_MASTER_TRANSMITTER_MODE_SELECTED)`

EV6: ACK received from slave - we can Xmit as master.

Flags: BUSY, MSL, ADDR, TXE, TRA
 end addr. Xmit Xmit buffer empty. RW Xmit

⑥ `I2C_SendData(, RegisterAddr)` - Codec reg. #

⑦ `I2C_CheckEvent(, I2C_EVENT_MASTER_BYTE_TRANSMITTING)`

EV8: data now shifting out to SDL.

Flags: BUSY, MSL, TXE, TRA

⑧ `I2C_SendData(, RegisterValue)` - (Codec reg. value)

⑨ `!I2C_GetFlagStatus(, I2C_FLAG_BTF)`

all data bytes finished

⑩ `I2C_GenerateStop(, ENABLE)`

EV1 - address match

EV7 - master byte received

EV2 - slave byte received

EV9 - master 10-bit addr. mode

EV3 - slave byte sent

Test flags in `SR1:SR2`

EV4 - slave STOP detected