

M480 Keyword spotting

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	本範例代碼使用機器學習的方式在 M480 上實現關鍵詞辨識 (Keyword spotting) 功能
BSP 版本	M480 Series BSP CMSIS V3.04.000
開發平台	NuMaker-PFM-M487 Ver 3.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 功能介紹

1.1 簡介

一套完整的深度學習語音辨識系統使用兩個平台完成如圖1-1，一為PC端，利用TensorFlow與Python撰寫完整的深度學習程式碼並訓練模型，因本方案使用的學習模式為監督式學習，因此需給系統大量的訓練資料和標籤(Labels)，接著將擷取到的特徵用深度神經網路(Deep Neural Networks, DNN)模型來訓練，並反覆修正訓練的模型，直到模型達到系統最佳化的狀態；二為NuMaker-PFM-M487平台，使用NuMicro® M480系列晶片下的NuMaker-PFM-M487開發版，利用PC建出來的深度學習模型與訓練結果(特徵參數)，應用到NuMaker-PFM-M487平台來完成可即時的語音辨識系統。

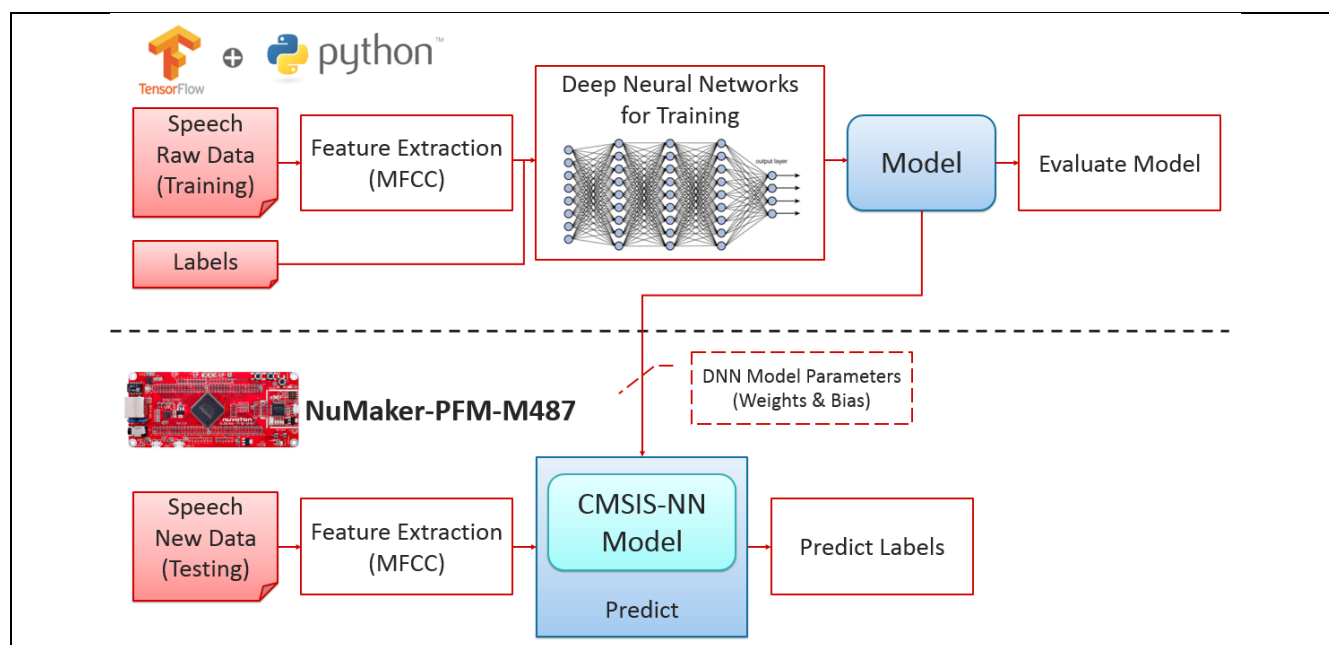


圖 1-1 語音辨識系統流程圖

更詳細說明，請參考應用手冊：

AN_0029_M480series_SpeechRecognition_MachineLearning_Rev1.00

1.2 深度學習介紹

機器學習(Machine Learning)是人工智慧(Artificial Intelligence, AI)的分支，運作流程是透過演算法，使用大量資料進行訓練，訓練完成後會產生模型，當未來有新的資料時，我們可以使用訓練產生的模型進行預測。機器學習應用相當廣泛，例如：推薦引擎、定向廣告、需求預測、

垃圾郵件過濾、醫學診斷、自然語言處理、搜索引擎、詐騙偵測、證券分析、視覺辨識、語音識別、手寫辨識等等。

而深度學習(Deep Learning)又是機器學習的分支，是人工智慧中，成長最快的領域，至今已有數種深度學習的框架，如深度神經網路(Deep Neural Networks, DNN)、卷積神經網路(Convolutional Neural Networks, CNN)、遞迴神經網路(Recurrent Neural Networks, RNN)等等，何時該用什麼樣的架構，答案是沒有一定的，需要依據數據的大小、性質、可接受的計算時間、學習的緊迫性、想用這些數據做什麼來判斷。深度學習特別在視覺辨識、語音識別、自然語言處理、生物醫學等領域的應用上，有非常好的效果。

深度學習就是讓機器模擬人腦的運作方式，進而和人類一樣具備學習的能力，不過由於人類神經網路太過複雜，所以為了方便以機器模擬，將神經元分為多層次，來模擬神經網路，神經網路通常會有1個輸入層(Input Layer)、1個輸出層(Output Layer)和多個可以被訓練的隱藏層(Hidden Layer)，超過2個隱藏層就可以被稱為深度學習。

1.2.1 深度神經網路(Deep Neural Networks, DNN)介紹

深度神經網路(Deep Neural Networks, DNN)是深度學習最基本的一種判別模型，可以使用反向傳播(Backpropagation)演算法進行訓練，權重(Weight)更新可以使用下式進行梯度下降法(Gradient descent)迭代式進行更新：

$$\Delta w_{ij}(t+1) = \Delta w_{ij}(t) + \eta \frac{\partial C}{\partial w_{ij}}$$

其中， η 為學習率(Learning Rate)， C 為損失函數(Loss Function)，函式的選擇與活化函數(Activation Function)相關，以本文件舉例，為了在一個多分類(Multi Class Classification)問題上進行監督式學習，選擇使用線性整流函數(Rectified Linear Unit, ReLU)作為活化函數，使用交叉熵(Cross Entropy)作為損失函數，交叉熵的定義如下：

$$C = - \sum_j d_j \log(p_j)$$

其中 d_j 代表輸出單元 j 的目標機率， p_j 代表應用活化函數後對單元 j 的機率輸出；而輸出層的歸一化指數函數(Softmax Function)定義如下：

$$p_j = \frac{\exp(x_i)}{\sum_k \exp(x_k)}$$

其中 p_j 代表類別 j 的機率，而 x_i 和 x_k 分別是對單元 i 和 k 的輸入。

深度神經網路的網路架構如圖1-2。

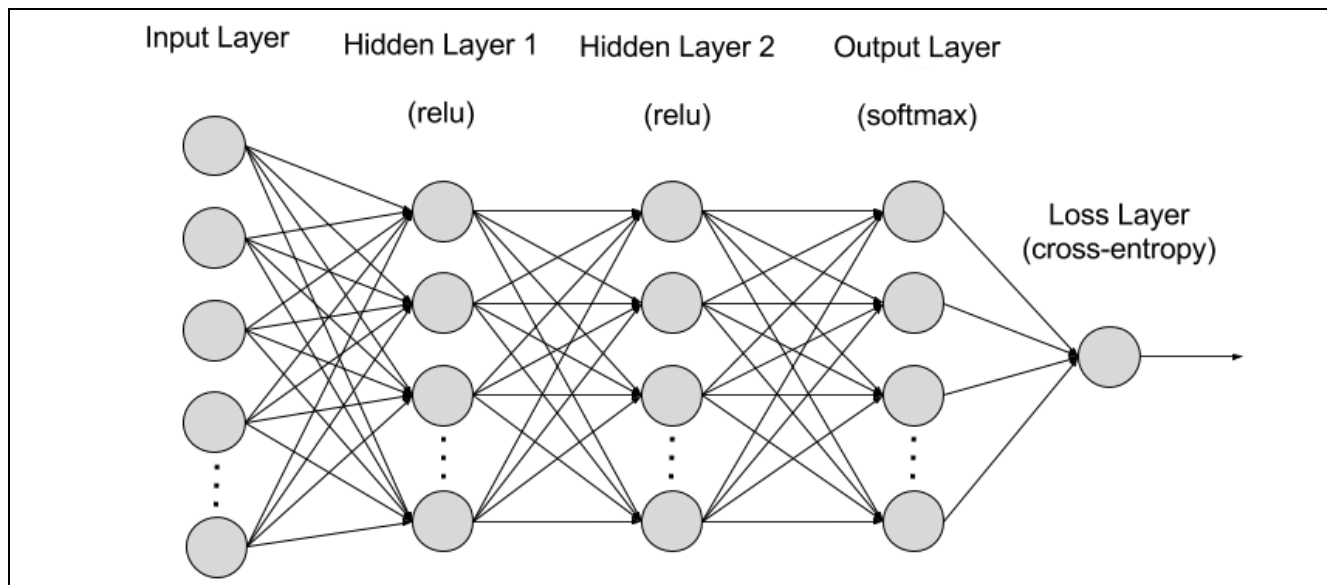


圖 1-2 深度學習網路(DNN)架構圖

1.3 TensorFlow 簡介

TensorFlow是由Google提供的開放原始程式碼程式庫，Google有很多產品使用TensorFlow技術開發深度學習與機器學習功能，例如：Gmail過濾垃圾信、Google語音辨識、Google 圖片辨識、Google翻譯...等等，前一小節深度學習的介紹，我們了解深度學習的核心，以張量(矩陣)運算模擬神經網路，因此TensorFlow的主要設計就是讓矩陣運算達到最高效能，並且能夠在不同的平台上開發。

1.3.1 TensorFlow 架構圖

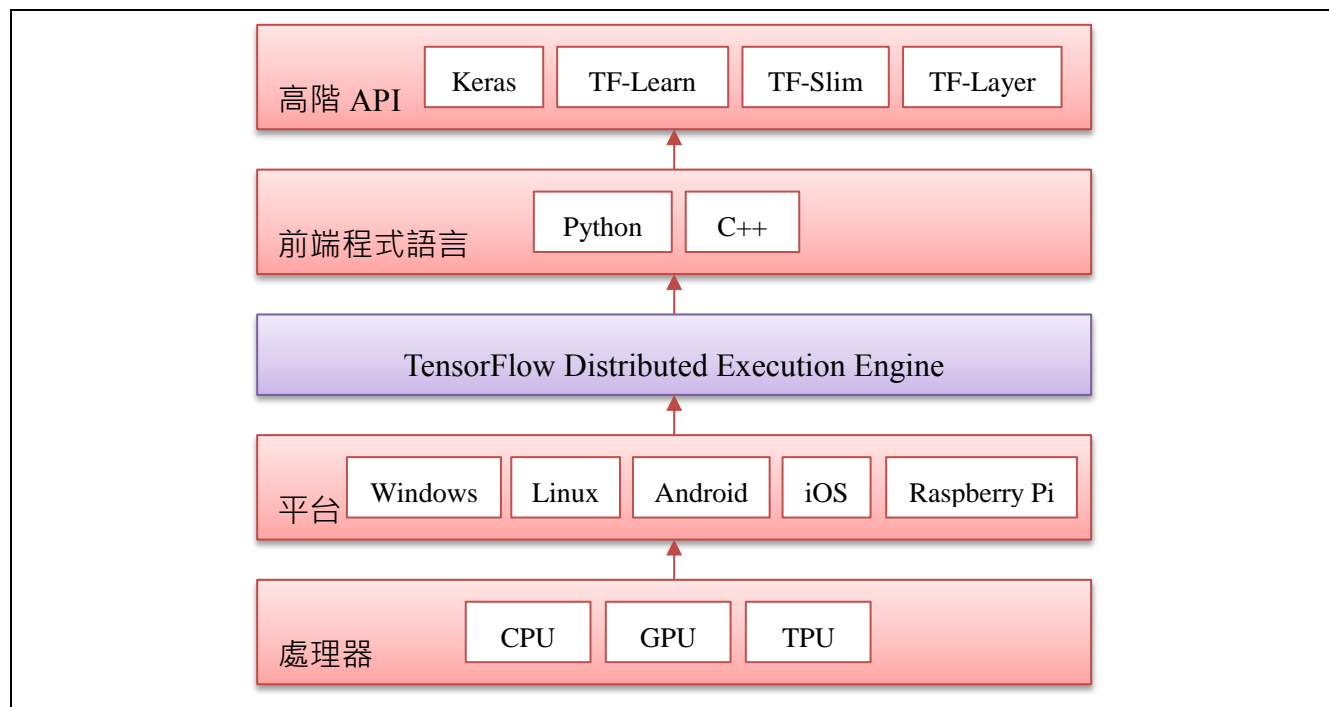


圖 1-3 TensorFlow 架構圖

以上架構圖，由下而上說明如下：

- 處理器：TensorFlow 可以在 CPU、GPU、TPU 上執行
 - CPU：每一台電腦都有中央處理器(CPU)，可以執行 TensorFlow，本語音辨識的應用使用 CPU 即可。
 - GPU：圖形處理器，有數千個小型且高效率的核心，可以發揮平行運算的強大功能。
 - TPU：Tensor Processing Unit 是 google 為人工智慧(AI)研發專屬的晶片，比 GPU 有更好的執行功能。

➤ 平台：

TensorFlow 具跨平台能力，可在目前主流的平台執行，本文件使用 Windows 10 作業系統。

➤ TensorFlow Distributed Execution Engine(分散式執行引擎)：

在深度學習中，最耗時間就是模型的訓練，尤其大型的深度學習模型，必須使用大量資料進行訓練，TensorFlow具備分散式運算能力，可同時在數百台的機器上執行模型訓練，大幅縮短模型訓練的時間。

➤ 前端程式語言：

TensorFlow可使用多種語言，而Python具有程式碼簡明、易學習、高生產力的特質、物件導向、函數式的動態語言，應用非常廣，本文件的深度學習語言也是使用Python來開發。

➤ 高階 API：

TensorFlow是比較低階的深度學習API，所以程式設計模型時，必須自行設計張量乘積、卷積等底層操作，因此本文件搭配高階的API—Keras，讓開發者使用更簡潔更可讀性的程式碼，就可以建構出各種複雜的深度學習模型。

1.4 執行結果

接上麥克風並說出英文數字及會打印出辨識之結果。

```
接收區：已接收5720位元組，速度2751
I2C clock 100000 Hz
NAU88L25 Software Reset.
NAU88L25 Configured done.
Detected Silence (12%)
Detected 7 (16%)
Detected 7 (35%)
Detected 7 (44%)
Detected 7 (35%)
Detected 7 (25%)
Detected 7 (29%)
Detected 7 (42%)
Detected 7 (43%)
Detected 7 (31%)
Detected 7 (23%)
Detected 7 (16%)
Detected 7 (17%)
Detected Silence (13%)
Detected Silence (30%)
Detected Silence (53%)
Detected Silence (60%)
Detected Silence (69%)
Detected Silence (62%)
```

2 代碼介紹

下圖為NuMicro® M480系列微控制器的程式流程圖，圖中方塊為程式中主要的程式，使用者可以由圖簡易地了解程式的架構。

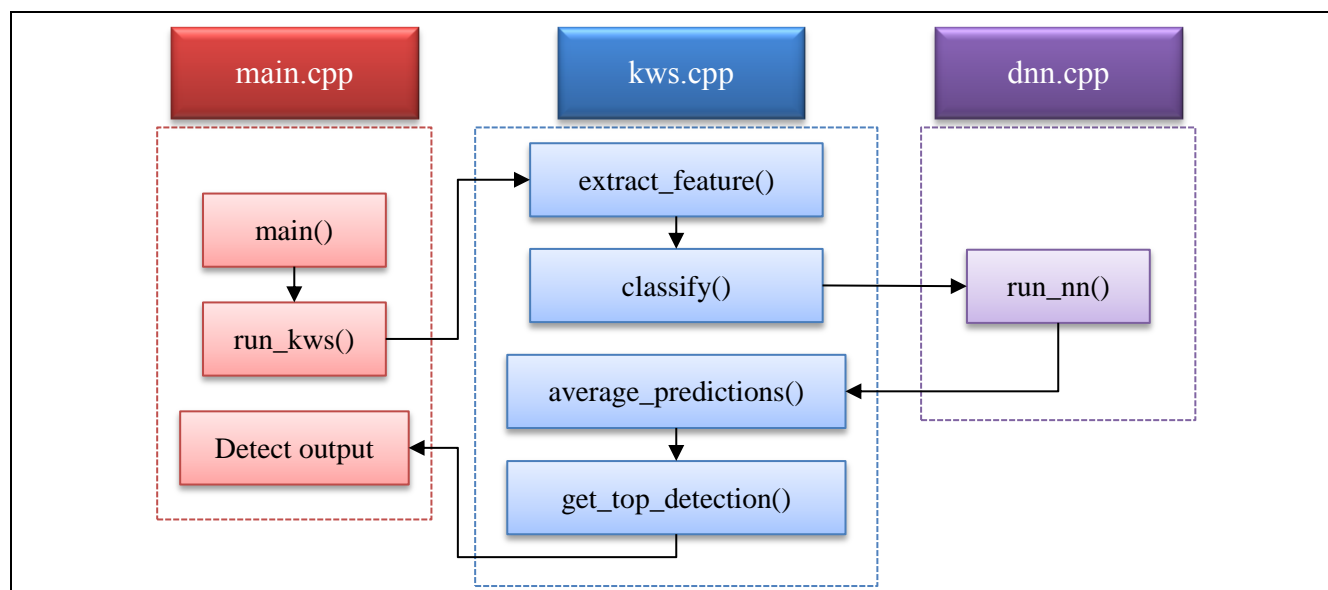


圖 2-1 NuMicro® M480 微控制器程式流程圖

2.1.1 KWS 範例主程式

```

/* Output Class */
char output_class[12][10] = { "Silence", "Unknown", "One", "Two", "Three", "Four",
    "Five", "Six", "Seven", "Eight", "Nine", "Zero"};
void run_kws()
{
    //Averaging window for smoothing out the output predictions
    int averaging_window_len = 3; //i.e. average over 3 inferences or 240ms

    kws->extract_features(2); //extract mfcc features
    kws->classify();          //classify using dnn
    kws->average_predictions(averaging_window_len);
    max_ind = kws->get_top_detection(kws->averaged_output);

    printf("Detected %s\r\n",output_class[max_ind]);
}
    
```

2.1.2 KWS Function (使用 MFCC 特徵擷取和 DNN 模型)

```
/* This overloaded function is used in streaming audio case */
void KWS::extract_features(uint16_t num_frames)
{
    //move old features left
    memmove(mfcc_buffer, mfcc_buffer+(num_frames*NUM_MFCC_COEFFS), (NUM_FRAMES-
num_frames)*NUM_MFCC_COEFFS);
    //compute features only for the newly recorded audio
    int32_t mfcc_buffer_head = (NUM_FRAMES-num_frames)*NUM_MFCC_COEFFS;
    for (uint16_t f = 0; f < num_frames; f++) {
        mfcc->mfcc_compute(audio_buffer+(f*FRAME_SHIFT), 2, &mfcc_buffer[mfcc_buffer_head]);
        mfcc_buffer_head += NUM_MFCC_COEFFS;
    }
}

void KWS::classify()
{
    nn->run_nn(mfcc_buffer, output);

    // Softmax
    arm_softmax_q7(output, OUT_DIM, output);

    //do any post processing here
}

void KWS::average_predictions(int window_len)
{
    //shift right old predictions
    for(int i=window_len-1; i>0; i--) {
        for(int j=0; j<OUT_DIM; j++)
            predictions[i][j]=predictions[i-1][j];
    }
    //add new predictions
    for(int j=0; j<OUT_DIM; j++)
        predictions[0][j]=output[j];
    //compute averages
    int sum;
    for(int j=0; j<OUT_DIM; j++) {
        sum=0;
        for(int i=0; i<window_len; i++)
            sum += predictions[i][j];
        averaged_output[j] = (q7_t)(sum/window_len);
    }
}
```



```

    }
}

int KWS::get_top_detection(q7_t* prediction)
{
    int max_ind=0;
    int max_val=-128;
    for(int i=0;i<OUT_DIM;i++) {
        if(max_val<prediction[i]) {
            max_val = prediction[i];
            max_ind = i;
        }
    }
    return max_ind;
}

```

2.1.3 NN Function

```

void DNN::run_nn(q7_t* in_data, q7_t* out_data)
{
    // Run all layers
    // IP1
    arm_fully_connected_q7(in_data, ip1_wt, IN_DIM, IP1_OUT_DIM, 1, 7, ip1_bias, ip1_out,
vec_buffer);
    // RELU1
    arm_relu_q7(ip1_out, IP1_OUT_DIM);
    // IP2
    arm_fully_connected_q7(ip1_out, ip2_wt, IP1_OUT_DIM, IP2_OUT_DIM, 2, 8, ip2_bias, ip2_out,
vec_buffer);
    // RELU2
    arm_relu_q7(ip2_out, IP2_OUT_DIM);
    // IP3
    arm_fully_connected_q7(ip2_out, ip3_wt, IP2_OUT_DIM, IP3_OUT_DIM, 2, 9, ip3_bias, ip3_out,
vec_buffer);
    // RELU3
    arm_relu_q7(ip3_out, IP3_OUT_DIM);
    // IP4
    arm_fully_connected_q7(ip3_out, ip4_wt, IP3_OUT_DIM, OUT_DIM, 0, 6, ip4_bias, out_data,
vec_buffer);
}

```

3 軟體與硬體環境

- 軟體環境

- BSP 版本

- ◆ M480 Series BSP CMSIS V3.04.000

- IDE 版本

- ◆ Keil uVersion 5.26

- 硬體環境

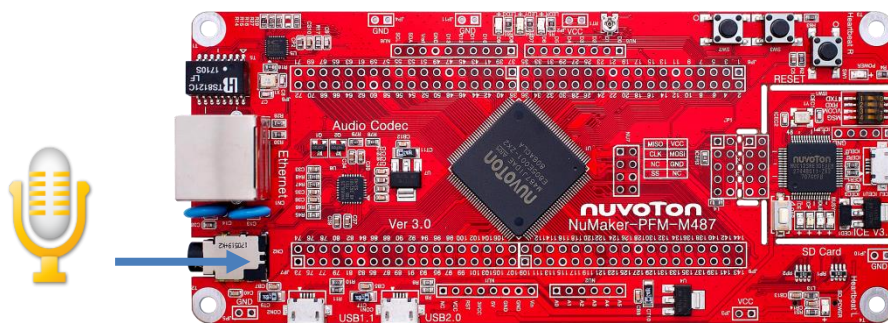
- 電路元件

- ◆ NuMaker-PFM-M487 or other M480 Development Board

- ◆ 3.5 mm 接頭耳機

- 示意圖

在NuMaker-PFM-M487的CN2插入麥克風。



4 目錄資訊

EC_M480_KWS_V1.00

Library

Sample code header and source files

CMSIS

Cortex[®] Microcontroller Software Interface Standard (CMSIS) by Arm[®] Corp.

Device

CMSIS compliant device header file

StdDriver

All peripheral driver header and source files

ML_PCTool

Machine Learning python source files

SampleCode

ExampleCode

Source file of example code

5 如何執行範例程式

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 M480_KWS.uvproj
2. 進入編譯模式介面
 - a. 編譯
 - b. 下載代碼至記憶體
 - c. 進入 / 離開除錯模式
3. 進入除錯模式介面
 - a. 執行代碼

6 修訂紀錄

Date	Revision	Description
Oct. 23, 2019	1.00	1. 初始發布.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

*Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*