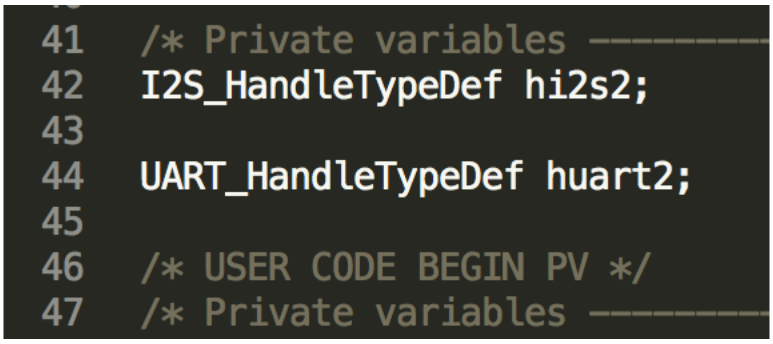
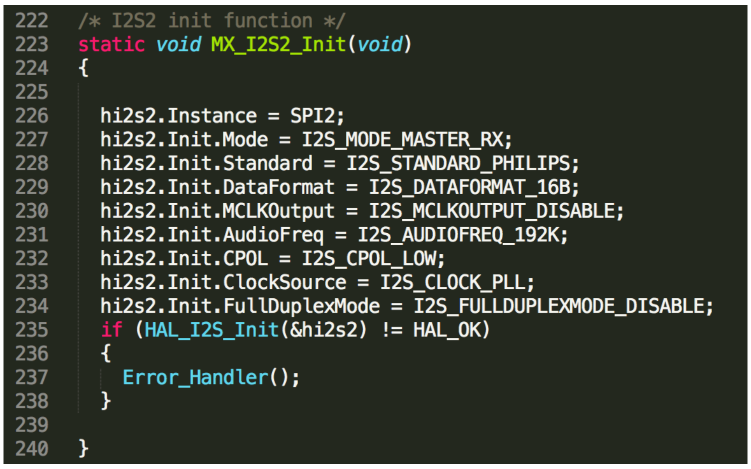
MP45DT02 : digital microphone

Extra ports that was opened and generated by STMCube.
- I2S2 : using I2S standard format on digital interface.
- UART2 : for sent data via USB by UART protocol.



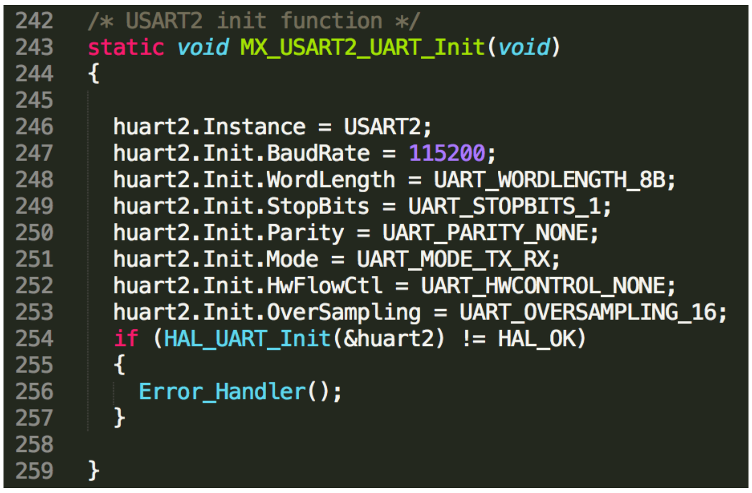
```
/* Private variables ---------
I2S_HandleTypeDef hi2s2;

UART_HandleTypeDef huart2;

/* USER CODE BEGIN PV */
/* Private variables ---------
```



```
/* I2S2 init function */
static void MX_I2S2_Init(void)
{

  hi2s2.Instance = SPI2;
  hi2s2.Init.Mode = I2S_MODE_MASTER_RX;
  hi2s2.Init.Standard = I2S_STANDARD_PHILIPS;
  hi2s2.Init.DataFormat = I2S_DATAFORMAT_16B;
  hi2s2.Init.MCLKOutput = I2S_MCLKOUTPUT_DISABLE;
  hi2s2.Init.AudioFreq = I2S_AUDIOFREQ_192K;
  hi2s2.Init.CPOL = I2S_CPOL_LOW;
  hi2s2.Init.ClockSource = I2S_CLOCK_PLL;
  hi2s2.Init.FullDuplexMode = I2S_FULLDUPLEXMODE_DISABLE;
  if (HAL_I2S_Init(&hi2s2) != HAL_OK)
  {
    Error_Handler();
  }

}
```

I2S2 Configuration

Mode : Master Receive
Standard : Philips
Data Format : 16 Bytes
Audio Frequency : 192K
Others : default

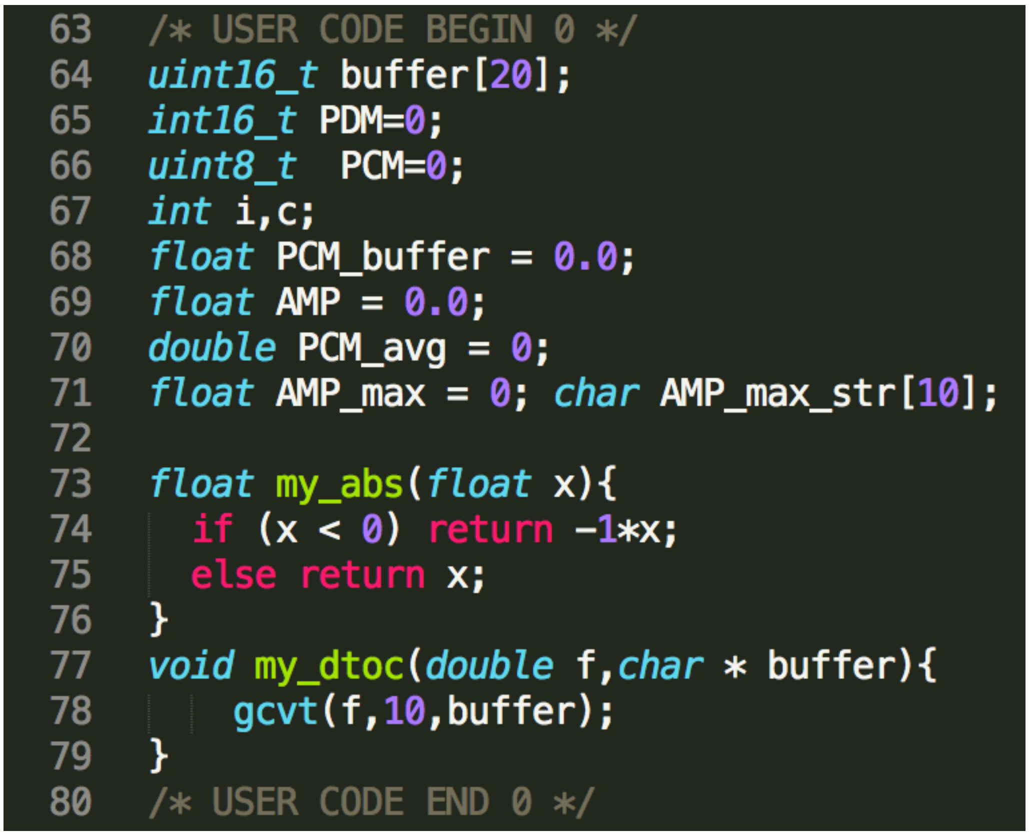


```
/* USART2 init function */
static void MX_USART2_UART_Init(void)
{

  huart2.Instance = USART2;
  huart2.Init.BaudRate = 115200;
  huart2.Init.WordLength = UART_WORDLENGTH_8B;
  huart2.Init.StopBits = UART_STOPBITS_1;
  huart2.Init.Parity = UART_PARITY_NONE;
  huart2.Init.Mode = UART_MODE_TX_RX;
  huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
  huart2.Init.OverSampling = UART_OVERSAMPLING_16;
  if (HAL_UART_Init(&huart2) != HAL_OK)
  {
    Error_Handler();
  }

}
```

UART2 Configuration

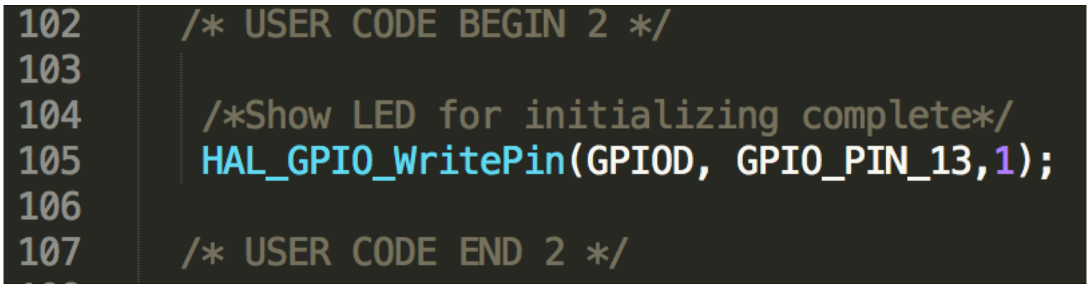
use default configuration
(Baud rate = 115200)

All variable and modified method that used in this project.



```c
/* USER CODE BEGIN 0 */
uint16_t buffer[20];
int16_t PDM=0;
uint8_t  PCM=0;
int i,c;
float PCM_buffer = 0.0;
float AMP = 0.0;
double PCM_avg = 0;
float AMP_max = 0; char AMP_max_str[10];

float my_abs(float x){
  if (x < 0) return -1*x;
  else return x;
}
void my_dtoc(double f,char * buffer){
    gcvt(f,10,buffer);
}
/* USER CODE END 0 */
```

Initial Codes

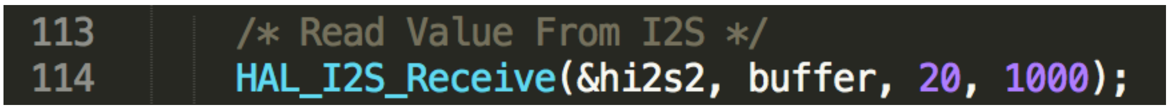


```c
/* USER CODE BEGIN 2 */

 /*Show LED for initializing complete*/
 HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13,1);

/* USER CODE END 2 */
```

Objective : just want to know that it was loaded successfully.

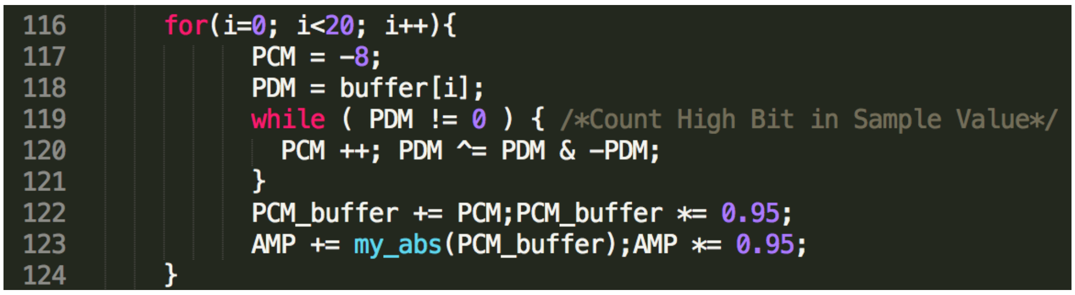
While-Loop Codes

1st Part : Receive a data of surrounded sound in 16 bits * 20 Blocks of data. (contains only high/low bits (1,0))



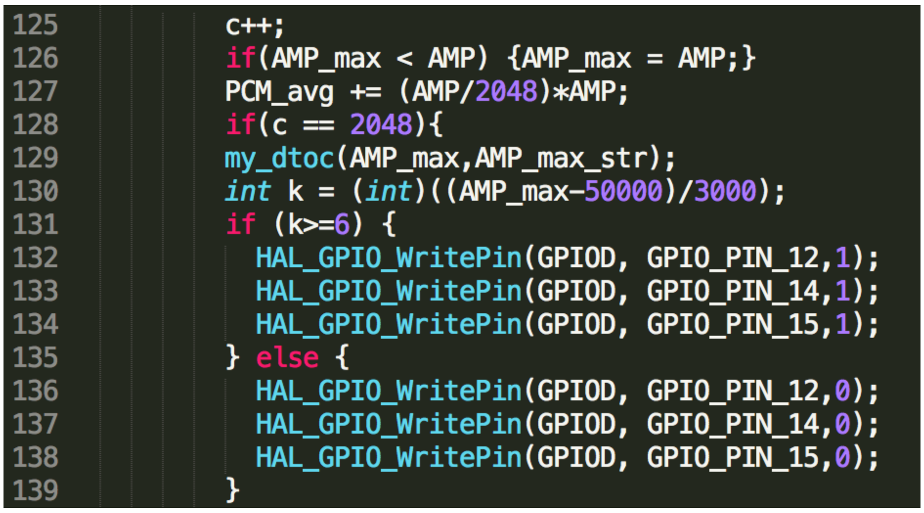
```
/* Read Value From I2S */
HAL_I2S_Receive(&hi2s2, buffer, 20, 1000);
```

2nd Part : Counting a high bits (1) until found a low bits and store it length in to PCM Array.



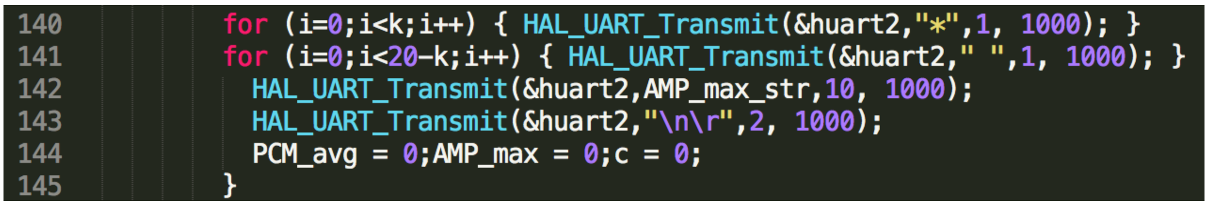
```
for(i=0; i<20; i++){
        PCM = -8;
        PDM = buffer[i];
        while ( PDM != 0 ) { /*Count High Bit in Sample Value*/
          PCM ++; PDM ^= PDM & -PDM;
        }
        PCM_buffer += PCM;PCM_buffer *= 0.95;
        AMP += my_abs(PCM_buffer);AMP *= 0.95;
}
```

3rd Part : Find a maximum value of 2048 block of 20*16 bits PCM values to represent a volume of surrounded sound in a very short period of time (~1 second) and convert in to more understandable scaled. *If it loud enough, all LED will light up.



```
c++;
if(AMP_max < AMP) {AMP_max = AMP;}
PCM_avg += (AMP/2048)*AMP;
if(c == 2048){
my_dtoc(AMP_max,AMP_max_str);
int k = (int)((AMP_max-50000)/3000);
if (k>=6) {
  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12,1);
  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14,1);
  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15,1);
} else {
  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12,0);
  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14,0);
  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15,0);
}
```

4th Part : Send (*) k time (converted value in 3rd Part) through USB by UART protocol to display in serial port terminal (Putty) and also sent unconverted value too, Then reset all variable and repeat all step forever.



```
for (i=0;i<k;i++) { HAL_UART_Transmit(&huart2,"*",1, 1000); }
for (i=0;i<20-k;i++) { HAL_UART_Transmit(&huart2," ",1, 1000); }
  HAL_UART_Transmit(&huart2,AMP_max_str,10, 1000);
  HAL_UART_Transmit(&huart2,"\n\r",2, 1000);
  PCM_avg = 0;AMP_max = 0;c = 0;
}
```

For more technical-info, Please look in MP45DT02 datasheet.