디지털산업정보학회 논문지 제17권 제3호-2021년 9월
http://dx.doi.org/10.17662/ksdim.2021.17.3.009

Edge Impulse 기계 학습 기반의 임베디드 시스템 설계*

홍 선 학**

Edge Impulse Machine Learning for Embedded System Design

Seon Hack Hong

〈Abstract〉

In this paper, the Embedded MEMS system to the power apparatus used Edge Impulse machine learning tools and therefore an improved predictive system design is implemented. The proposed MEMS embedded system is developed based on nRF52840 system and the sensor with 3-Axis Digital Magnetometer, I2C interface and magnetic measurable range ±1200uT, BM1422AGMV which incorporates magneto impedance elements to detect magnetic field and the ARM M4 32-bit processor controller circuit in a small package. The MEMS embedded platform is consisted with Edge Impulse Machine Learning and system driver implementation between hardware and software drivers using SensorQ which is special queue including user application temporary sensor data. In this paper by experimenting, TensorFlow machine learning training output is applied to the power apparatus for analyzing the status such as "Normal, Warning, Hazard" and predicting the performance at level of 99.6% accuracy and 0.01 loss.

Key Words : Edge Impulse, Bluetooth5.0. MEMS System, Embedded System, IoT

Ⅰ. Introduction

In this paper, Embedded MEMS system design is implemented by using the Edge Impulse machine learning technology for predicting the performance and analyzing the status of embedded system on the ower apparatus in the advanced information society in oder to guarantee the stable performance in the industry fields and daily lives. Embedded MEMS system has the advantage of the low-power BLE and small size features.

In this paper, we apply the Bluetooth low power technology which is the Internet of Things(IoT) protocol for low data rate, low power and long range sensor applications in a variety of global markets.

We provide a system driver implementation of the MEMS based MCU, wireless devices and sensor devices. EdgeImpulse learning algorithm based on

* 본 연구는 2021년 서일대학교 교내학술연구비 지원으로 수행되었음.
** 서일대학교 컴퓨터전자공학과 교수(교신저자)

the TensorFlow machine learning is applied to the Embedded MEMS system for enhancing the predict performance[1-3].

Ⅱ. Edge Impulse Architecture

2.1 TensorFlow Lite

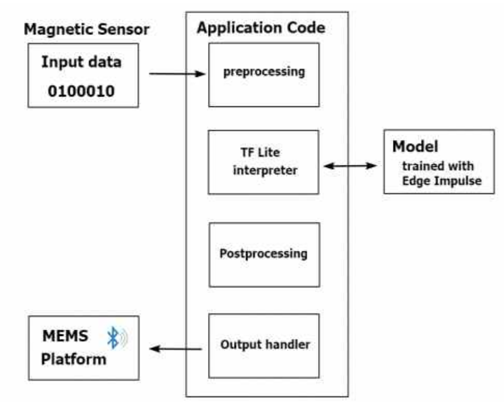
TensorFlow is Google's open source machine learning library, with the motto "An Open Source Machine Learning Framework for Everyone", It was developed internally at Google and first released to the public in 2015. After that it's aimed at Linux, Windows and macOS desktop and server platform and offers a lot of tools and optimizations around training and deploying models in the microcontrollers. In this paper we used Keras, TensorFlow's high-level API that makes it easy to build and train deep learning networks.

The biggest need when TensorFlow was launched was the ability to train models and run them in desktop environment. To meet memory lower size requirements for microcontrollers and mobile platforms, in 2017 Google started a companion project to mainline TensorFlow called TensorFlow Lite.

Edge Impulse project is based on the open source platform in the various fields for developing the TensorFlow study and enhancing the performance.

<Figure 1> shows the basic TensorFlow application architecture.

This library is aimed at running neural network models efficiently and easily on microcontrollers. To reduce the size and complexity of the framework, it drops features that are less common on these platforms.



<Figure 1> shows the basic TensorFlow application architecture.

TensorFlow Lite could fit within just a few hundred kilobytes, making it much easier to fit into a size-constrained application such as MEMS system. It also has highly optimized libraries for Arm Cortex series CPUs. Another key advantage is that it has good support for 8-bit quantization of networks.

Because a model might have millions of parameters, the 75% size reduction from 32 bit float to 8 bit integers alone makes it worthwhile, but there are also specialized code paths that allow inference to run much faster on the smaller data type[4-6].

2.2 TensorFlow Lite for Microcontroller

For memory size environments even a few hundred kilobytes was too large, the biggest constraint was binary size and they needed something that would fit within 20 KB or less.

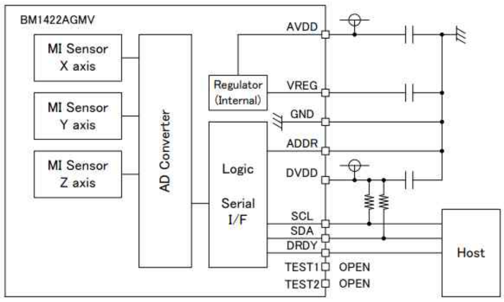
A lot of the dependencies that mobile developers takes for granted, like the C Standard Library, weren’t present either, so no code that relied on these libraries could be used. Inference was the primary use case, quantized networks were important for performance, and having a code base that was simple enough for developers to explore and modify was a priority.

Embedded environments imposed a lot of constraints on how the code could be written, so it identified some key requirements for the library as below, At first, operating system is need for independencies, no standard C or C++ library dependencies at linker time, floating-point hardware is not expected, and then dynamic memory allocation is not needed. Before TensorFlow Lite can run a model, it first must be converted into the TensorFlow Lite Format and then saved to disk as a file.

In this paper we used the Edge Impulse tool based on TensorFlow Lite which could also apply special optimization aimed at reducing the size of the model and helping it run faster, often without sacrificing performances.

2.3 Hardware System Design

In this paper, we used an BM1422AGMV is a 3-axis magnetic sensor which incorporates magneto impedance (MI) elements to detect magnetic field and a control IC in a small package. It has an I2C interface and 12 -bit / 14-bit digital output and is applied for wristwatch, smartphone. Input voltage range (AVDD/DVDD) is 1.7V to 3.6V, operating current is 0.15mA, and magnetic measurable range is ±1200μT. <Figure 2> shows the BM1422AGMV structure[5].

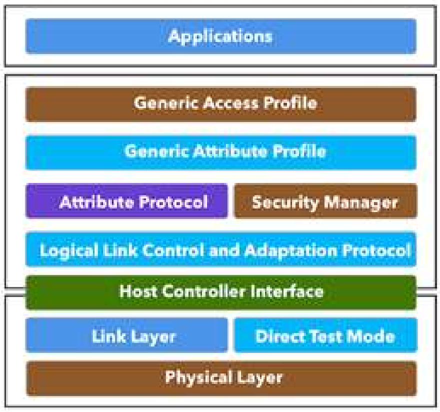


<Fig 2> BM1422AGMV structure

In this paper, the main micro-controller with MEMS system is ARM Cortex-M4 processor with floating point unit(FPU) has a 32-bit instruction set that implements superset of 16 and 32 bit instructions to maximize code density and performance. This processor implements several features that enable energy-efficient arithmetic and high-performance signal processing. The ARM cortex Microcontroller Software Interface Standard(CMSIS) hardware abstraction layer for the ARM Coretex processor series is implemented ans available for the M4 CPU.

2.4 Architecture of BLE

In this paper, we used the BLE technology for transferring the event data to communicate between devices. <Figure 3> shows the different layers within the BLE architecture. The three main components in the block are Application, Host and Controller. The application layer is user-case dependent and refers to the implementation on top of the Generic Access Profile and Generic Attribute Profile. It is how application handles data received from and sent to other devices and the logic behind it.

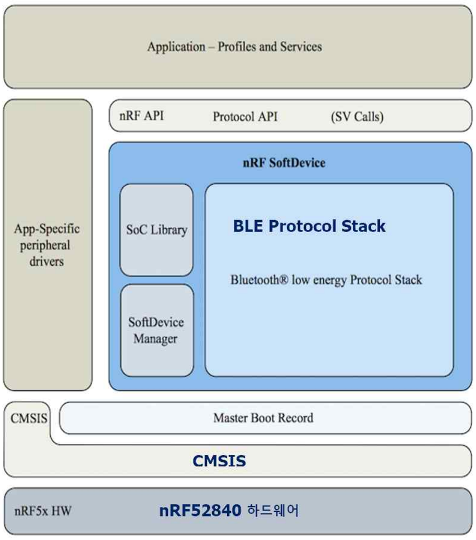


<Fig 3> BLE Architecture

Especially SoftDevice as like in <Figure 4>, is a wireless protocol stack library for building system on chip solutions. The SoftDevices are precompiled into a binary image and functionality verified according to the wireless protocol specification, so that all we have to think about is creating the application.

The unique hardware and software framework provide runtime memory protection, thread safety, and deterministic real-time behavior.

The application Programming Interface (API) is declared in the header files for the C programming language. These characteristics make the interface similar to a hardware driver abstraction where device services are provided to the application[7, 8].



<Fig 4> SoftDevice Architecture

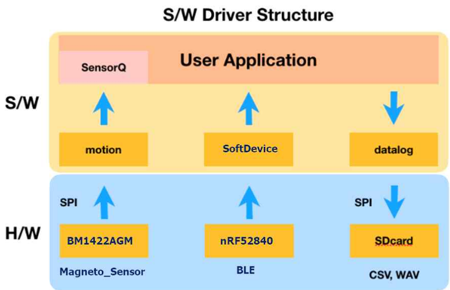
Ⅲ. System Software Architecture

We experimented the MEMS system with hardware architecture including magneto sensor for detecting the event of power system. Therefore, we designed the embedded platform for sensing the variation and monitoring its changing status and for transferring the information of its states with BLE technology.

3.1 Driver Architecture

In some case of any events from sensor would happen, we used the special queue structure to store the data of the time before and after events occurred. The queue saved the upcoming data from magneto motion driver, therefore we used push member function for collecting magneto driver, After the events happened, the pop member function worked for the saved data taking out from the queue.

Software driver structure as shown in <Figure 5> represented the relationship between hardware and software drivers. SensorQ, which is called special queue included in user application temporary stored data of sensor[9, 10].



<Fig 5> Driver Architecture

3.2 TensorFlow Model Setup

In this paper we used that the requirement for building Embedded MEMS system application is programming and mathematical expertise. Discerning a system's state based on complex patterns in multiple streams of data might require knowledge of some advanced techniques, like statistical analysis and signal processing.

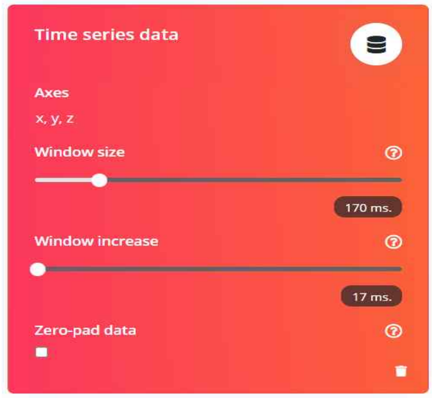
To build this, we might need to know how to mathematically filter the accelerometer data to get an estimate of vibration frequency. Instead of having to design a heuristic algorithm from scratch, a machine learning developers occasionally could find a suitable model architecture, collect and label a dataset, and iteratively create a model through training and evaluation. In fact, a recent paper showed how a simple convolutional neural network is able to detect congestive heart failure in a pattern from a single heartbeat with 100% accuracy.

In this paper we used the 3-axis magneto (BM1422AGMV) which produced three values representing the amount of magnetometer on the devices x, y, and z-axes. The magnetic value on the Embedded MENS system could do this 25 times per second (a rate of 25 Hz). Our model took these values directly as its input, meaning we won't need to do any preprocessing. After data has been captured and inference has been run, our application would determine whether a valid vibration detected, send some output to the terminal, and light the LED[11,12].

Ⅳ. Experiment Result

In this paper, we used the Edge Impulse which took raw data, used signal processing to extract features, and then used a learning block to classify new data. There are three steps in which the first step is the time series data with window size of 170ms and window increase of 17 ms, the second is the spectral analysis, and the last step is neural network of Keras with NN Classifier. <Figure 6> shows the Time series data in Create impulse.

Especially during NN Classifier analysis, we used number of training cycles 500, Learning rate 0.001, minimum confidence rating 0.60. and in training output process, we got the training performance 99.6% and loss 0.01. as like <Figure 7> At this stage, we took 1ms in processing time and 392Bytes in peak RAM usage on-device performance.



<Fig 6> Time Series Data

Last training performance (validation set)

ACCURACY 99.6%

LOSS 0.01

Confusion matrix (validation set)

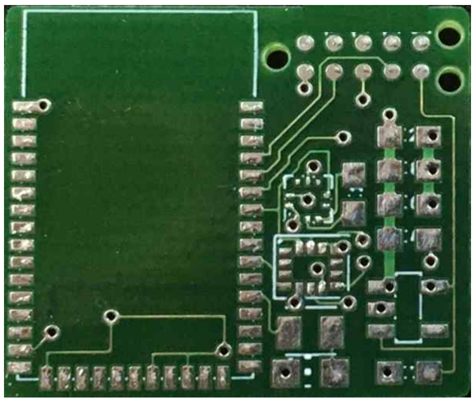
	HAZARD	NORMAL	WARNING
HAZARD	100%	0%	0%
NORMAL	0%	100%	0%
WARNING	2.4%	0%	97.6%
F1 SCORE	0.99	1.00	0.99

<Fig 7> NN Classifier Result

Similarly, a CNN with multiple layers is able to learn how to discern each event through its telltale component parts. To do this, a CNN learns a series of filters, arranged in layers. each filter learns to spot a particular type of features in the data. When it notices this feature, it passes this high-level information to the next layer of the network. For example, one filter in the first layer of the network might learn to spot something simple, like a period of next operation. When it identifies such a structure, it passes this information to the next layer of the network [13-16].

Ⅴ. Conclusion

In this paper, we implemented the Edge Impulse Embedded MEMS system for integrating the magnetometer based on the TensorFlow machine learning training output in order to predict the events on the power system. Especially we experimentally designed the integrated Embedded MEM system including the wireless BLE, MCU and sensors. In this paper, we designed a MEMS system with 22*25mm level of hardware design including MPU, sensors and machine learning software driver design technology as like <Figure 8>.



<Fig 8> MEMS System Layout

Afterwards, we would enhance the performance of system platform with adapting the more flexible technologies.

참고문헌

[1] Sun Hag Hong, Seong Jun Lee, Hyo Jun Park, “COS MEMS system design with embedded

technology," KEPCO Journal, 12, 2020, pp.405-411.
[2] TensorFlow.org, https://www.Tensor flow.org, 10. March. 2021
[3] EDGE IMPULSE, docs.edgeimpulse.com, API and SDK References, March 2021.
[4] Arduino NANO 33 BLE, NANO 33 BLE. 2019
[5] Rohm.com. ROHM Semiconductor, 3-Axis Digital Magnetometer IC, Datasheet. March 2021.
[6] Sun Hag Hong, Seong Jun Lee, Yeon Sub Lee, Taeil Choi, "Power Supply design of COS MEMS system," ICEIC December 2021.
[7] STMicroelectronics, "Cortex M0+ programming manual for STM32M0, STM32M0 Series," October 2019.
[8] Sun Hag Hong ,"Embedded system Design with COS LoRa technology," Korea Society of Digital Industry and Information Management. 3rd, 14. 2018, pp.29-38
[9] Naveen Suda, Danny Loh, Machine Learning on Arm Cortex-M Microcontrollers, ARM, 2019
[10] MUHAMMAD ALI MAZIDI, "STM32 ARM PROGRAMMING FOR EMBEDDED SYSTEMS," MicroDigitalEd, 2020
[11] Pete Warden and Daniel Situnayake, TinyML, Oreilly, 2020
[12] Sun Hag Hong, Seong Jun Lee, Hyo Jun Park, "Implementation of Improved Power Protection Apparatus using Embedded System," IEIE Summer Conference, 2020.
[13] Hong Seon Hack, Cho Kyung Soon, "Computer Vision Platform Design with MEAN Stack," Korea Society of Digital Industry and Information Management. 09. 2015. pp.79-87.
[14] Hong Seon Hack, "Mobile Arduino Embedded Platform Design," Korea Society of Digital Industry and Information Management. 4th, 12. 2013, pp.33-41.
[15] Hong Seon Hack, "Real Time Linux System Design," Korea Society of Digital Industry and Information Management. 10th, 06. 2014, pp.13-20.
[16] Hong Seon Hack, Cho Kyung Soon. "3D Scanning Embedded System Design," Korea Society of Digital Industry and Information Management. 13th, 12. 2017, pp.49-56.

■ 저자소개 ■

홍 선 학
(Hong, Seon Hack)

1992년~현재
서일대학교 컴퓨터전자공학과 교수
2005년 뉴질랜드 와이아리키대학 교환교수
2008~2009년
유네스코 라오스/인도네시아 대학교 코딩 교육

관심분야 : 제어 / 임베디드 / 리눅스 설계
E-Mail : hongsh@seoil.ac.kr

논문접수일 : 2021년 7월 27일
수 정 일(1차) : 2021년 8월 10일
수 정 일(2차) : 2021년 8월 20일
수 정 일(3차) : 2021년 8월 27일
게재확정일 : 2021년 9월 10일