Voice recognition wheelchair control system using STM32-based system

Wong Yoke Theng
Faculty of Engineering
Universiti Teknologi Malaysia
Skudai 81310 Malaysia
yokewong@graduate.utm.my

Yew Puay Yu
Faculty of Engineering
Universiti Teknologi Malaysia
Skudai 81310 Malaysia
yewyu@graduate.utm.my

Eric Kerk Hong Ye
Faculty of Engineering
Universiti Teknologi Malaysia
Skudai 81310 Malaysia
hongeric@graduate.utm.my

***Abstract*—This article introduces a speech recognition wheelchair with STM32-based system to ease the wheelchair user to control their wheelchair through sound module. The voice recognition system is embedded in the control system using STM32. The system will execute the movement of wheelchair based on the command of user. For current project, the system can recognize four keywords such as left, right, forward and backward. This voice recognition system will ease the people with physical disability especially with upper limbs disability where they could not control the wheelchair by his/her own or others help to execute the movement.**

***Index Terms*—microcontroller, embedded systems, edge computing, STM32, speech recognition wireless communication**

I. INTRODUCTION

Voice is the simplest and most straight forward way of conveying command and to communicate by human being. Voice recognition is used in a lot of applications nowadays. Speech recognition system embedded in wheelchair is beneficial to the disabled users to ensure they could easily control the wheelchair without the help of others or the need to control the wheelchair through manual way (remote control) [1]. This system is made up of both hardware and software. The voice signal received by voice recognition module would be compared and matched with the pre-processed keywords stored in the microcontroller [2]. Lastly, identified signal will be execute through the I/O port.

II. MOTIVATION

This topic designs to implement a smart voice recognition control system. This implementation could help to improve the safety awareness of particular people as users are able to monitor and controlling the movement of wheelchair by themselves.

III. METHODOLOGY

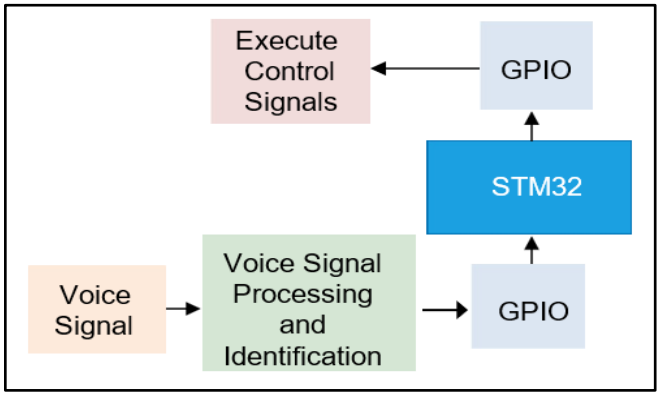

Fig. 1. Design block diagram of the voice recognition wheelchair control system.

The flow chart shown in Fig 1. is hardware design of the system. STM32 is used as main control unit [3]. The voice signal received by the microphone INMP441 and after that received signal is being compared with the pre-trained keyword. Execution is done on the identified signal to the output to blink the respective LED with specified pattern.

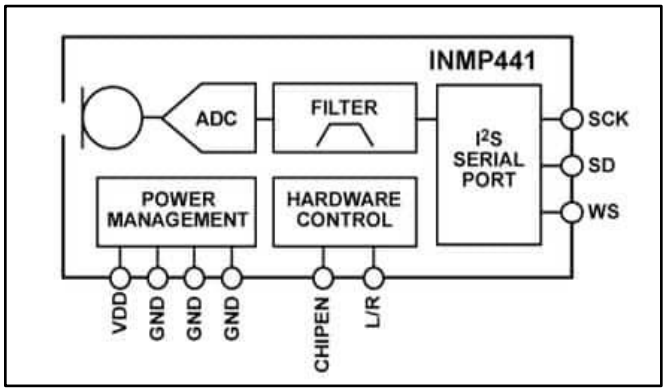

Fig. 2 Diagram of INMP441 Microphone.

a) INMP441 OMNIDIRECTIONAL MICROPHONE

Microphone INMP441 is chosen because it is a low power omnidirectional microphone which able to receive signals from all direction [4]. Besides, it is a digital microphone with a built-in ADC (analog to digital converter) and filter to attenuate noise signals.

Figure 2 shows the diagram of INMP441 microphone. The important feature of digital output through 24-bit I2S interface ease the software integration process as no analog to digital conversion is needed.

b) HARDWARE IMPLEMENTATION

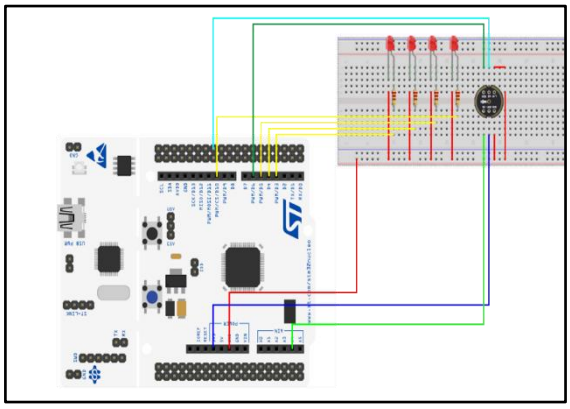

Fig. 3: Connection of STM32 microcontroller with INMP441.

Fig. 3 shows implementation of STM32 microcontroller with INMP441 microphone. WS (word select) and SD (serial data output) of the microphone is connected to the I2S peripheral. On the other hand, L/R channel of the microcontroller is connected to the ground to indicate left mono-output I2S is used. VDD is connected to 3.3V as the microphone is a low power microphone. This connection is implying in Table 1.

Pin on microphone	Integration of microphone and MCU
GND	GND
VDD	3V3
L/R	GND
WS	PB12
SCK	PB10
SD	PC1

Table. 1: Pin connection of the microphone with MCU.

c) MODEL TRAINING

Model training is the most important part for embedded system. Data is obtained from the Google speech datasets. In this project, we do not use custom data samples as the available dataset on Google contains over 1500 samples which are sufficient for our test case. The collected data with minimized background noise are then extracted by the Mel Frequency Cepstral Coefficient (MFCC) to obtain frequency domain signal. The signals are then feed into the Convolutional Neural Network (CNN) for keyword spotting training.

d) BOARD CONFIGURATION

MCU board configuration is needed for software integration. The board configuration includes the GPIO, DMA, USART and I2C setting. This is clearly shown in Fig. 4.

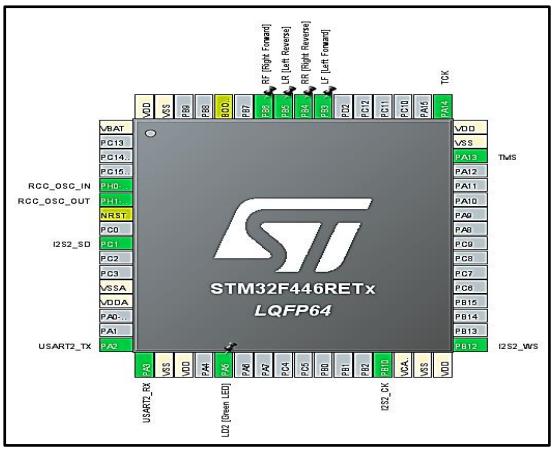

Fig. 4: STM32 Board Configuration.

e) FEATURE EXTRACTION

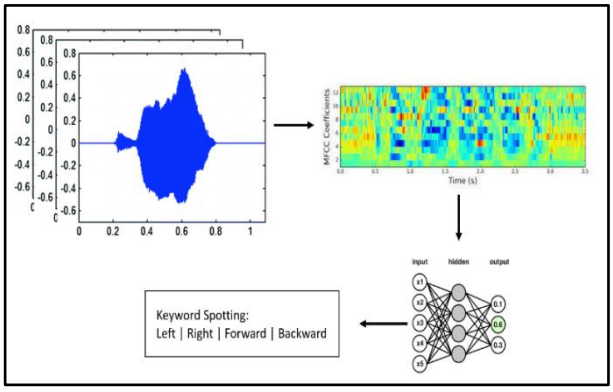

Fig.5: Flow Chart of the Training Process.

MFCC feature extractor is used for sound processing. First, the analog time varying input signal is split into short time frames. The purpose of framing is to simplify the sound signal. Framing helps to break down the audio signal into several portion with approximate voltage by assuming that on short time scales, the signal does not vary much. After framing, windowing is done for each individual frame to minimize the discontinuity and the spectral distortion of the signal. Fast Fourier Transform (FFT) of each frame is carried out to transform the time varying signal into frequency domain to ease audio interpretation and the power spectrum is computed.

Moreover, filtering is applied to the power spectrum to filter the noise from the signal to produce a cleaner spectrogram and Mel frequency transformation is carried out. This is a very important

process which would mimic human hearing perception. This process makes our features match more closely to what human actual hearing. The power of spectrum is thus mapped onto the Mel scale and take the logarithm of the powers at each Mel frequency.

Discrete Cosine Transformation of list of Mel log power is carried out to decorrelate the filter bank coefficient and yield a compressed representation of the filter banks. MFCC coefficient is the amplitude of the resulting spectrum.

f) CONVOLUTION NEURAL NETWORK

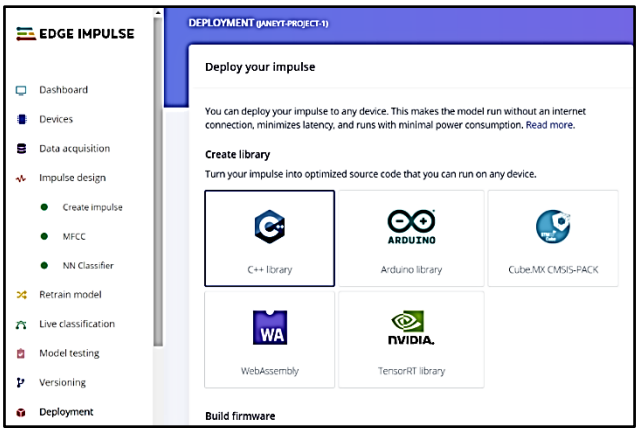

Fig. 6: Data Deployment from Edge Impulse.

Edge Impulse platform is used to train the model for keywords-spotting. It is an end-to-end tool that helps embedded engineers to test on machine learning applications by making use of the TensorFlow ecosystem for training, optimizing, and deploying deep learning models to the embedded devices. There are preset deep learning model architectures designed to work well with embedded devices. Edge Impulse generates a Python implementation of the model using TensorFlow's Keras APIs. Customization the layers of the deep learning network can be done by tweaking the parameters or adding new layers that are reflected in the underlying Keras model to edit the training code to suit own preference.

The audio samples with background noises and the specified keywords are feed into the Edge Impulse with MFCC feature extracting for Neural Network training for keywords recognition.

Besides, Edge Impulse also provide build-in optimization using TensorFlow's Model Optimization Toolkit to quantize models, reducing their weights' precision from float32 to int8 with minimal impact on accuracy.

Once a model has been trained, Edge Impulse provides a convenient way to deploy it to the target device. For our case, the trained model is deployed as C++ SDK, a library of optimized source code that implements both the signal processing pipeline and the deep learning model.

IV. RESULTS

The accuracy for keyword spotting is around 80.5%, as per shown in Fig. 7.

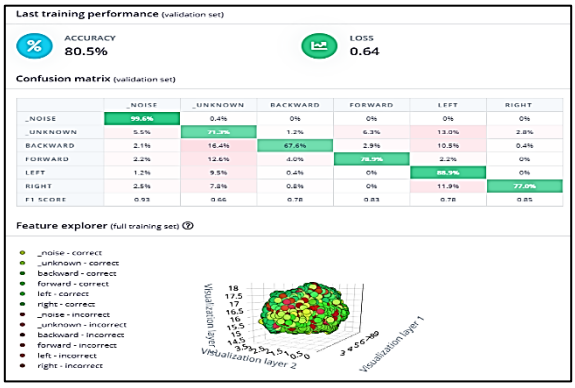

Fig.7: Training performance of keyword-spotting.

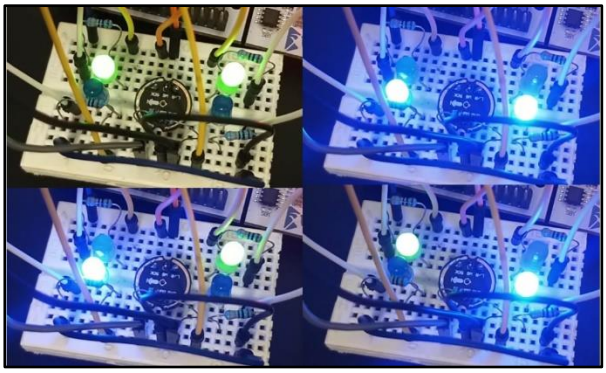

Fig.8: LEDs to implement Forward, Backward, Left, Right.

Fig. 8 shows the result of our demonstration. When the word "left" is spotted, then the blue LED at the left and green LED at the right turns on as for real case implementation, the left wheel will move backward while right wheel will move forward to perform left turn. When the word "right" is recognized, the green LED (forward) at the left and blue LED (backward) at the right will turn on. When the word "forward" and "backward" is recognized, both the LED at the front and back will light up respectively.

V. DISCUSSION

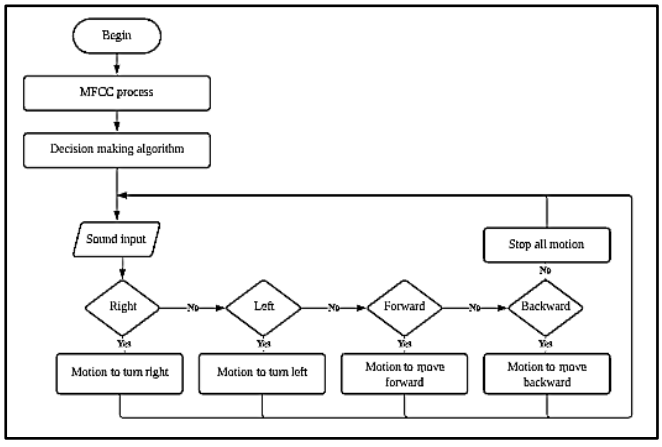

Fig.9: Flow chart of the project.

In this project, pre-processed word using MFCC would be stored inside the database of the microcontroller. Once INMP441 receives an instruction (sound input), it would send the received filtered command and thus triggered the microcontroller. STM32 would try to compare and match with its database. Once the command is successfully matched, either left, right, forward or backward, it would proceed to mimic the movement by blinking the LEDs. Last but not least, it would continuously receive instruction to perform next operation.

VI. CONCLUSION

Edge Impulse provides a user-friendly platform to implement Machine Learning for embedded system. The datasets or samples are extracted using MFCC and then being trained by Convolutional Neural Network for the keyword spotting system. The expected keywords "left", "right", "forward" and "backward" can be recognized from the training model with accuracy of up to 80%. However, the accuracy can be further improved by feeding more samples for training or using implemented sound module, INMP441 in this case to record the samples.

VII. REFERENCE

[1] Zhiyu Wang, Shanwei Wang, Shengyan Zeng. Design and implementation of Wireless Voice Control car based on STM32F103ZET6 [J]. Computer knowledge and Technology, 2018 (12): 197/99.

[2] Jing Jia. Design of embedded speech recognition Module based on STM32 [J]. Digital Technology and applications, 2016 (12): 197 / 99.

[3] Ruijie Tao, Zhengyu Zhang, Lei Chen. Design of Intelligent Voice car system based on XC7A35T and STM32F103RCT6 [J]. Electronic Design Engineering, 2018 (3): 170 / 174.

[4] Qixin Cao, Jianjun Du, Chuntao Leng, etc. Omnidirectional mobile multi-AGV system for cooperative handling [J]. Journal of Hua Zhong University of Science and Technology, 2013 (10): 241345.