Traffic sign recognition demo with OctaFlash

Rev. 1.0, 2022-12-01

Contents

1. Motivation of this demo

This demo is used to show the application of MXIC OctaFlash in embedded AI scenarios.

With the development of AI technology in the embedded field, the amount of data that needs to be stored in Flash is becoming larger and larger. When running neural network model on embedded platform, it is usually necessary to store and quickly read or write a large amount of weight data. Sometimes you need to read and write other large data, such as pictures and audio. The capacity of internal Flash in MCU is not enough. Moreover, some high-performance MCU does not contain internal Flash, such as NXP RT1170 series. At this time, it is necessary to use large-capacity high-performance external Flash. MXIC OctaFlash is a good choice.

MXIC Octal Serial NOR Flash has a great read-write performance. Its working frequency can reach up to 133MHz, and can work both in STR mode and DTR mode with 8IO. The LM/UM series is a multiple bank architecture solution based on the ultra-high performance Macronix OctaBus™ interface, which dedicates to raising the product performance with the fastest 250MHz frequency, combining with the brand-new Data Transfer Rate (DTR) feature. The data transfer rate has therefore been increased from existing 100MB/s of Quad I/O Serial NOR Flash to 500MB/s, while the read latency has also been lowered immensely. The improved performance will help system in running eXecute In Place (XIP) on octaflash with more efficiency and shorten the access time in high density, which accelerates overall system performance.

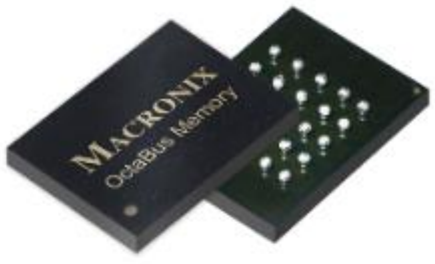


Figure 1-1 MXIC OctaFlash

- This demo is based on evkmimxrt1170_tensorflow_lite_micro_label_image_cm7 (SDK-v2.9.0) example. The new added/renamed file is as Table 1-1 shows.

index	new added/renamed file
1	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\board\display_support.h
2	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\board\emwin_support.c

3	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\board\emwin_support.h
4	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\board\emwin_support.h
5	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\component\i2c\fsl_adapter_i2c.h
6	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\component\i2c\fsl_adapter_i2c.c
7	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\enwin
8	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\music.h
9	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\main.h
10	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\video\30_speed.c
11	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\video\attention_general.c
12	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\eiq
13	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\video\give_way.c
14	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\video\lifted_general.c
15	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\video\no_way_general.c
16	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\video\no_way_one_way.c
17	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\video\right_of_way_general.c
18	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\video\stop.c
19	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\video\turn_right_down.c
20	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\video\turn_straight.c
21	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\image\image_from_eiq.h
22	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\driver\fsl_sai.c
23	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\driver\fsl_sai.h

24	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\video\fsl_hx8394.c
25	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\video\fsl_hx8394.h
26	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\model\flower_labels.h
27	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\source\model\flower_model.h
28	evkmimxrt1170_flower_tensorflow_lite_micro_label_image_cm7\codec

Table 1-1 new added file

2. Introduction to demo

2.1 Functions implemented

This demo uses the NXP MIMXRT1170 development board, which can identify different traffic signs, as shown in Figure 2-1. Place 11 different traffic signs on a turntable with a diameter of about 25 cm. The speed of rotation can be adjusted through the remote control. The camera will continuously take photos and display the captured content on the LCD in real time. The captured content will be processed and then recognized by AI. The recognition result will be displayed in the upper left corner of the LCD, and the audio system will play the corresponding audio of the result.

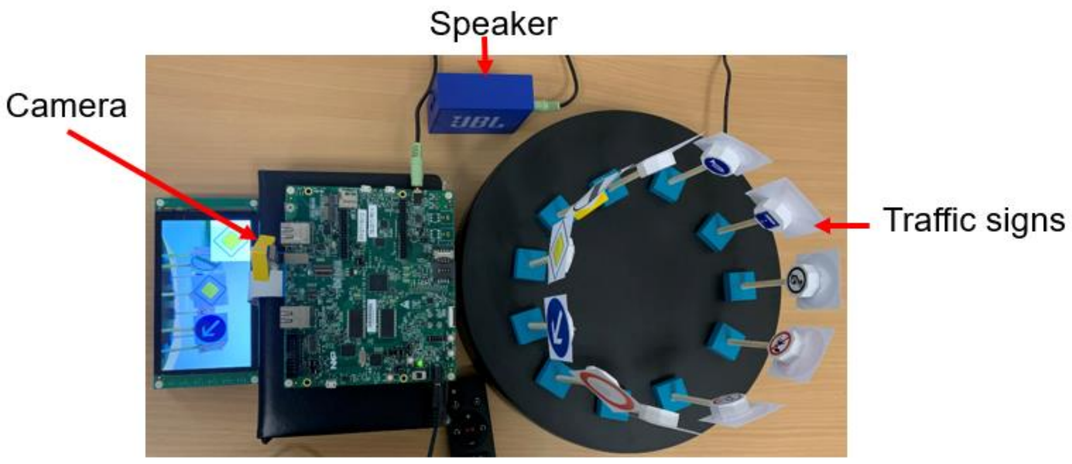


Figure 2-1 Demo working picture

2.2 Hardware components

- *NXP MIMXRT1170 Board mounted with MX25LM51245GXDI00*

- *5.5 inch 720*1280 TFT LCD display with touch sensitive overlay, 2-lane or 4-lane MIPI interface(RK055HDMIPI4M)*

- *Camera Module (OV5640)*

- *Speaker (logitech Z120)*

- *Rotating table with a diameter of 25cm*

- *11 traffic signs*

2.3 Storage Briefing

As the storage device of the whole system, the total capacity of this Flash is 256Mb. Because MCU has no internal flash, the code is stored in MX25LM51245G. The system will boot from MX25LM51245G in XIP mode. The content and size of the storage are shown in the following table2-1.

Storage content	Storage location	Size
Weights of AI Network	MX25LM51245GXDI00	922KB
168x168 RGB image	MX25LM51245GXDI00	899KB
Voice Files	MX25LM51245GXDI00	706KB
Source code	MX25LM51245GXDI00	471KB

Table 2-1 Content and capacity stored

2.4 System Boot-up Procedure

Figure 2-2 shows the Connects used on the development board. The system should be connected before starting.

- Plug the power cord to boot whole system
  - Connect the power cord to the USB connector J43 and system boot-up.
  - After system boot-up, the system will take a picture and start to recognize it.
- Plug the power cord of the rotary table to get the recognition result
  - Turn on the rotary table switch and use the remote control to adjust the direction and speed of rotation.
  - The system will take a picture and process the picture to get the recognition result.

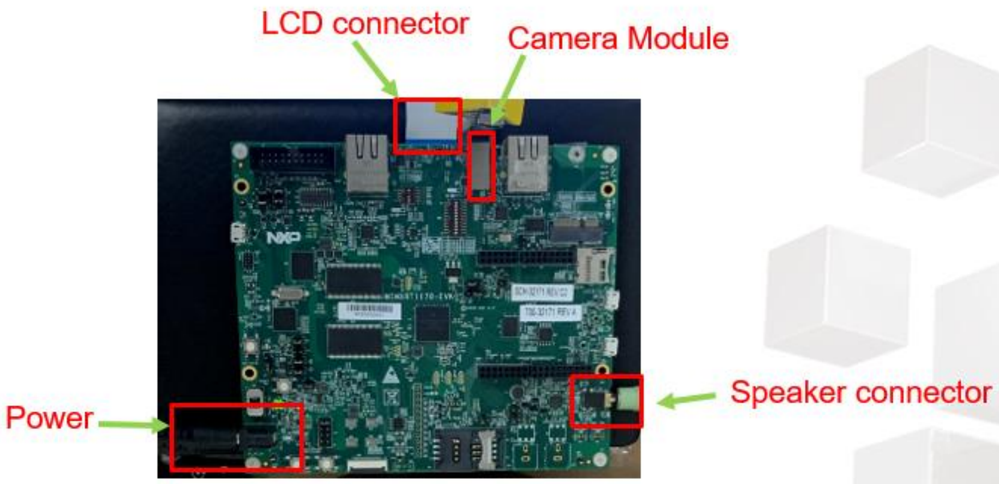


Figure 2-2 Connects used on the development board

2.5 Flow of recognition

- The rotary table rotates at a suitable speed.
- The camera continuously captures traffic signs.
- LCD displays the captured pictures in real time.
- Adjust the distance between the target and the camera to ensure that the target can appear exactly in the flashing width in the middle of the LCD and can cover the entire box.
- The system grabs the pictures in the LCD box and processes them into 128 * 128 pictures.
- Input image data into AI neural network for recognition.
- The predicted result will be displayed in the upper left corner of the LCD, and the speaker will play the predicted audio.

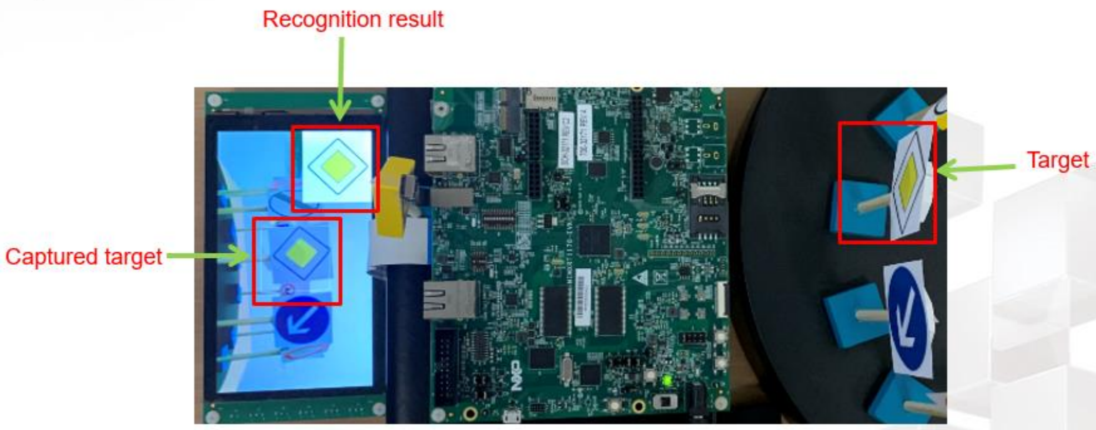
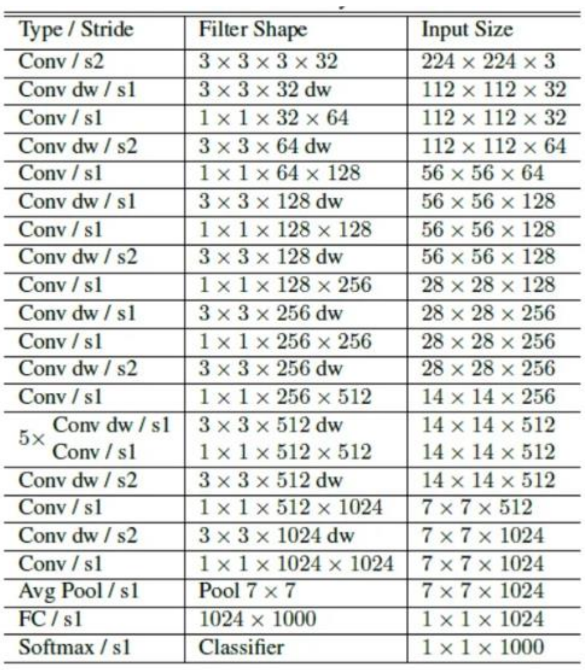


Figure 2-3 Display of recognition result

3. AI Development flow

3.1 Selection of neural network

This demo uses the MobileNet neural network. The following Figure 3-1 shows the structure of a typical MobileNet network. Its input is 224 * 224 pictures, and its output is 1000 categories.
Our system runs on MCU. In order to shorten the prediction time, we use 128 * 128 pictures as input. This demo can recognize 43 traffic signs, so the output is 1 * 1 * 43.



Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figure 3-1 MobileNet body architecture

3.2 Selection of data sets

The goal of this demo is to identify different traffic signs, as shown in the Figure 3-2.

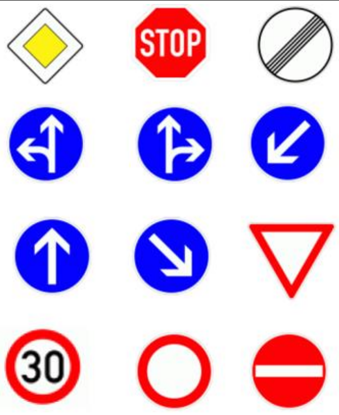


Figure 3-2 Traffic signs

For the training data set, we use the GTSRB (German Traffic Sign Benchmarks) open source on Github, which contains 43 traffic signs. According to their description, we use the Mobilenet neural network to train this data set, and then run it on the PC. The prediction accuracy can reach 98%. Please refer to this link for details. Figure 3-3 lists some predicted results, and the green font indicates that the predicted results are correct.

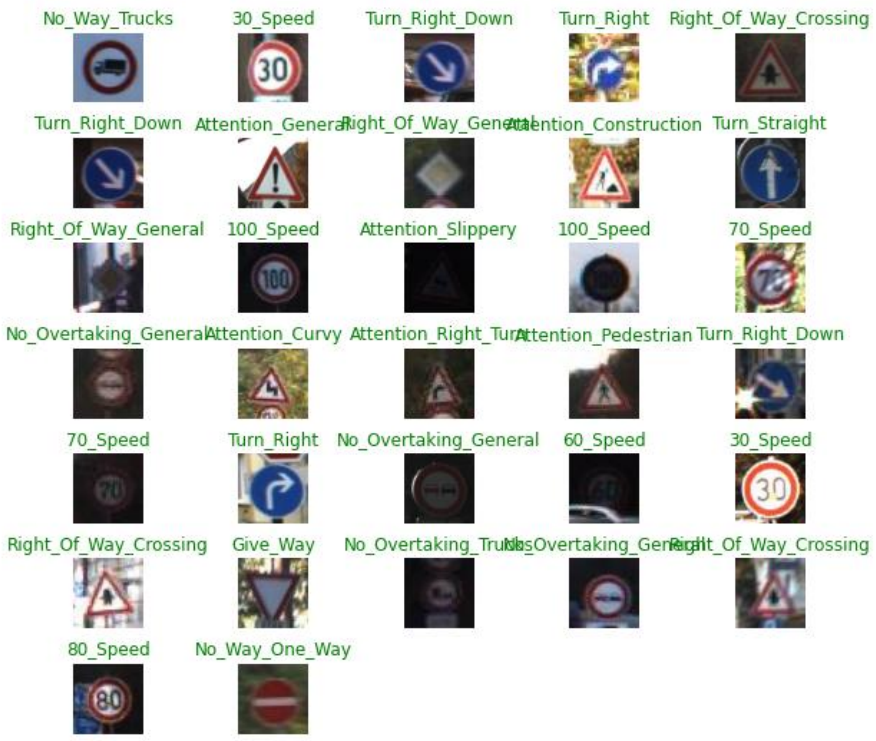
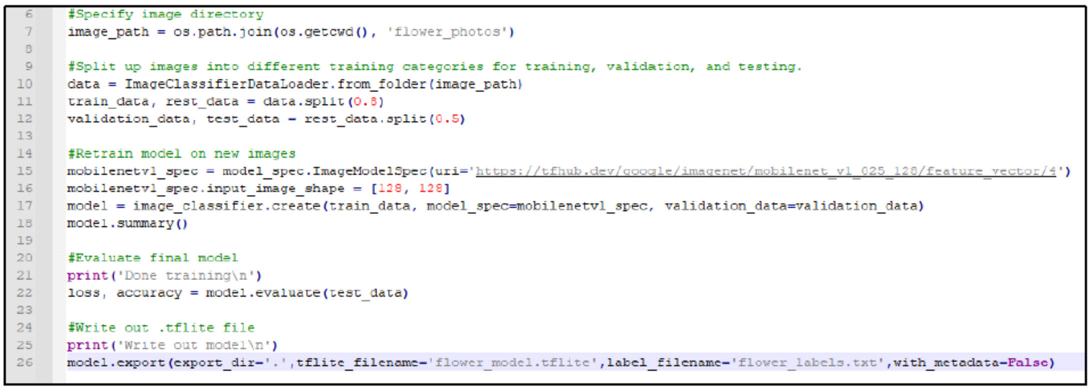


Figure 3-3 Predicted results

3.3 Retrain

For this demo, we refer to an AI demo officially provided by NXP, which contains the detailed development process. Please refer to this link. First, install relevant tools according to the first two chapters of this document. Then refer to the third chapter for retraining. The main content is to run a script, as shown in Figure 3-4, which contains the steps of training and generating TF lite file.

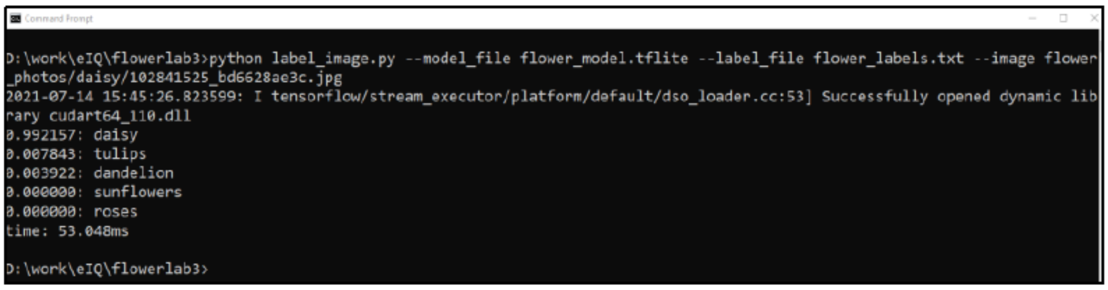


```
#Specify image directory
image_path = os.path.join(os.getcwd(), 'flower_photos')

#Split up images into different training categories for training, validation, and testing.
data = ImageClassifierDataLoader.from_folder(image_path)
train_data, rest_data = data.split(0.8)
validation_data, test_data = rest_data.split(0.5)

#Retrain model on new images
mobilenetv1_spec = model_spec.ImageModelSpec(uri='https://tfhub.dev/google/imagenet/mobilenet_v1_025_128/feature_vector/4')
mobilenetv1_spec.input_image_shape = [128, 128]
model = image_classifier.create(train_data, model_spec=mobilenetv1_spec, validation_data=validation_data)
model.summary()

#Evaluate final model
print('Done training\n')
loss, accuracy = model.evaluate(test_data)

#Write out .tflite file
print('Write out model\n')
model.export(export_dir='.',tflite_filename='flower_model.tflite',label_filename='flower_labels.txt',with_metadata=False)
```

Figure 3-4 Retrain script

We need to download the GTRSB data set image on Github. The original image is in ppm format. We need to convert the image into jpg format through a script. Then replace all the pictures with the original ones. Training generates two files, one is flower_Model.tflite, the other is flower_labels.txt. We can also modify the names of these two files by modifying the names in the last line of the script.

With these two files, you can run label_ Image. py script to test the recognition results. As shown in the Figure 3-5 below.

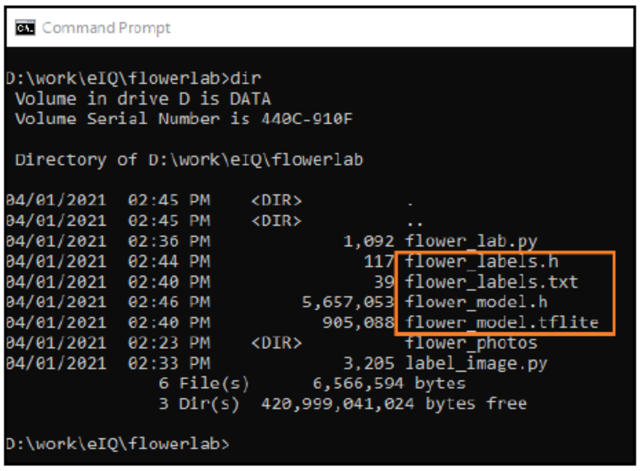


```
D:\work\eIQ\flowerlab3>python label_image.py --model_file flower_model.tflite --label_file flower_labels.txt --image flower
_photos/daisy/102841525_bd6628ae3c.jpg
2021-07-14 15:45:26.823599: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic lib
rary cudart64_110.dll
0.992157: daisy
0.007843: tulips
0.003922: dandelion
0.000000: sunflowers
0.000000: roses
time: 53.048ms

D:\work\eIQ\flowerlab3>
```

Figure 3-5 Run recognition on PC

3.3 Convert TensorFlow model

Refer to Chapter 4 to generate two files, one is *flower_model.h*, the other is *flower_labels.h*, as shown in the Figure 3-6, needs to be put into the MCU project to compile into the final binary code.



```
D:\work\eIQ\flowerlab>dir
 Volume in drive D is DATA
 Volume Serial Number is 440C-910F

 Directory of D:\work\eIQ\flowerlab

04/01/2021  02:45 PM    <DIR>          .
04/01/2021  02:45 PM    <DIR>          ..
04/01/2021  02:36 PM             1,092 flower_lab.py
04/01/2021  02:44 PM               117 flower_labels.h
04/01/2021  02:40 PM                39 flower_labels.txt
04/01/2021  02:46 PM         5,657,053 flower_model.h
04/01/2021  02:40 PM           905,088 flower_model.tflite
04/01/2021  02:23 PM    <DIR>          flower_photos
04/01/2021  02:33 PM             3,205 label_image.py
               6 File(s)      6,566,594 bytes
               3 Dir(s)  420,999,041,024 bytes free

D:\work\eIQ\flowerlab>
```

Figure 3-6 Two final files

4. MCU driver development flow

4.1 New tensor flow project

The final step is to take the TensorFlow Lite Micro Label Image example and modify it to use the newly retrained model. Refer to Chapter 5 of the reference document for the steps of creation and modification.

4.2 MCU software flow

This demo adds some other functions, such as playing audio files and pictures of actual results, so the software needs to be further modified. The overall software flow is shown in Figure 4-1.

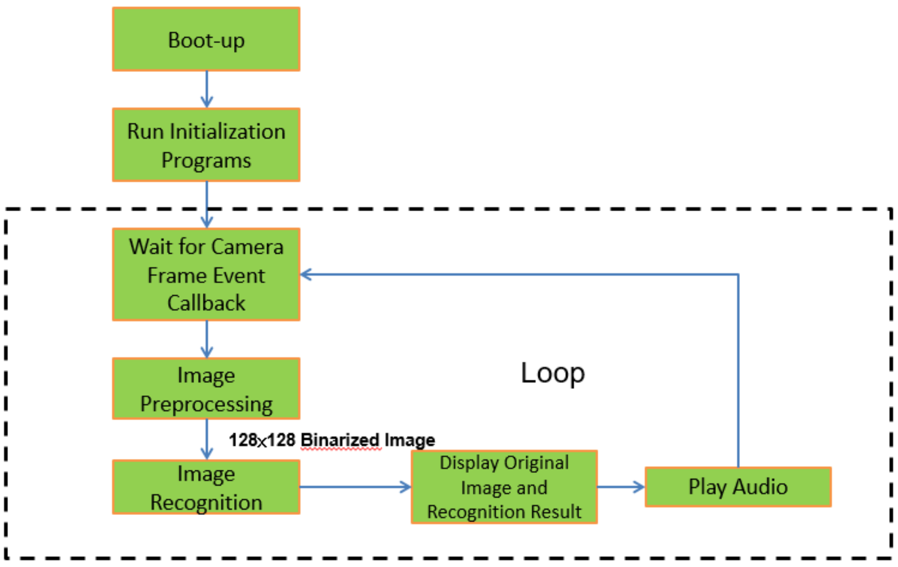

Boot-up
Run Initialization Programs
Wait for Camera Frame Event Callback
Loop
Image Preprocessing
128x128 Binarized Image
Image Recognition
Display Original Image and Recognition Result
Play Audio

Figure 4-1 Software flow

4.3 Picture production process

4.3.1 Get JPG format file

Find the picture corresponding to the traffic sign, as shown in the Figure 4-2 .Use the screenshot tool and save it as a JPG format file

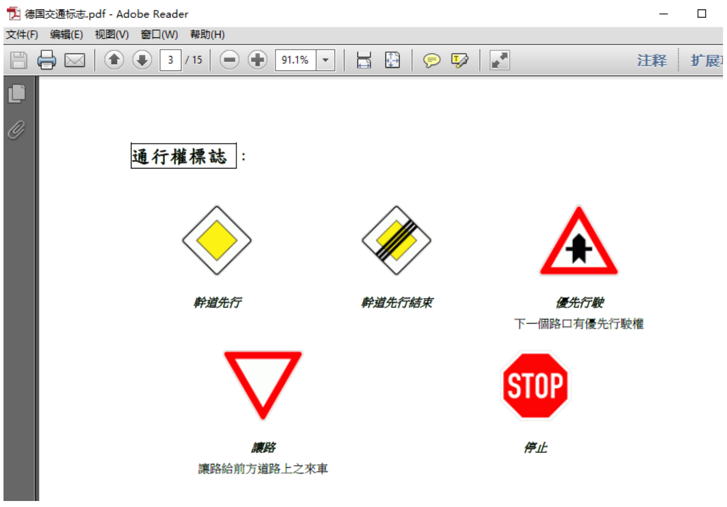


Figure 4-2 Traffic sign

4.3.2 Get bmp format file

Open it with the drawing tool, add the corresponding text, adjust it to the appropriate size, and save it as a file in bmp format, as shown in the Figure 4-3.

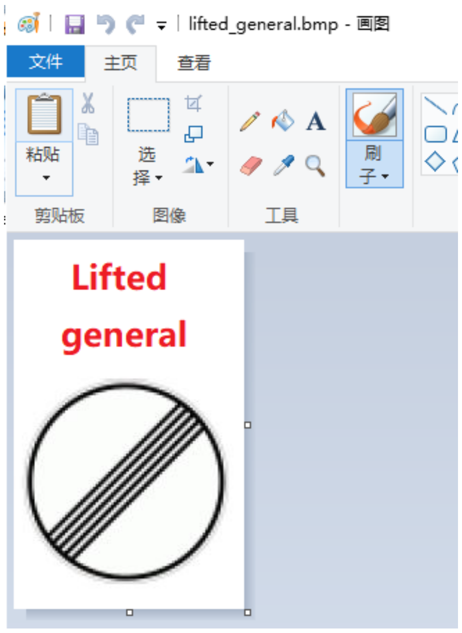


Figure 4-3 Add the corresponding text

4.3.3 Convert bmp file to C file

Use Bin2C tool to import the file in bmp format, as shown in the figure4-4.

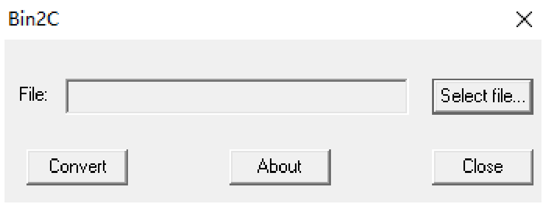


Figure 4-4 Bin2C

Generate the corresponding file in c format, as shown in the Figure 4-5.

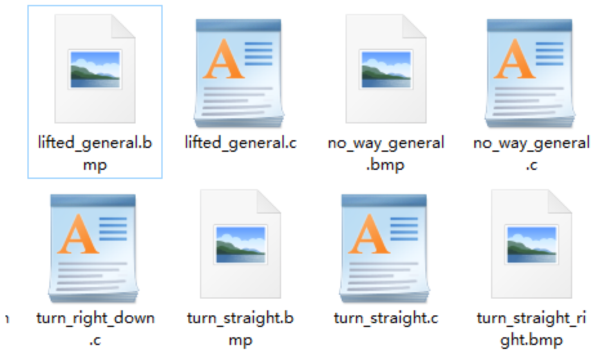


Figure 4-5 C files

4.3.4 Copy the C file to the video directory

Copy the C files of all pictures to the specified path, as shown in Figure 4-6.

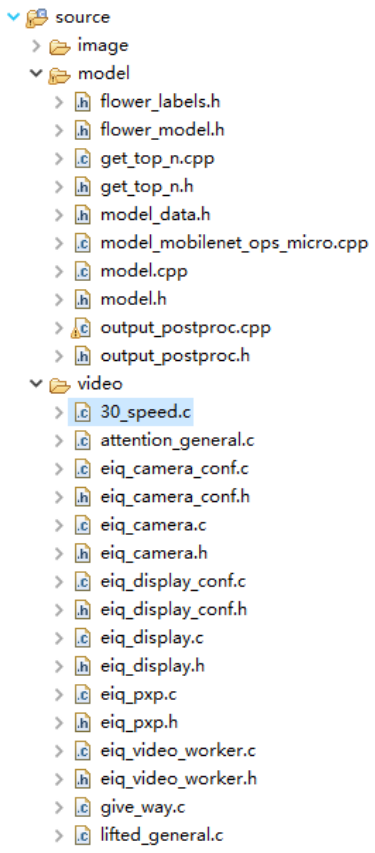


Figure 4-6 C files path

4.3.5 Add code to display pictures

After the image is added, we need to add the code to display the image. This part of code only contains two parts. One is the emwin library, which provides many APIs that can be directly used to display the image or text. Some initialization is required before calling these APIs. The other part is to display the corresponding pictures according to the predicted results. Add the relevant code of the emwin library, as shown in Figure 4-7.

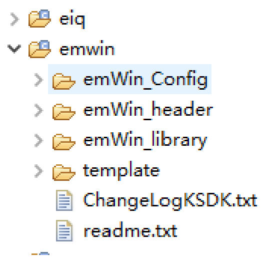
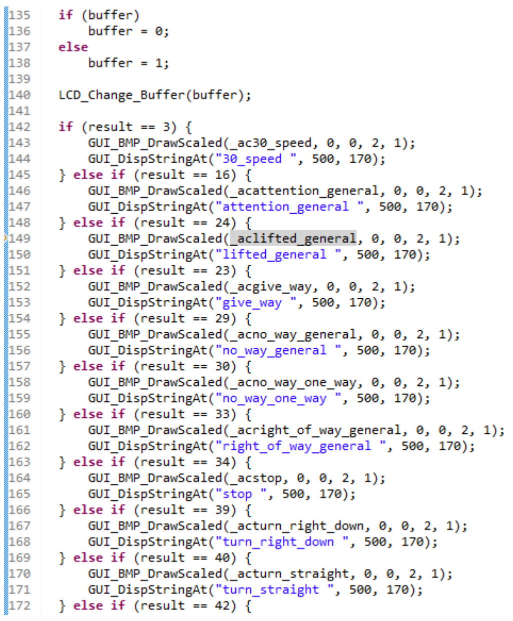


Figure 4-7 emwin library

Add code to display pictures.



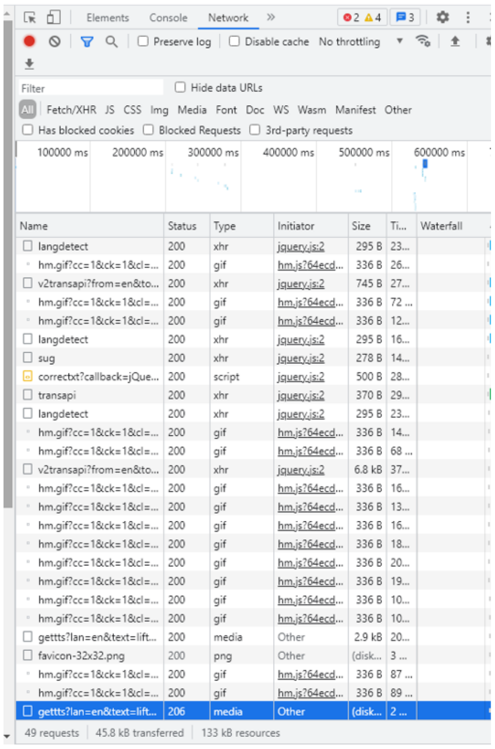
```
if (buffer)
    buffer = 0;
else
    buffer = 1;

LCD_Change_Buffer(buffer);

if (result == 3) {
    GUI_BMP_DrawScaled(_ac30_speed, 0, 0, 2, 1);
    GUI_DispStringAt("30_speed ", 500, 170);
} else if (result == 16) {
    GUI_BMP_DrawScaled(_acattention_general, 0, 0, 2, 1);
    GUI_DispStringAt("attention_general ", 500, 170);
} else if (result == 24) {
    GUI_BMP_DrawScaled(_aclifted_general, 0, 0, 2, 1);
    GUI_DispStringAt("lifted_general ", 500, 170);
} else if (result == 23) {
    GUI_BMP_DrawScaled(_acgive_way, 0, 0, 2, 1);
    GUI_DispStringAt("give_way ", 500, 170);
} else if (result == 29) {
    GUI_BMP_DrawScaled(_acno_way_general, 0, 0, 2, 1);
    GUI_DispStringAt("no_way_general ", 500, 170);
} else if (result == 30) {
    GUI_BMP_DrawScaled(_acno_way_one_way, 0, 0, 2, 1);
    GUI_DispStringAt("no_way_one_way ", 500, 170);
} else if (result == 33) {
    GUI_BMP_DrawScaled(_acright_of_way_general, 0, 0, 2, 1);
    GUI_DispStringAt("right_of_way_general ", 500, 170);
} else if (result == 34) {
    GUI_BMP_DrawScaled(_acstop, 0, 0, 2, 1);
    GUI_DispStringAt("stop ", 500, 170);
} else if (result == 39) {
    GUI_BMP_DrawScaled(_acturn_right_down, 0, 0, 2, 1);
    GUI_DispStringAt("turn_right_down ", 500, 170);
} else if (result == 40) {
    GUI_BMP_DrawScaled(_acturn_straight, 0, 0, 2, 1);
    GUI_DispStringAt("turn_straight ", 500, 170);
} else if (result == 42) {
```

Figure 4-7 Show pictures

4.4 Audio production process

4.4.1 Download MP3 files of audio

Download the corresponding mp3 file from the translation website. Double-click the file in media format to download it automatically, as shown in the Figure 4-8.


Elements Console Network | 2 4 | 3

Preserve log | Disable cache | No throttling

Filter | Hide data URLs

All | Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

Has blocked cookies | Blocked Requests | 3rd-party requests

100000 ms 200000 ms 300000 ms 400000 ms 500000 ms 600000 ms

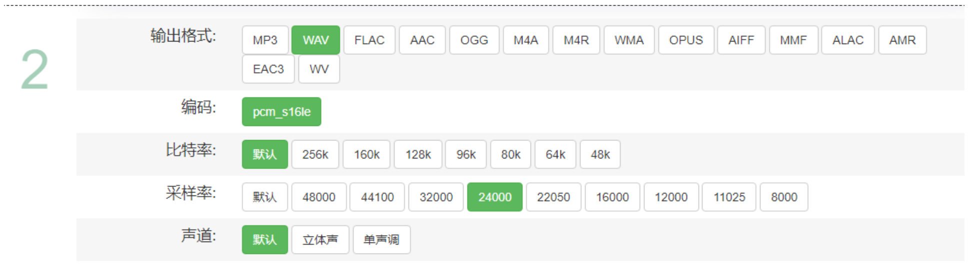
Name	Status	Type	Initiator	Size	Ti...	Waterfall
langdetect	200	xhr	jquery.js:2	295 B	23...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	26...	
v2transapi?from=en&to...	200	xhr	jquery.js:2	745 B	27...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	72 ...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	12...	
langdetect	200	xhr	jquery.js:2	295 B	16...	
sug	200	xhr	jquery.js:2	278 B	14...	
correctxt?callback=jQue...	200	script	jquery.js:2	500 B	28...	
transapi	200	xhr	jquery.js:2	370 B	29...	
langdetect	200	xhr	jquery.js:2	295 B	23...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	14...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	68 ...	
v2transapi?from=en&to...	200	xhr	jquery.js:2	6.8 kB	37...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	16...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	13...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	16...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	18...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	20...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	19...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	10...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	10...	
gettts?lan=en&text=lift...	200	media	Other	2.9 kB	20...	
favicon-32x32.png	200	png	Other	(disk...	3 ...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	87 ...	
hm.gif?cc=1&ck=1&cl=...	200	gif	hm.js?64ecd...	336 B	89 ...	
gettts?lan=en&text=lift...	206	media	Other	(disk...	2 ...	

49 requests | 45.8 kB transferred | 133 kB resources

Figure 4-8 Download MP3 files

4.4.2 Convert mp3 file to wmv file

Upload the downloaded mp3 file to the format conversion website, change the sampling rate to 24000, and default other parameters, as shown in the following Figure 4-9.


2

输出格式: MP3 [✓ WAV] FLAC AAC OGG M4A M4R WMA OPUS AIFF MMF ALAC AMR EAC3 WV

编码: [✓ pcm_s16le]

比特率: [✓ 默认] 256k 160k 128k 96k 80k 64k 48k

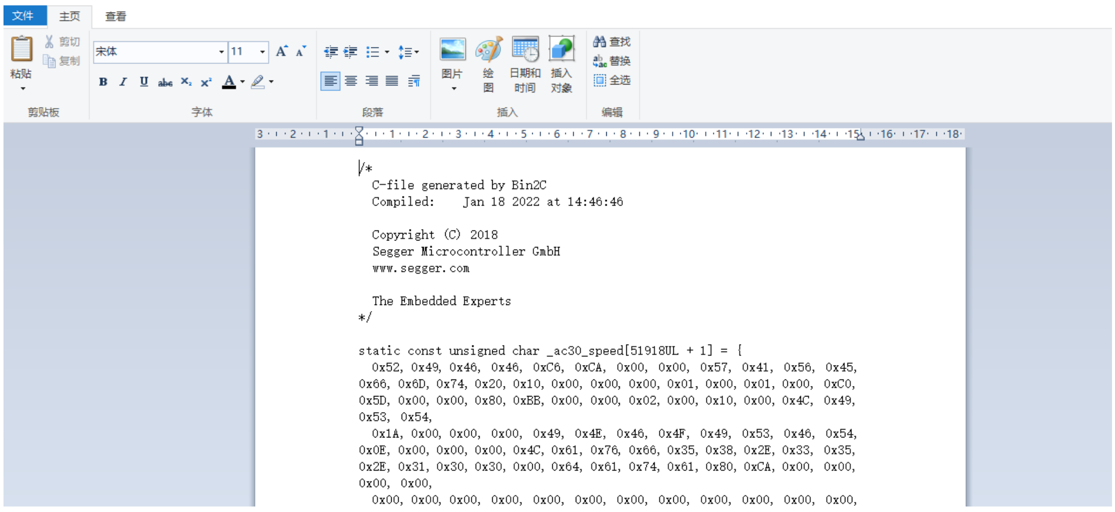
采样率: 默认 48000 44100 32000 [✓ 24000] 22050 16000 12000 11025 8000

声道: [✓ 默认] 立体声 单声调

Figure 4-9 Convert mp3 file to wmv file

4.4.3 Convert wmv file to C file

Download the converted wmv file, import it into Bin2C tool, and generate the corresponding file in c format. Its content is shown in Figure 4-10.

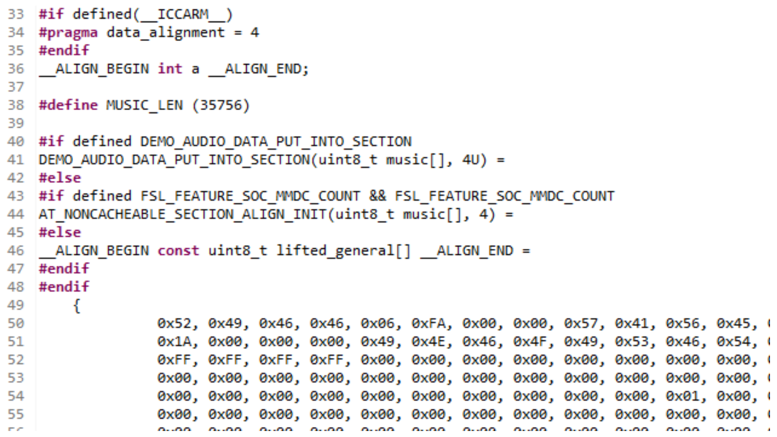


```
/*
  C-file generated by Bin2C
  Compiled:    Jan 18 2022 at 14:46:46

  Copyright (C) 2018
  Segger Microcontroller GmbH
  www.segger.com

  The Embedded Experts
*/

static const unsigned char _ac30_speed[51918UL + 1] = {
  0x52, 0x49, 0x46, 0x46, 0xC6, 0xCA, 0x00, 0x00, 0x57, 0x41, 0x56, 0x45,
0x66, 0x6D, 0x74, 0x20, 0x10, 0x00, 0x00, 0x00, 0x01, 0x00, 0x01, 0x00, 0xC0,
0x5D, 0x00, 0x00, 0x80, 0xBB, 0x00, 0x00, 0x02, 0x00, 0x10, 0x00, 0x4C, 0x49,
0x53, 0x54,
  0x1A, 0x00, 0x00, 0x00, 0x49, 0x4E, 0x46, 0x4F, 0x49, 0x53, 0x46, 0x54,
0x0E, 0x00, 0x00, 0x00, 0x4C, 0x61, 0x76, 0x66, 0x35, 0x38, 0x2E, 0x33, 0x35,
0x2E, 0x31, 0x30, 0x30, 0x00, 0x64, 0x61, 0x74, 0x61, 0x80, 0xCA, 0x00, 0x00,
0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

Figure 4-10 C file content of audio

4.4.4 Copy the C file to the video directory

Create a *music.h* file and add the table contents corresponding to all audio files, as shown in Figure 4-11.

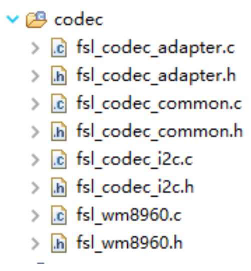


```
#if defined(__ICCARM__)
#pragma data_alignment = 4
#endif
__ALIGN_BEGIN int a __ALIGN_END;

#define MUSIC_LEN (35756)

#if defined DEMO_AUDIO_DATA_PUT_INTO_SECTION
DEMO_AUDIO_DATA_PUT_INTO_SECTION(uint8_t music[], 4U) =
#else
#if defined FSL_FEATURE_SOC_MMDC_COUNT && FSL_FEATURE_SOC_MMDC_COUNT
AT_NONCACHEABLE_SECTION_ALIGN_INIT(uint8_t music[], 4) =
#else
__ALIGN_BEGIN const uint8_t lifted_general[] __ALIGN_END =
#endif
#endif
    {
                0x52, 0x49, 0x46, 0x46, 0x06, 0xFA, 0x00, 0x00, 0x57, 0x41, 0x56, 0x45,
                0x1A, 0x00, 0x00, 0x00, 0x49, 0x4E, 0x46, 0x4F, 0x49, 0x53, 0x46, 0x54,
                0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00,
                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

Figure 4-11 Header file containing audio files

4.4.5 Add code to play sound

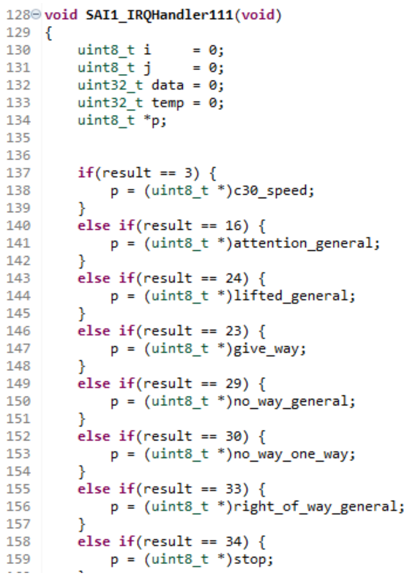
The code for playing audio is also divided into two parts. One part is the underlying driver. Because the SAI controller is used, its corresponding driver should be added, as shown in the Figure 4-12.



```
codec
  fsl_codec_adapter.c
  fsl_codec_adapter.h
  fsl_codec_common.c
  fsl_codec_common.h
  fsl_codec_i2c.c
  fsl_codec_i2c.h
  fsl_wm8960.c
  fsl_wm8960.h
```

Figure 4-12 Driver of SAI controller

The other part is to add the controller interrupt function, in which the corresponding audio is played according to the predicted result, as shown in the Figure 4-13.



```c
void SAI1_IRQHandler111(void)
{
    uint8_t i     = 0;
    uint8_t j     = 0;
    uint32_t data = 0;
    uint32_t temp = 0;
    uint8_t *p;


    if(result == 3) {
        p = (uint8_t *)c30_speed;
    }
    else if(result == 16) {
        p = (uint8_t *)attention_general;
    }
    else if(result == 24) {
        p = (uint8_t *)lifted_general;
    }
    else if(result == 23) {
        p = (uint8_t *)give_way;
    }
    else if(result == 29) {
        p = (uint8_t *)no_way_general;
    }
    else if(result == 30) {
        p = (uint8_t *)no_way_one_way;
    }
    else if(result == 33) {
        p = (uint8_t *)right_of_way_general;
    }
    else if(result == 34) {
        p = (uint8_t *)stop;
```

Figure 4-13 Interrupt function of SAI controller

Revision History

Revision No.	Date	Description
1.0	1-DEC-2022	Initial release.

Macronix International Co., Ltd.