

國立交通大學

電機與控制工程學系

碩士論文

以獨立成份分析應用於瞌睡偵測之  
即時嵌入式無線腦機介面系統

**ICA-Based Embedded Wireless BCI System for  
Real-Time Drowsiness Detection**

研究生：林真如

指導教授：林進燈 博士

中華民國九十六年八月

以獨立成份分析應用於瞌睡偵測之即時嵌入式

無線腦機介面系統

**ICA-Based Embedded Wireless BCI System for  
Real-Time Drowsiness Detection**

研 究 生：林真如

Student：Jhen-Ru Lin

指 導 教 授：林進燈 博士

Advisor：Dr. Chin-Teng Lin

國立交通大學



A Thesis

Submitted to Department of Electrical and Control Engineering  
College of Electrical and Computer Engineering  
National Chiao Tung University  
in Partial Fulfillment of the Requirements  
for the Degree of Master  
in  
Electrical and Control Engineering  
June 2007  
Hsinchu, Taiwan, Republic of China

中華民國 九十六 年 八 月

# ICA-Based Embedded Wireless BCI System for Real-Time Drowsiness Detection

Student: Jhen-Ru Lin

Advisor: Dr. Chin-Teng Lin

Department of Electrical and Control Engineering  
National Chiao Tung University

## Abstract

Many traffic accidents on the highway are caused by the driver's inattention due to drowsiness. Hence, many researchers have devoted to develop algorithms to prevent drowsiness. Recently, several studies have shown that drowsiness related information is available in eye closures, heart rate and electroencephalogram (EEG). In this study, we developed a drowsiness detection system based on EEG recordings and data analysis. One of the biggest challenges in EEG-based system lies on the contaminations from inevitable EEG artifacts from eye movements, blinks, muscle, heart, and line noise. Independent component analysis (ICA) has been proven to be an effective technique to remove various types of artifacts. However, most of the ICA was performed offline on personal computers instead of an online analysis. Here, we design, develop and demonstrate an embedded wireless brain computer interface (BCI) including three functional blocks: EEG acquisition, amplification and wireless transmission, on-line ICA process and spectral analysis, and real-time drowsiness detection and feedback delivery to accurately and continuously detect and report subject drowsiness level based on the EEG data.

KEYWORD: Brain Computer Interface, Brain Signal Processing, Digital Signal Processor, Drowsiness Detection and Warning, Embedded System, Independent Component Analysis, Wireless transmission

# 以獨立成份分析應用於瞌睡偵測之即時嵌入式

## 無線腦機介面系統

學生：林真如

指導教授：林進燈 博士

國立交通大學電機與控制工程研究所

### 中文摘要

近年來，有許多高速公路車禍發生的原因是由於駕駛者精神狀態不佳所造成。因此許多學者都在針對疲勞狀態偵測開發演算法。近年來，許多研究發現疲勞的特徵可以從眨眼的頻率、心跳頻率與腦波中萃取出來。這本研究中，我們開發一套基於腦波的量測與分析的疲勞偵測系統。而以腦波量測與分析的為主的系統最大的挑戰，就是腦波中的雜訊。腦波的量測與分析中不可避免的雜訊是眼動、眨眼、肌肉的雜訊、心跳的雜訊以及室電的干擾。而有研究證明獨立成分分析可以有效的移除多種的雜訊。然而大多數的獨立成份分析都是在電腦上進行離線分析而不是即時線上分析。

本論文以數位訊號處理器為基礎，搭配無線傳輸模組、生理訊號放大器，來實現一套即時可攜式無線腦機介面系統，包含三大發展主軸，分別為「腦電位訊號量測與無線傳輸」、「線上獨立成份分析演算法」及「即時疲勞狀態偵測與提醒演算法」。最後這套系統展示了連續並準確地偵測受測者的腦波中的精神狀態。

關鍵字：腦機介面，腦波訊號處理，數位訊號處理器，疲勞狀態偵測與提醒，嵌入式系統，獨立成份分析，無線傳輸

## 誌 謝

本論文的完成，首先要感謝指導教授林進燈博士，讓我學習到許多寶貴的知識，在學業及研究方法上也受益良多。另外也要感謝口試委員：UCSD 的鍾子平博士、中央大學的蘇木春博士、交通大學的范倫達博士以及新紐光電的鍾仁峰博士的建議與指教，使得本論文更為完整。

其次，感謝交通大學腦中心的曲在雯博士，給予我相當多的幫助與建議，讓我獲益良多。此外，也衷心的感謝學長弘義、紹航、力碩、立偉、騰毅、世安，學姐玉潔、君玲，同學弘章、俊傑、明達、智文、德瑋及靜瑩、仲良、吉隆的相互砥礪，以及學弟妹依伶、家欣，學弟德正、青甫、孟修、柏銓、玠瑤、尚文、有德、哲睿、介恩、盈宏、Julien 在研究過程中所給我的鼓勵與協助。

感謝我的父母親林錦星先生與張麗華女士對我的教育與栽培，並給予我精神及物質上的一切支援，使我能安心地致力於學業。此外也感謝姊姊林琦對我不斷的關心與鼓勵。

謹以本論文獻給我的家人及所有關心我的師長與朋友們。

# Table of Contents

Abstract.....	ii
中文摘要.....	iii
致謝.....	iv
Table.....	vii
Figure .....	viii
Chapter 1 Introduction.....	1
1.1 Motivation and Problems .....	1
1.2 Organization of Thesis.....	2
1.3 Notation.....	3
Chapter 2 Background and Previous Works .....	6
2.1 Brain Computer Interface.....	6
2.2 Independent Component Analysis.....	8
2.3 Drowsiness Detection Methods.....	9
Chapter 3 Embedded Brain Computer Interface Architecture .....	11
3.1 System Overview .....	11
3.2 System Function Block .....	12
3.3 Requirement Analysis of Hardware and Software .....	14
3.3.1 EEG Acquisition Circuit .....	14
3.3.2 Wireless Transmitting Module.....	15
3.3.3 Embedded Digital Signal Processor .....	18
3.4 Development of the System .....	20
3.4.1 Boot Loader .....	20
3.4.2 Operating System.....	24
3.4.3 User Program .....	29
3.5 Optimization of the System .....	30
3.5.1 Floating Point to Integer .....	30
3.5.2 Programming.....	31
3.5.3 Variable Regulation .....	31
3.6 Comparison to Past BCI Systems.....	32
Chapter 4 On-line Independent Component Analysis Implementation .....	34
4.1 Independent Component Analysis.....	35

4.2 Rejecting Components with Standard Deviation .....	41
4.3 Testing .....	43
4.3.1 Testing of Artificial Mixed Data .....	43
4.3.2 Testing of Real EEG Data.....	45
<b>Chapter 5 Experiment Designs and Results .....</b>	<b>48</b>
<b>5.1 The Experiment Design .....</b>	<b>48</b>
5.1.1 Experimental Setup .....	48
<b>5.2 Drowsiness Detection Algorithm.....</b>	<b>50</b>
<b>5.3 The Results .....</b>	<b>56</b>
5.3.1 Result of 4 Channel Online EEG Analysis System.....	56
<b>Chapter 6 Conclusions and Future Works.....</b>	<b>59</b>
<b>References .....</b>	<b>60</b>



## Table

Table 3-1 Wireless package definition .....	16
Table 3- 2 RF3105/RF3100 module.....	17
Table 3- 3 Comparison to Other Operating Systems .....	28
Table 3- 4 Comparison top past BCI systems .....	32
Table 4- 1 The result of each ICA with different variables.....	45
Table 5- 1 Characteristics of EEG bands .....	50
Table 5- 2 Criterion of Drowsiness .....	52





## Figure

Fig.2- 1 Learning to Control a BMic for Reaching and Grasping by Primates.....	8
Fig.3- 1 System architecture.....	11
Fig.3-2 Data flow of the system.....	12
Fig.3-3 Block diagram of the signal acquisition and amplification unit.....	13
Fig.3-4 Block diagram of the wireless data transmission unit.....	13
Fig.3-5 The signal flow of EEG signal recording circuit.....	14
Fig.3- 6 The picture of EEG signal recording circuit .....	15
Fig.3-7 The data flow from A/D converter to wireless receiver .....	15
Fig.3-8 The picture of the integrated signal recording circuit .....	16
Fig.3-9 RF3105/RF3100 module .....	17
Fig.3- 10 flowchart of receiving procedure .....	18
Fig.3-11 The system diagram of the board .....	19
Fig.3-12 Picture of the board (upside and downside) .....	19
Fig.3-13 U-Boot boot up flow .....	20
Fig.3- 14 Setting the UART port.....	22
Fig.3-15 Boot up message.....	22
Fig. 3-16 Map of U-boot in external memory.....	23
Fig.3- 17 Map of U-boot in external memory after loading U-boot and resetting .....	24
Fig.3- 18 Map of kernel in external memory after loading kernel and resetting .....	26
Fig.3- 19 Map of kernel and JFFS2 .....	27
Fig.3- 20 Map of rest external memory .....	28
Fig.3- 21 Cross-platform development framework .....	29
Fig.3- 22 Flow to create an executable image .....	30
Fig.4- 1 Flowchart of the EEG signal procedure .....	34
Fig.4- 2 N mixed signals and N sources .....	35
Fig.4- 3 The typical super Gaussian variables is spiky. For comparison, the normal Gaussian is given by a dashed line. The other is sub Gaussian one. ....	37
Fig.4- 4 The brain activity recorded at one point.....	40
Fig.4- 5 Flow of ICA training .....	41
Fig.4- 6 The observed 4-channel mixtures signals .....	42
Fig.4- 7 The result of informax ICA, estimated by the signals in Fig.4 -6.....	42
Fig.4- 8 The observed 5-channel mixtures signals .....	43
Fig.4- 9The result of informax ICA, estimated by the signals in Fig.4-8.....	43
Fig.4- 10The original signals .....	44
Fig.4- 11The observed mixtures of the source signals in Fig.4-10.....	44

Fig.4- 12 The result of fast ICA, estimated using only the observed signals in Fig.4-11 .....	45
Fig.4- 13 The result of Information Maximization ICA, estimated using only the observed signals in Fig.4-11 .....	45
Fig.5- 1 A 360-degree 3D VR dynamic driving environment.....	49
Fig.5- 2 The VR-based four-lane highway scene .....	50
Fig.5- 3 The Response Time in One Session.....	50
Fig.5- 4 The original signal of 4Hz & 5Hz sinusoid.....	53
Fig.5- 5 The power spectrum of 4Hz & 5Hz sinusoid.....	53
Fig.5- 6 The original signal of 10Hz & 12Hz sinusoid.....	54
Fig.5- 7 The power spectrum of 10Hz & 12Hz sinusoid.....	54
Fig.5- 8 The flowchart of our algorithm .....	55
Fig.5- 9 The procedure of FFT.....	55
Fig.5- 10 The flow of drowsiness estimation algorithm.....	56
Fig.5- 11A Demonstration of Our System .....	57
Fig.5- 12 A Demonstration of EEG Hat.....	57
Fig.5- 13 The Result of GUI.....	57



# Chapter 1 Introduction

## 1.1 Motivation and Problems

Driver's fatigue is a causal factor in many accidents because of the marked decline in the driver's abilities of perception, recognition, and vehicle control abilities while sleepy. Hence, the development of the drowsiness monitoring technology to prevent accidents behind the steering wheel has received increasing interest in the field of automotive safety. Lately, several studies have shown that drowsiness related information is available in eye closures, heart rate and electroencephalogram (EEG). Previous studies showed that the eye blink duration and the blink rate typically increase while blink amplitude decreases as a function of the cumulative time, and the saccade frequencies and velocities of electrooculogram (EOG) decline when people get drowsy [25]. Although these studies showed that eye-activity variations were highly correlated with the human fatigue and could accurately and quantitatively estimate alertness levels, the step size (temporal resolution) of these eye-activity based methods was too slow (10s or longer) to track momentary changes in vigilance. Contrarily, the temporal resolution of EEG-based methods could reach 1-2 sec that makes them faster enough to track second-to-second fluctuations in the subject's performance. Although EEG signals have been proved to index the cognitive states of a person, signal analysis is very challenge in EEG-based systems because of the pervasive contaminations from eye movements, blinks, muscle, heart, and line noise to the EEG. Independent component analysis (ICA) has been proved to be an effective technique to remove various types of artifacts [23][24]. However, most of the ICA in these studies was performed offline on personal computers instead of an online system. For eventual practical acceptance in the workplace, it is highly desirable to make all data acquisition and analysis on-lined. Here, we report our work in the

design and test of a wireless embedded brain computer interface (BCI) that comprises three functional modules: (1) EEG acquisition, amplification and wireless transmission; (2) on-line ICA process, and (3) real-time drowsiness detection to accurately and continuously detect subject drowsiness level based on the EEG data and feedback delivery.

## 1.2 Organization of Thesis

The organization of this thesis goes as follows:

- Chapter 2

In chapter 2 we briefly describe the history of BCI, ICA and other drowsiness detection methods. Three kinds of drowsiness detection methods will be described here.

- Chapter 3

In chapter 3 we briefly describe the basic architecture of BCI. Then we will propose a new embedded BCI system that can assess subject drowsiness levels in near real time. In this chapter we will address the following issues:

- Hardware specification: The hardware specification of the embedded BCI system will be described here. Hardware specification includes an EEG recording, amplification and wireless transmitting unit and an embedded digital signal processing (DSP) board.
- The operating system (OS): The OS for the embedded DSP board will be described here.
- The applications implemented on the embedded digital signal processor: The applications on the signal processor include an on-line ICA process, spectral estimation and a real-time drowsiness detection

algorithm. The techniques used to verify the application will be described here as well.

- Chapter 4

In chapter 4 we briefly describe the theory of ICA and the technique for improving the convergence of ICA algorithm.

- Chapter 5

In chapter 5 we describe the experimental design. The experiments are designed to simulate highway driving during which drivers often struggle to maintain their alertness and attention. The correspondence between a driver's behavioral and estimated performance obtained by the embedded DSP will be shown and discussed here.

- Chapter 6

We will conclude our work in Chapter 6.



### 1.3 Notation

<b>Abbreviation</b>	<b>Original text</b>	<b>Chinese Translation</b>
<b>A/D</b>	<b>Analog to Digital</b>	<b>類比數位訊轉換</b>
<b>API</b>	<b>Application Program Interface</b>	<b>應用程式介面</b>
<b>BCI</b>	<b>Brain Computer Interface</b>	<b>大腦人機介面</b>
<b>DMA</b>	<b>Direct Memory Access</b>	<b>直接記憶體存取</b>
<b>DSP</b>	<b>Digital Signal Processor</b>	<b>數位訊號處理器</b>
<b>ECG</b>	<b>Electrocardiogram</b>	<b>心電圖</b>

<b>EEG</b>	<b>Electroencephalogram</b>	<b>腦電波</b>
<b>EMG</b>	<b>Electromyogram</b>	<b>肌電圖</b>
<b>EOG</b>	<b>Electro-oculogram</b>	<b>眼電圖</b>
<b>FHSS</b>	<b>Frequency Hopping Spread Spectrum</b>	<b>跳頻展頻</b>
<b>FPGA</b>	<b>Field Programmable Array</b>	<b>可程式化邏輯閘陣列</b>
<b>FSK</b>	<b>Frequency Shift Keying</b>	<b>頻率鍵移</b>
<b>GCC</b>	<b>GNU Compiler Collection</b>	<b>GNU 編譯器</b>
<b>GSM</b>	<b>Global System for Mobile Communication</b>	<b>全球行動通訊系統</b>
<b>GPRS</b>	<b>General Packet Radio Service</b>	<b>通用分組無線服務</b>
<b>IDE</b>	<b>Integrated Development Environment</b>	<b>整合發展環境</b>
<b>IP</b>	<b>Internet Protocol</b>	<b>網際網路通訊協定</b>
<b>ICA</b>	<b>Independent Component Analysis</b>	<b>獨立成份分析</b>
<b>LCD</b>	<b>Liquid Crystal Display</b>	<b>液晶平面顯示器</b>
<b>MMU</b>	<b>Memory Management Unit</b>	<b>記憶體管理單元</b>
<b>OS</b>	<b>Operating System</b>	<b>作業系統</b>
<b>PCA</b>	<b>Principle component analysis</b>	<b>主成分分析</b>
<b>PDA</b>	<b>Personal Digital Assistant</b>	<b>個人數位助理</b>
<b>RF</b>	<b>Radio Frequency</b>	<b>射頻</b>

<b>SoC</b>	<b>System on Chip</b>	系統晶片
<b>STFFT</b>	<b>Short-Time FFT</b>	短時快速傅立葉轉換
<b>TCP</b>	<b>Transmission Control Protocol</b>	傳輸控制協定
<b>TCS</b>	<b>Telephone Control Service</b>	電話傳送控制協定
<b>UART</b>	<b>Universal Asynchronous Receiver Transmitter</b>	通用異步收發器
<b>USB</b>	<b>Universal Serial Bus</b>	通用串列匯流排



## Chapter 2 Background and Previous Works

### 2.1 Brain Computer Interface

Brain computer interface (BCI) is an interface between human and computer. It is based on the specific brain activity generated by a specific thought of a human. That is, we can obtain information from brain activity via signal processing and use the recognized pattern to control a computer. At the beginning, the purpose of BCI is not only prosthesis but also is to help handicapped people [48], gradually. Because of the disability of muscle, handicapped people can not do things independently. For example, handicapped people cannot move, control devices without aid. Hence, to help these handicapped people, many researchers have devoted themselves to develop BCI. That is, as long as handicapped people are still cognitively healthy, they might able to move on an automatic wheel chair, and control the on/off switches of lamps via EEG recording and analysis. Through decades, it have been found in many studies that the cognitive state of a person can be extracted from brain activity [1][2]. More and more researchers are devoted to the study of BCI. BCI has helped handicapped to live independent. Recent studies in primates, human subjects of Serruya et al. and Taylor et al. [3][4] have demonstrated that animals can learn to utilize their brain activity to control the displacements of computer cursors. Chapin et al. and Wessberg et al. also demonstrated that animals can learn to utilize their brain activity to control one- (1D) to three-dimensional (3D) movements of simple and elaborate robot arms [5][6]. However, many domestic researches were focusing on EEG data recording instead of EEG analysis [7]-[13]. Gao et al. have developed wireless BCI based on steady-state visual evoked potential (SSVEP) [14]. They used twelve buttons illuminated at different rates on a computer monitor to simulate a telephone. Users could input phone numbers by gazing at these buttons. The frequency-coded SSVEP



was used to judge which button the user attended to. Another study of Gao et al. used digital signal processor (DSP) to process EEG signals and wirelessly controlled appliances with visual evoked stimulus [15]. Pfurtscheller et al. have designed and implemented an EEG-based communication device called “Virtual Keyboard” (VK). Classification of the EEG patterns was based on band power estimates and hidden Markov models (HMMs) [16][17]. Another research of Pfurtscheller et al. proposed an EEG-based Pocket BCI system that converted brain activity into control signals left and right direction of a wheelchair [18]. Ashwin et al. described [19] a system that monitored EEG of epileptic patients to improve the quality of their lives and also helped healthcare providers to make a better diagnosis for patients with neurological disorders. The use of Bluetooth connectivity helps physicians to monitor patient activity while the patient resumes his or her normal activity.

Compared to the portability, many other applications using ECG are always based on embedded systems. The EEG-based application on embedded systems is rarely seen. Han-Nam et al. have developed a system which enabled a patient to be treated at home through digital telemetry and public communication line. ECG signals were transmitted wirelessly and not hindering a patient's movement [20]. Other ECG systems only record ECG signals and leave the signal analysis to doctors [21][22].

Traditionally, a BCI system can be divided into two functional blocks. One is data acquisition and recognition part, another is control part. An example of BCI is shown as Fig.2-1. Jose. et al. demonstrate that primates can learn to reach and grasp virtual objects by controlling a robot arm through a closed-loop brain-machine interface (BMiC) [30]. It uses multiple mathematical models to extract several motor parameters (i.e., hand position, velocity, gripping force, and the EMGs of multiple arm muscles) from the electrical activity of frontal parietal neuronal ensembles. Using visual feedback, monkeys succeeded in producing robot reach-and-grasp movements

even when their arms did not move.

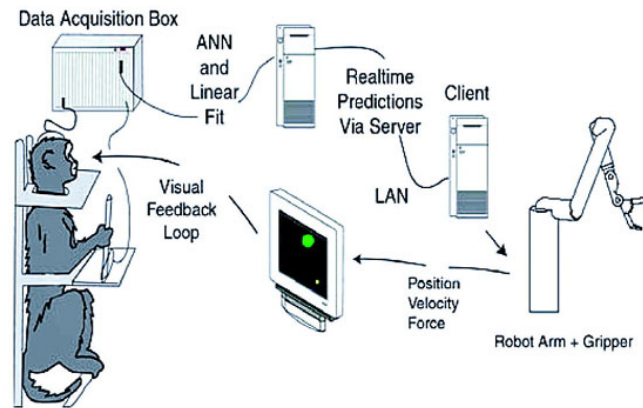


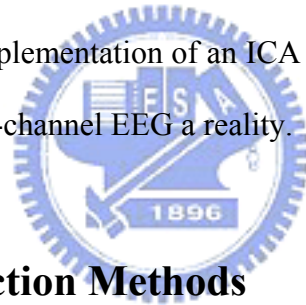
Fig.2- 1 Learning to Control a BMIc for Reaching and Grasping by Primates

## 2.2 Independent Component Analysis

The original ICA was used for voice signal separation. A classical application of ICA is the “cocktail party problem”, where a number of people are talking simultaneously in a room (like at a cocktail party), and one is trying to follow one of the discussions. The human brain can handle this sort of auditory source separation problem, but it is a very difficult problem in digital signal processing.

EEG is a non-invasive record of brain electrical activity measured as changes in potential difference between pairs of electrodes placed on the human scalp. Because of volume conduction through brain tissue, cerebrospinal fluid, skull, and scalp, EEG data collected anywhere on the scalp mixes signals from multiple suitably-oriented cortical areas. Thus, the analysis of EEG signals is also a “cocktail party problem” and often very challenging. One of the most pervasive problems in EEG analysis and interpretation is the interference in the data produced by often large and distracting artifacts arising from eye movements, eye blinks, muscle noise, heart signals, and line noise. One common strategy for avoiding EEG artifacts is to reject all EEG recordings containing artifacts larger than some arbitrarily selected EEG voltage value.

However, when limited data are available, or when blinks and muscle movements occur too frequently as in children and some patient groups, the amount of data lost to artifact rejection may be unacceptable. Use of principal component analysis (PCA) has been proposed to remove eye artifacts from multi-channel EEG. However, PCA cannot completely separate eye artifacts from brain signals, especially when they have comparable amplitudes. Jung et al. [23][24] use ICA for removing a wide variety of artifacts from EEG records. Their results on EEG data collected from normal and autistic subjects show that ICA can effectively detect, separate, and remove contamination from a wide variety of artificial sources in EEG records with results comparing favorably with those obtained using regression and PCA methods. Nevertheless, most ICA was done off-line on PCs, hindering portability and on-line BCI. This study reports an implementation of an ICA algorithm, making on-line blind source separation of the multi-channel EEG a reality.



## **2.3 Drowsiness Detection Methods**

With the increasing casualties on highway due to drowsiness, many researchers have devoted to develop algorithms to prevent drowsiness. One method is to detect drowsiness based on changes in blink behavior. The drowsiness detection Ulrika proposed was based on changes in blink behavior and classification was made on a four graded scale using MATLAB [25]. Although several studies showed that eye-activity variations were highly correlated with the human fatigue and can accurately and quantitatively estimate alertness levels, the step size (temporal resolution) of those eye-activity based methods is often too slow (10s or longer) to track fast changes in vigilance. Another method is to detect drowsiness information by heart rate monitoring sensors placed in the steering wheel [26]. But the drowsiness

classification needs coordination, its not autonomy parameter.. Matsushita et al. have developed a wearable fatigue monitoring system with a 2-axis accelerometer and an on-board signal processing microcontroller [27]. As a result, the measured values of the acceleration trace length showed some inconsistency with user-interviews consist of subjective questionnaires about the user's fatigue [27].

It has also been known for more than half a century that signal changes related to alertness, arousal, sleep, and cognition are present in EEG signals, but relatively little has been done to capture this information in real time [25]. Recently, Lin et al. have developed a drowsiness-estimation system based on EEG by combining ICA, power-spectrum analysis, correlation evaluations, and linear regression model to estimate a driver's cognitive state when he/she drove a car in a virtual reality (VR)-based dynamic simulator [28][29]. The proposed ICA-based method applied to power spectrum of ICA components successfully removed most of EEG artifacts and estimated the driver's drowsiness fluctuation indexed by the driving performance measure. However, their drowsiness detection was still performed off-line on a personal computer. For eventual practical acceptance in the workplace, it is highly desirable to make all data acquisition and analysis on-lined. In this study, we design, develop and demonstrate an embedded wireless BCI that comprises three functional modules: EEG recordings, amplification, digitization and wireless transmission, on-line ICA process and spectral estimation, and real-time drowsiness detection algorithm to accurately and continuously detect subject drowsiness level based on the EEG data in near real time.

# Chapter 3 Embedded Brain Computer Interface Architecture

## 3.1 System Overview

Fig.3-1 shows the three major components of the developed embedded wireless BCI system: EEG amplifying and transmitting unit, EEG receiving and processing unit, and display unit. The amplifying and transmitting unit, which is also called EEG signal pre-processing unit, is capable of handling 4 channels of EEG data. In the beginning, the subject will be attached to six electrodes: five on forehead (one ground and four EEG channels), one on mastoid bone after ear. Then the EEG data is amplified by 5000 times, converted to digital signals, and then transfer via radio frequency (RF) module. All the process on the amplifying and transmitting unit is controlled by a Field-Programmable Gate Array (FPGA). After receiving EEG data from the amplifying and transmitting unit, the EEG data is processed by receiving and processing unit then the result is sent to the display unit.

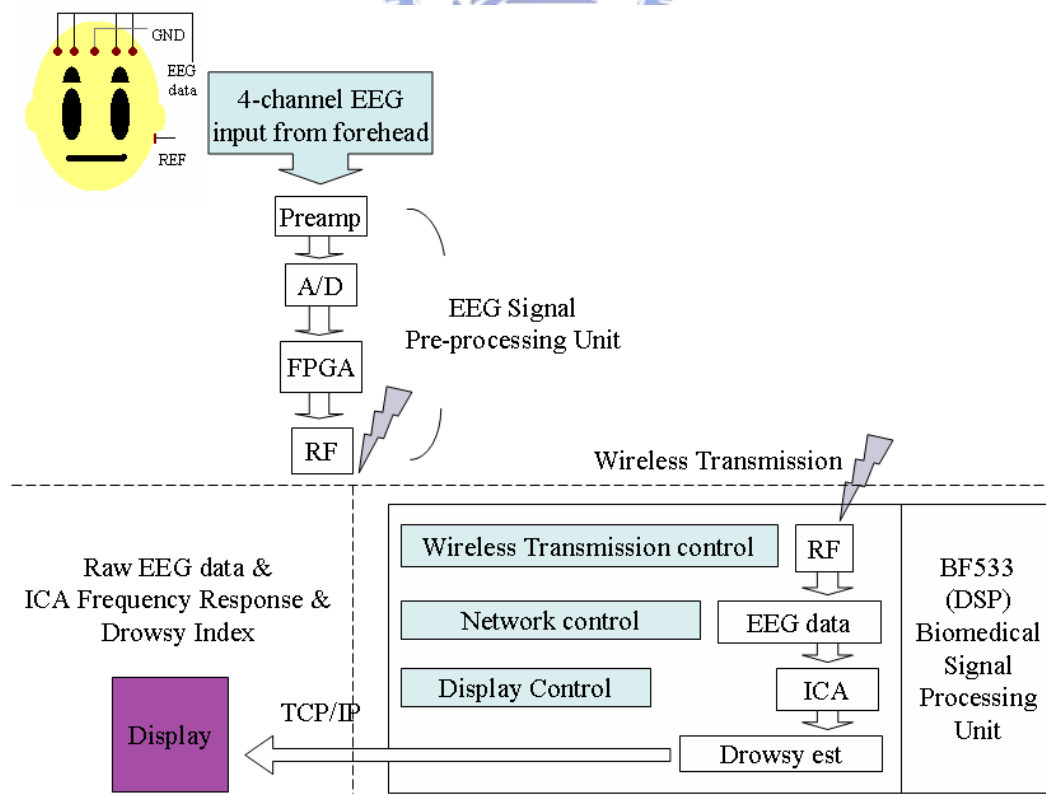


Fig.3-1 System architecture

### 3.2 System Function Block

The architecture of the embedded wireless BCI system we proposed can be divided into three modules: signal acquisition and amplification unit, wireless data transmission unit, and processing and display unit. The block diagram of the proposed system is given in Fig.3-2. The data flow of the system is divided into two paths. (1) The EEG signal is first acquired by signal acquisition and amplifying unit, and then transmitted from wireless data transmission unit to wireless data receiver. Second, the data processing unit will process the EEG data and transmit the raw data to remote PC by TCP/IP at the same time; (2) after processing EEG data, the system will transmit the result to remote PC by Ethernet.

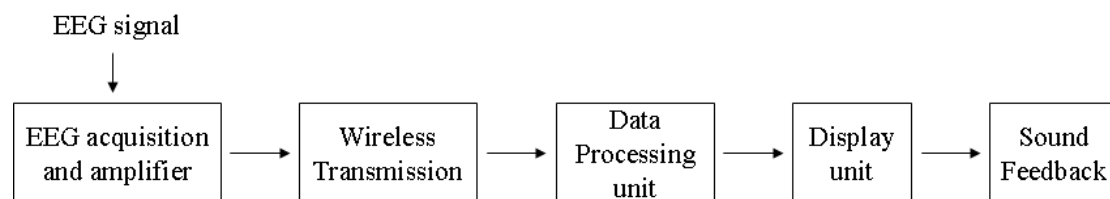


Fig.3-2 Data flow of the system

#### (a) Signal Acquisition and Amplifying Unit

The signal acquisition and amplifying unit is used to measure the EEG signal and filter out the noise. This block diagram is shown in Fig.3-3. The EEG amplifying circuit is constructed of a pre-amplifier with the gain of 100, an isolation amplifier to protect the subject, a band-pass filter which was composed of a low-pass filter and a high-pass filter to reserve 1-100Hz, , a differential amplifier which had the gain of fifty.

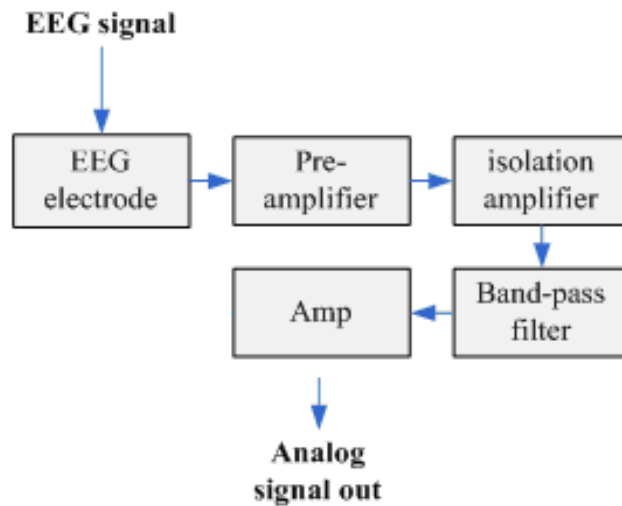


Fig.3-3 Block diagram of the signal acquisition and amplification unit

**(b) Wireless Data Transmission Unit**

Fig.3-4 shows the wireless data transmission unit which converts analog signals to digital signals, and then encodes and transmits through the wireless transmitter and receiver. To do that, we use CPLD (Complex Programmable Logic Device) to control the A/D converter and encode the data for the transmitter.

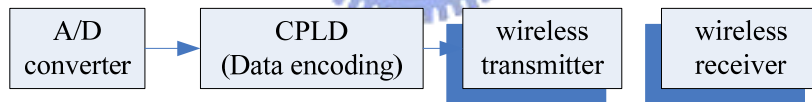


Fig.3-4 Block diagram of the wireless data transmission unit

**(c) Processing, Display and Feedback Unit**

The processing and display unit is the main part of the wireless brain computer interface. The operating core is ADI (Analog Devices Instruments) BF (Blackfin) 533. The DSP processor is helpful for EEG data processing with a large number of mathematical calculations. In this thesis, the DSP processor processes EEG data after managing of peripheral devices, such as wireless receiving devices and network. Then the result will be display and the feedback signal will be outputted.

### 3.3 Requirement Analysis of Hardware and Software

#### 3.3.1 EEG Acquisition Circuit

Generally speaking, the amplitude of bio-signal of human is very small for data processing. Recording EEG is also difficult, because the amplitude of brain wave is wakened through meninges, cerebrospinal fluid, and the scalp. The normal peak-to-peak amplitude of EEG is  $1\sim 150\mu\text{V}$  and frequency band is  $1\sim 100\text{Hz}$ . Hence, identically before processing the EEG signal, we should amplify the EEG signal first. In addition, a filter circuit was also designed to extract signals at desired frequency bands. Moreover, the materials of sensors also affect the recording quality because EEG signals are always easily contaminated. The signal flow of our EEG acquisition circuit and the picture of the EEG amplifier are shown below. First of all, the EEG signals are amplified by 99 times. Then the signals are filtered by a 1-100 Hz band pass. Third, the signals are amplified again by of 51 times. That is, the totally gain is about 5,000. Finally, the processed EEG data are converted to 8-bit digital EEG signals by an analog to digital signal converter. The range of digital EEG signals is 0~254.

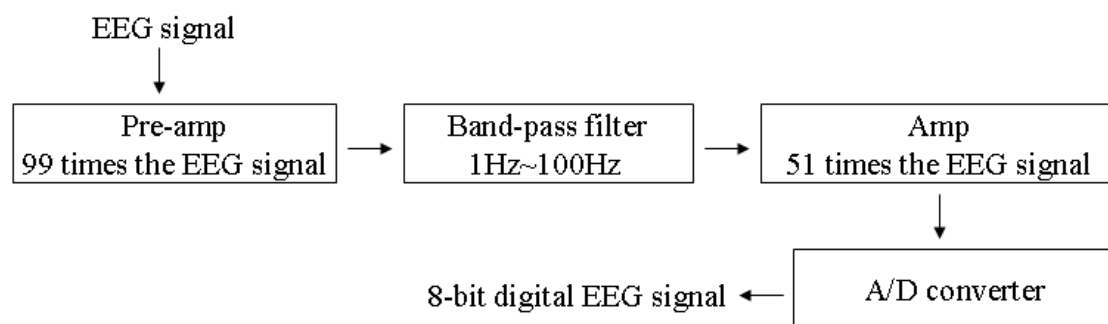


Fig.3-5 The signal flow of EEG signal recording circuit



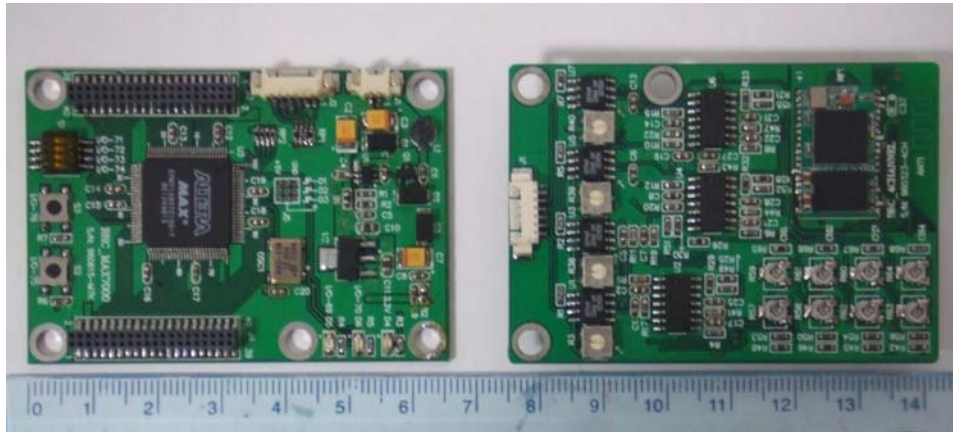


Fig.3- 6 The picture of EEG signal recording circuit

### 3.3.2 Wireless Transmitting Module

With the digital EEG signal output of A/D converter, we use Complex Programmable Logic Device (CPLD) to encode the digital EEG signals for wireless transmission. The encoded wireless packages are transmitted through wireless module RF3105 to RF3100. Here, we use low-power Radio Frequency (RF) wireless transmission protocol. The data flow of wireless transmitting and physical photos are shown in Fig.3-7 and Fig.3-8, respectively. The package format of the wireless transmission is shown in Table 3-1. To ensure the integrity of the EEG data, each converted 8-bit digital EEG signal is embraced in 13-bit transmitting interval. The baud rate of the wireless module we chose is 19200 bit/sec. The total size of the package is  $5 \times 13 \text{ bits} = 65 \text{ bits}$ . That is, the highest transmitting rate is  $19200/65 = 295 \text{ Hz}$ .

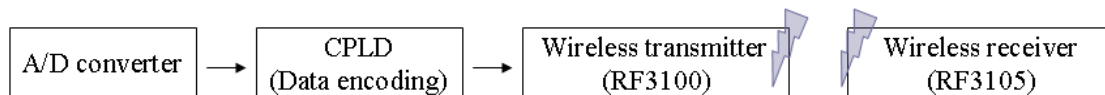


Fig.3-7 The data flow from the A/D converter to the wireless receiver

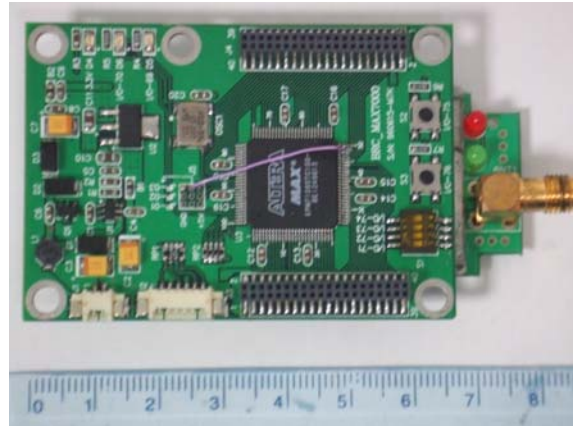


Fig.3-8 The picture of the integrated signal acquisition circuit

Table 3-1 Wireless package definition

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Identification code	EEG Channel 1	EEG Channel 2	EEG Channel 3	EEG Channel 4

RF3105/RF3100 Wireless RS232 Converter converts traditional RS232 to wireless communication [36][37]. There are three selections of frequencies: 433~435 MHz, 868~870 MHz, or 902~928 MHz. It is a transparent transceiver module that integrates low transmission power and high receiving sensitivity designs. RF3105/RF3100 provides high communication speeds up to 76.8 Kbps and long distances up to 600 meters. It meets the wireless regulation and saves the application for approval of frequencies [36][37]. Compared to Bluetooth, RF3105/RF3100 consumes less power (12dBm, which is smaller than 24dBm of Global System for Mobile Communication, GSM) and longer transmission range. The picture of RF3105 and RF3100 is shown below. The specification of RF3105 and RF3100 are listed in Table below.

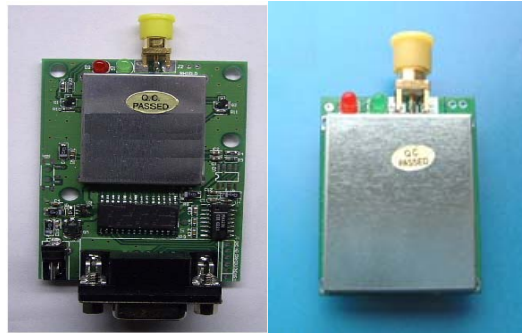


Fig.3-9 RF3105/RF3100 module

Table 3- 2 RF3105/RF3100 modules

Model	RF3100 - C	RF3105 - C
Working Frequency	915MHz	
Modulation	FSK	
Channels	At least 25 channels	
Functions	Half-Duplex, Bi-direction, Transparent wireless data transmit and receive	
Transmission Power	12dBm	
Baud Rates	19200 Kbps	
Communication Distance	200-600m (Whip antenna, outdoor, depending on different communication speeds)	
Port Data Format	8N1	
Type Connector	RS232 (9 pins, D-Sub, Female) Connector	
LED	Red: Transmitting, Green: Receiving	

The flowchart of receiving procedure on wireless receiver is described in Fig.3-10. After reading the ID code which is 255, we continuously read 4 channels of EEG data. Then feed them to the data analysis procedure. To simplify the transmission procedure, our half-duplex transmitting module allows only one

direction during transmission.

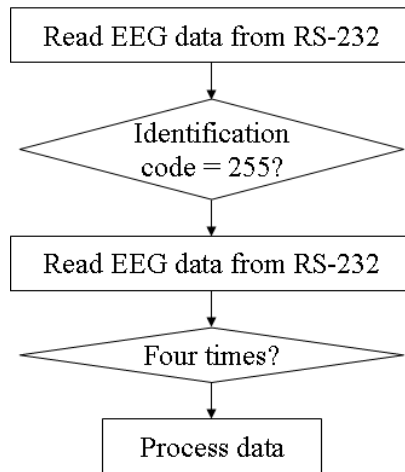


Fig.3- 10 flowchart of receiving procedure

### 3.3.3 Embedded Digital Signal Processor

In this thesis, the selected core processor is ADSP-BF533 (Blackfin 533) developed by Analog Devices Inc. The system diagram of the board we designed is shown in Fig.3-11 below and the picture of the board is shown in Fig.3-12. The Blackfin processor provides both microcontroller (MCU) and DSP functionality in a unified architecture, allowing flexible partitioning between the needs of control and signal processing. If the application demands, the Blackfin processor can act as 100% MCU (with code density on par with industry standards), 100% DSP (with clock rates at the leading edge of DSP technology), or a combination of the two. The maximum high performance of BF533 processor can be up to 500MHz. It has two 16-bit MACs, two 40-bit ALUs, four 8-bit video ALUs, and 40-bit shifter. One of its features is RISC-like register and instruction model for ease of programming and compiler-friendly support. The board is designed to support the development and porting of open-source  $\mu$ Clinux applications and includes the full complement of memory along with serial and network interfaces. Besides an ADSP-BF533 500 MHz Blackfin processor, the board includes:

- 16 MB SDRAM (64M x 16 bits) and 4 MB FLASH memory
- RS-232 serial interface
- 6 Keypads and 240\*320 pixels LCD
- JTAG interface for debug and FLASH programming
- Bluetooth transmitting/ receiving module

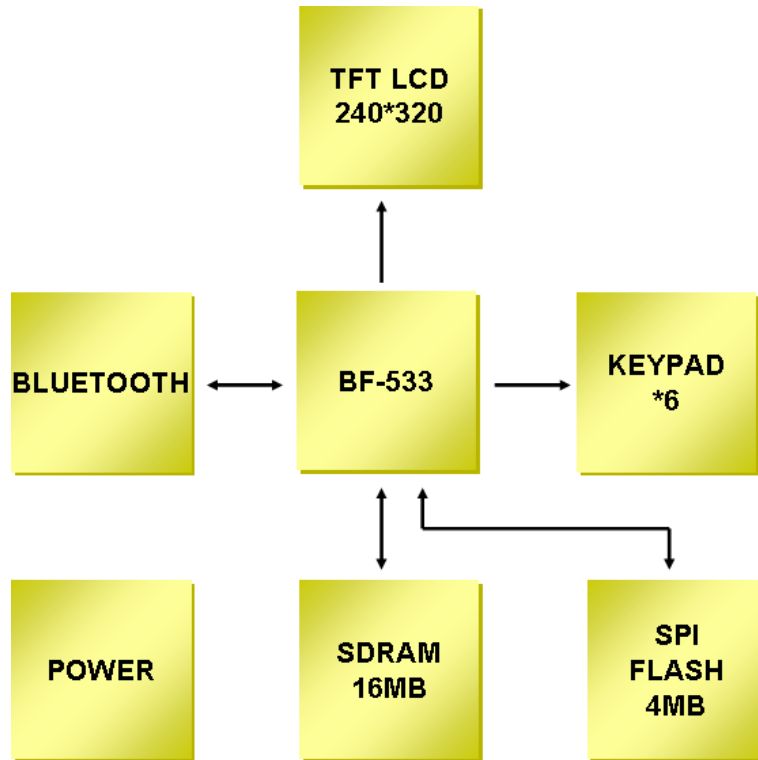


Fig.3-11 The system diagram of the board

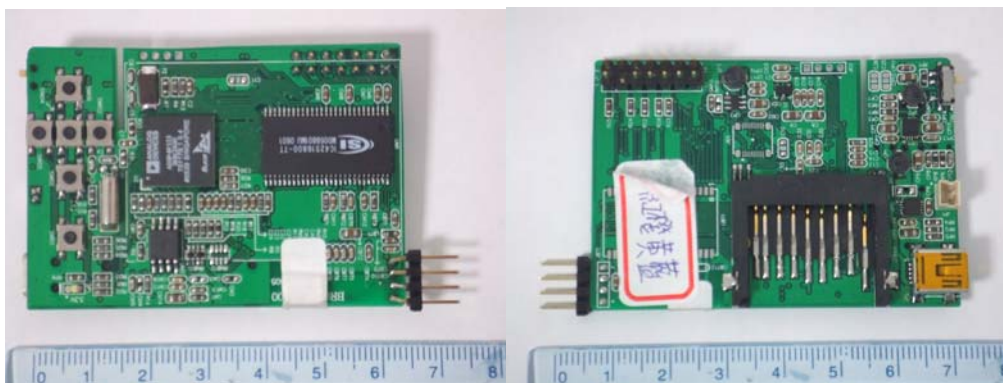


Fig.3-12 Picture of the board (upside and downside)

### 3.4 Development of the System

#### 3.4.1 Boot Loader

Boot loader is executed before entering operating system. It initializes processor stack, register, and basic peripheral devices and brings up the operating system. The setting of Boot loader is concerned with processor type and operating system, and its function is similar to BIOS of PCs. As a result, each boot loader should've been modified and ported properly according to different types of processors. The boot loader we use is U-Boot. The U-Boot boot up flow is described in Fig.3-13 The U-boot entry which is `u-boot_1.1.3/cpu/bf533/start.o` is defined in `u-boot_1.1.3/board/stamp/u-boot.lds`.

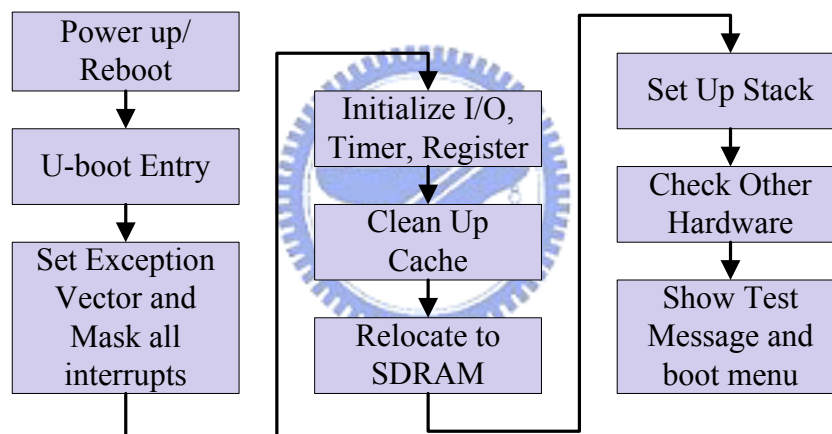


Fig.3-13 U-Boot boot up flow

```
[may@brc2 stamp]$ make stamp_config (1)
```

```
[may@brc2 stamp]$ make (2)
```

To obtain U-boot, configure settings for BF533 STAMP as Step (1). Then type 'make' to compile U-Boot image. After compiling the U-Boot image, `u-boot.bin` will be generated. The `u-boot.bin` is a binary file that can be used to upgrade U-Boot if an older version of U-Boot is already installed on the target system. Finally, the U-Boot image should be downloaded to the target board. There are three methods to

download U-Boot image. One is to download image via ICE, others are downloading via network or UART (Universal Asynchronous Receiver Transmitter) port. The detailed steps are described below.

**(a) Downloading Image via VDSP++**

Before downloading U-Boot, u-boot.hex must be generated first. The u-boot.hex file is not created at that time when U-Boot was compiled. It is necessary if we want to use Analog Device's VDSP++ programmer to program U-Boot into a platform. We can create u-boot.hex with Step (3):

```
[may@brc2 stamp]$ bfin-elf-objcopy -I binary -O ihex u-boot.bin u-boot.hex (3)
```

To download u-boot.hex image, first open the VDSP++ (version 4.0 or higher version is recommended) and set up a session for BF533 STAMP and connect the JTAG line between the host PC and the target. After setting up and successfully open a session, use the Toos → Flash Programmer to modify the contents of flash.

The flash programmer allows users to erase and modify the contents of flash ROM. First, we should load a driver. Downloading the flash device driver program (.dxe) onto the processor facilitates loading the flash image onto the flash chip. To make sure that there are no others un-cleaned fracture on the flash ROM, we can enable the advanced options and erase all segment on the flash ROM. Finally, we load the u-boot.hex. onto the target, the bar below shows the programming progress. Programming virgin flash will be slow (approximately twenty minutes). After that, unplug the JTAG and connect the UART line between the host PC and the target, and start a hyper terminal. Use 57600 (default baud rate) , 8 data bits, no parity check, one stop bit, and flow control disabled to connect as in Fig.3-14. After resetting the target board, the boot message will be shown as Fig.3-15. If the boot message does not show, make sure the S1 shown in Fig.8 is set to 00 (booting from external memory).

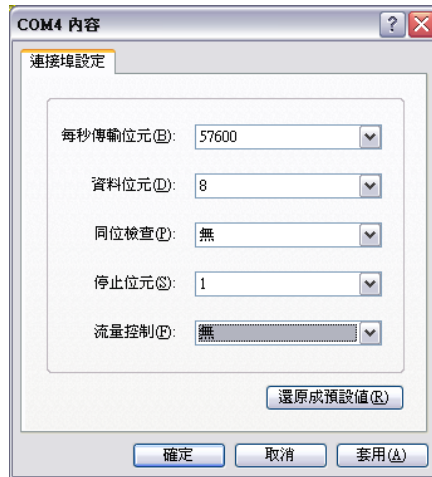


Fig.3- 14 Setting the UART port

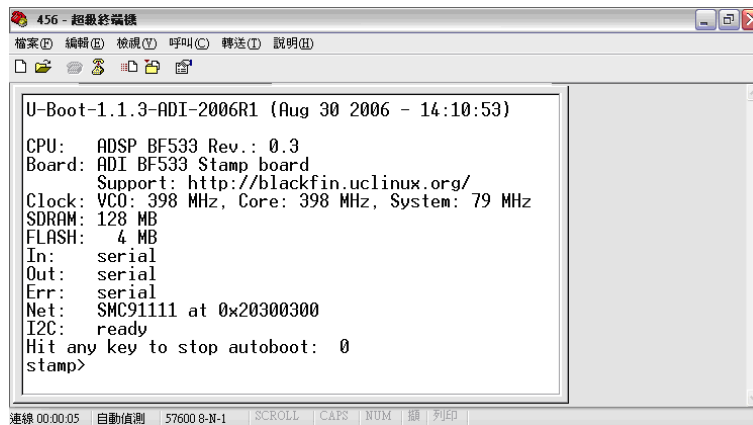


Fig.3-15 Boot up message

### (b) Downloading Image from UART

If there already a U-Boot image exists in flash ROM, then we can use the original U-Boot to erase flash and download new U-Boot image. Before erasing U-Boot image via console, the new U-boot must be assured to be feasible, because if the new U-boot is not feasible and you do not have ICE, the target will not boot up normally. To load image via UART, type loadb via console as Step (4), the system will be halted to wait for image transfer. Then use Transfer→Transfer files to select u-boot.bin file to download. In the meantime, Kermit protocol should be selected. After transferring the u-boot.bin, the U-Boot image will be temporarily loaded in



SDRAM. An example map of U-Boot image location (assume image size is 0x1B800h) after transferring is shown in Fig.3-16 . To write U-Boot image to flash ROM, disable flash write protection as step (5). Then erase entire flash by step (6).

```
stamp>load (4)
stamp>protect off all (5)
stamp>erase all (6)
```

After erasing entire flash ROM, copy U-Boot image from memory to flash as step (7). The \$(filesize) parameter which is the size of new U-Boot is determined by original U-Boot. The map of U-Boot image location after writing U-boot image and target resetting is shown in Fig.3-17.

```
stamp> cp.b 0x1000000 0x20000000 $(filesize) (7)
stamp> reset (8)
```

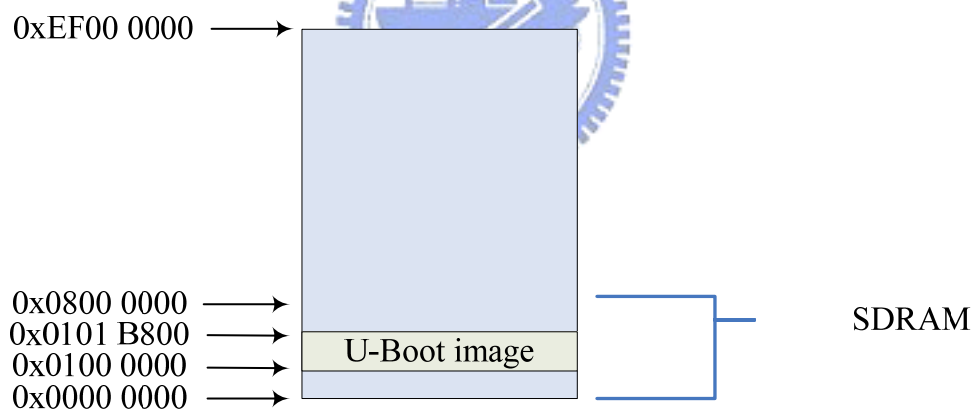


Fig. 3-16 Map of U-boot in external memory

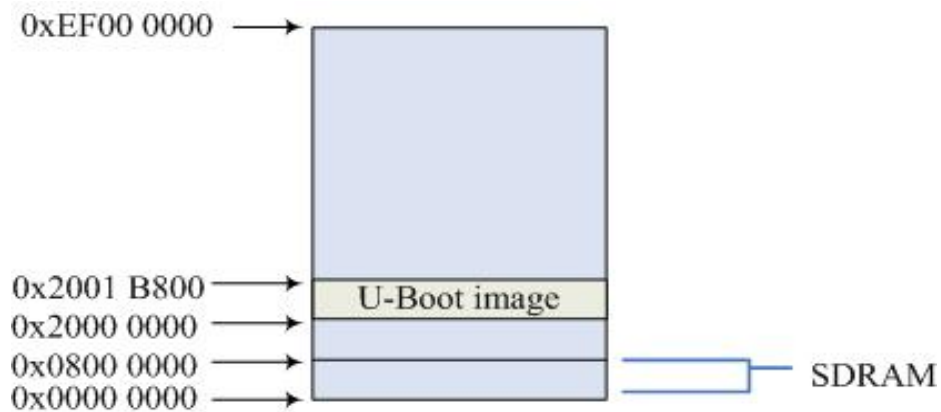


Fig.3- 17 Map of U-boot in external memory after loading U-boot and resetting

### (c) Downloading Image Network (TFTP)

Downloading image via TFTP is far faster than downloading image via UART, hence, technical personnel tend to use TFTP to download image. In the first place, tftpd packages should be installed on host PC. After that, put u-boot.bin on host to directory /tftpboot on target U-Boot then do as step (9) to download image.

Stamp> tftp 0x1000000 u-boot.bin	(9)
----------------------------------	-----

### 3.4.2 Operating System

$\mu$ Clinux is an operating system that is derived from the Linux kernel. It is intended for microcontrollers without Memory Management Units (MMU's). It is available on many processor architectures, including the Blackfin processor. The basic compiling step is listed as step (10) to (16). First, decompress  $\mu$ Clinux kernel source code, then configure  $\mu$ Clinux and a configuration prompt will show up. Second, choose "BF533-STAMP" as "Vendor/Product Selection" and exit from "Vendor/Product Selection". Third, enter "Kernel/Library/Defaults Selection" and Check "Customize Vendor/User Settings" and "Customize Vendor/User Settings", after that, save and exit from main configuration menu. Finally, we should set kernel

and user application. Do step (11) to clean up all prior object code. Next, do Step (12) to compile kernel objects. Do Step (13) to compile user application. After Step (14), we can customize romfs. The file in romfs will be described in later. There will be some error after step (15), it could be ignore here. After step (32), repeat Step (14) to (16) to obtain final kernel image. Kernel image is in directory uCliunix-dist/images.

[may@brc2 stamp]\$make menuconfig	(10)
[may@brc2 stamp]\$make clean	(11)
[may@brc2 stamp]\$make lib_only	(12)
[may@brc2 stamp]\$make user_only	(13)
[may@brc2 stamp]\$make romfs	(14)
[may@brc2 stamp]\$make image	(15)
[may@brc2 stamp]\$make	(16)

After downloading the kernel image, do the step (17) to erase address 0x20020000 to 0x203FFFFFF. Then copy the kernel image to address 0x20020000. After that, use U-Boot boot command as step (19) to set auto booting, getting kernel from 0x20020000. Then save changed U-Boot environment variables and reset target. Since the U-Boot image locates before address 0x20020000, we can take advantage of the rest space of flash ROM bank 0 and other flash ROM banks to load the kernel image. Take RAM based file system which means that the kernel contains file system for example, its map (assume image size is 0x180000h) after writing the kernel image and resetting the target is shown in Fig.3-18.

stamp> erase 0x20020000 0x203FFFFFF	(17)
stamp> cp.b 0x1000000 0x20020000 \$(filesize)	(18)
stamp> setenv bootcmd bootm 0x20020000	(19)
stamp> save	(20)

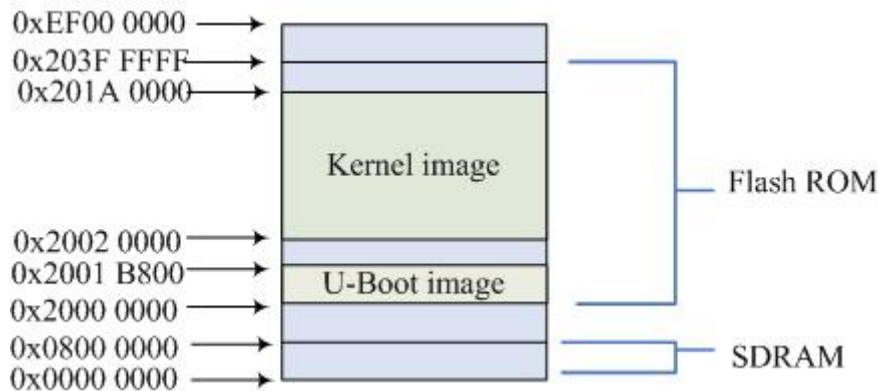


Fig.3-18 Map of kernel in external memory after loading kernel and resetting

One of the most important features of Linux is its support for many different file systems. This makes it very flexible and well able to coexist with many other operating systems. In Linux file system, everything starts from the root directory, represented by '/', and then expands into sub-directories. A summary of the directories in this Linux file system are given below:

- **/bin** - (Binaries) This directory contains the executable binaries for programs that were selected in the application configuration window when compiling the kernel.
- **/dev** - (Devices) On the target system this directory will contain the device files for the system. For more information see Device Files.
- **/etc** - (Etc.) This directory contains various system configuration files.
- **/home** - (Home) This directory provides a location for system users to store their files.
- **/lib** - (Libraries) This directory is for shared libraries.
- **/mnt** - (Mount) This directory contains any mounted file systems.

- **/proc** - (Processes) This directory stores virtual files which contain information on system processes.
- **/tmp** - (Temporary) In  $\mu$ CLinux this directory is a link to `/var/tmp`.
- **/usr** - (User) This directory stores various user programs.
- **/var** - (Various) Unlike the other directories which exist on the device `/dev/root`, this directory exists on the device `/dev/ram0`, this makes it the only writable directory as the other directories exist on a read-only file system.

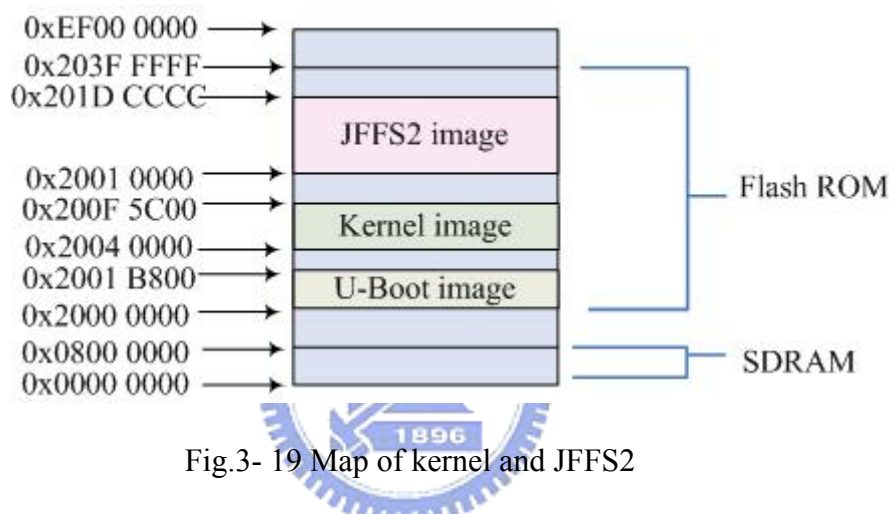


Fig.3- 19 Map of kernel and JFFS2

If we compile the kernel without any change, the kernel will be too big to fit in flash memory; hence, to control the size of kernel image is very important. The first image I compiled was too big. (3.5M is larger than the rest flash space on BF533. ) We have to erase more sectors on the flash. As shown in Fig.3-20 below, Four ASYNC MEMORY BANKS were available. But the first bank is restored for U-Boot. So we take advantage of the rest space of bank 1, the start address of the rest space could be the start address of kernel. For safety, we keep a space between U-Boot's end address and kernel's start address. On the other side, some unneeded settings must be removed from the kernel configuration such as `expand`, `version`, `crontab`, `ftp`, `telnet`, `wget`, `tar`, etc. That is, the smaller the kernel image is, the lower the cost is.

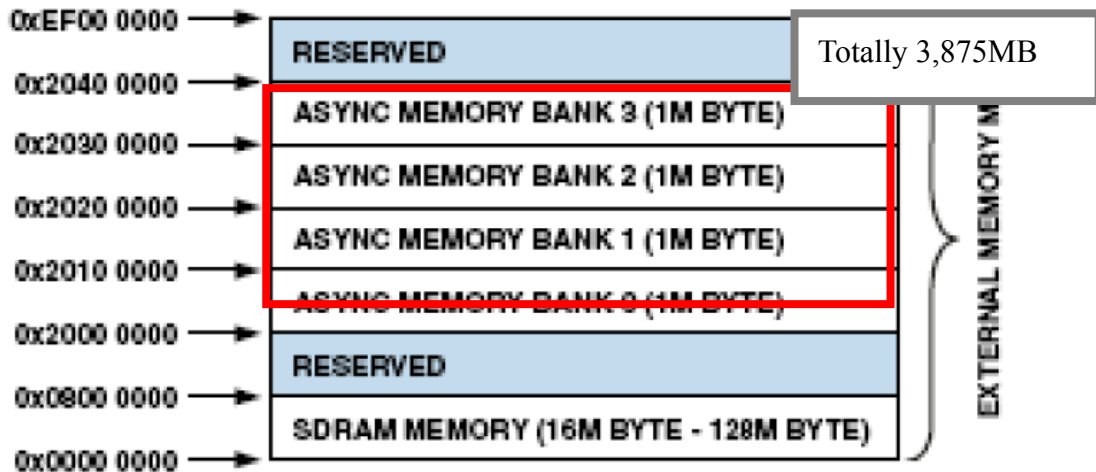


Fig.3- 20 Map of rest external memory

Linux is one of the most prominent examples of free software and open source development; its underlying source code can be modified, used, and redistributed by anyone, freely. The original uClinux was a derivative of Linux 2.0 kernel intended for microcontrollers without Memory Management Units (MMUs). However, the Linux/Microcontroller Project has grown both in brand recognition and coverage of processor architectures. Today's uClinux as an operating system includes Linux kernel releases for 2.0 2.4 and 2.6 as well as a collection of user applications, libraries and tool chains. It's popular for its portability and strong support for many applications. Compare to other operating system, as describe in Table.3-1, uClinux perform better on BF533 than others.

Table 3- 3 Comparison to Other Operating Systems

Remark\OS	μClinux	WinCE	uC/OS-II
Scheduling	round robin	preemptive	preemptive
Network	good	good	inferior
Stability	good	inferior	good

Open source	yes	no	yes
Cross platform	easy	no	difficult
System Resource	larger	largest	small
Prices	free	expensive	No free

### 3.4.3 User Program

Cross-platform development and programming techniques are used for developing embedded applications. The cross-platform development framework is shown as Fig.3-21. Owing to the lack of available resources on embedded systems, the cross-compilation, code editing, code linking, and source debugging are performed on powerful host PC. The system software, operating system, and application programs are first compiled as object codes. Then the linker on the host links up object codes and forms an executable image. Thus we have to understand thoroughly how executable images were downloaded to the target embedded systems. Executable image can be downloaded via internet, UART (Universal Asynchronous Receiver Transmitter), and ICE (In- Circuit Emulator).

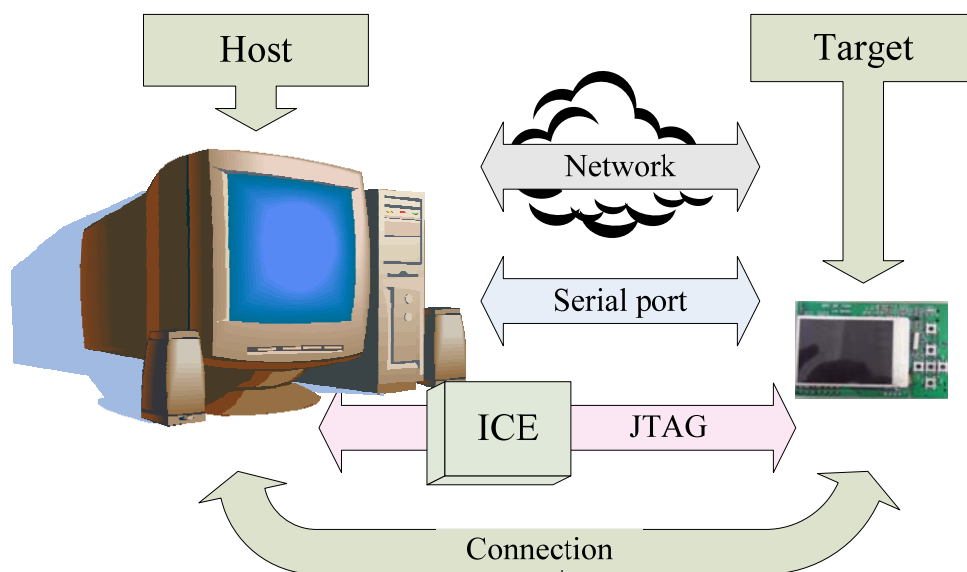


Fig.3- 21 Cross-platform development framework

To run user application on target board, the program must be cross-compiled first. We use the cross-compiler to cross-compile an example program as described in step (22). Following steps are executed on target operating system. With the network file system or RAM-based file system described before, we can execute the program without a doubt. The flow to create an executable image on a host PC is described in Fig.3-22.

```
[may@brc2 stamp]$ bfin-uclinux-gcc -Wl,-elf2flt hello.c -o hello      (22)
root:~>chmod 777 hello                                           (23)
root:~>./hello                                                    (24)
```

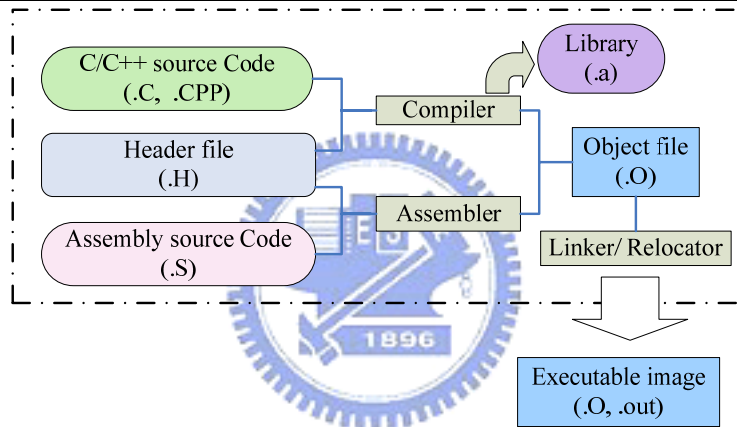


Fig.3- 22 Flow to create an executable image

### 3.5 Optimization of the System

#### 3.5.1 Floating Point to Integer

With the hardware constraint of embedded systems, plenty modification of source code was made. The main issue on the embedded DSP (digital signal processor) was lack of FPU (Floating Point Unit). Due to the floating point of non-FPU DSP is emulated, the original execution time takes 22 seconds because there are large numbers of floating variables of original source code. The solution is to make commonly used variable integer. That is to multiply the floating with 100000 to preserve decimal points. After that, divide 100000 to retain the integer result. This



method speed up execution time to 24% the original execution time.

### **3.5.2 Programming**

#### **(a) TLU (Table-Look-Up Techniques)**

A lookup table is a data structure, usually an array or associative array, used to replace a runtime computation with a simpler lookup operation. The speed gain can be significant, since retrieving a value from memory is often faster than undergoing an expensive computation.

#### **(b) Loop Unrolling**

Loop unrolling is a technique for optimizing parts of computer programs — a member of the loop transformation family. The idea is to save time by reducing the number of overhead instructions that the computer has to execute in a loop. For example, for the modification of loop unrolling, the unrolled program has to make only 20 loops, instead of 100.





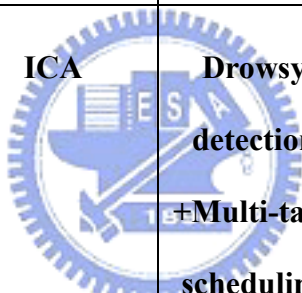
### **3.5.3 Variable Regulation**

The rest modification was made on variables of ICA. Five modifications is done, First the initial learning rate is modified to 0.001. Second, the stop weight change is modified to 0.001, too. In addition, momentum = 0.4 is added to ICA. With loading the last weight matrix, max steps can be restricted and the training steps in once can be separated into other times.

### 3.6 Comparison to Past BCI Systems

A comparison table is listed below. The system we proposed is with online ICA and drowsiness detection. Moreover, it's provided with 4 channel EEG analysis.

Table 3-4 Comparison top past BCI systems

	PC	2004 (OMAP)  30cm*20cm	2005, 2006 (OMAP)  30cm*20cm	2007 (BF533)  18cm*13cm	2007 (BF533)  6.5cm*4cm
Our Application	N/A	ICA  +Multi-task scheduling	Drowsy detection	ICA+ Drowsy detection	ICA+ Drowsy detection
EEG Channels	N/A	N/A	2 channels	4 channels	4 channels
Data Processing	N/A	Offline	Online	Online	Offline
# of CPU	Single-core	Dual-core		Single-core	
CPU Core	1.3v	1.5v		0.8 V to 1.26 V	

<b>Voltage</b>			
<b>Maximum Frequency</b>	<b>3GHz</b>	<b>DSP:192MHz</b> <b>ARM:168MHz</b>	<b>DSP:600MHz</b>
<b>Prices</b>	<b>\$119</b>	<b>\$15</b>	<b>\$5</b>



## Chapter 4 On-line Independent Component Analysis Implementation

Flowchart of the EEG signal processing is shown in Fig.4-1. The 4-channel EEG data was first recorded via the EEG amplifier we designed. Then change the sampling rate to 64Hz. It consists of down sampling, independent component analysis, useless component rejection, short time FFT and online drowsiness detection. The independent component analysis is applied to the down sampled EEG signals to obtain the independent components. After that, the useless components are rejected according to the standard deviation of independent components. The effectiveness of eye blinking and other artifacts removal by using ICA had been demonstrated in the Jung et al.'s study[1][2]. The short time FFT is then applied to the useful independent components to calculate the frequency response of ICA components. Finally, the feature of drowsiness symptoms can be evaluated by the online drowsiness detection analysis.

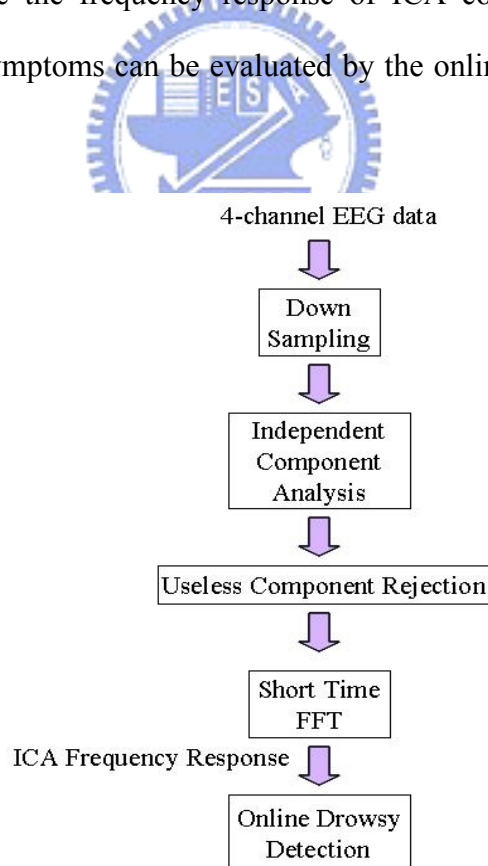


Fig.4- 1 Flowchart of the EEG signal procedure

## 4.1 Independent Component Analysis

Independent component analysis (ICA) is a case of blind source separation, that is, we assume the signals may be mutually statistically independent or decorrelated. The goal of ICA is to separate signals (components) from a set of mixed signals without the aid of information (or with very little information) about the nature of the signals. A familiar case of ICA is the “cocktail party problem”, where the underlying speech signals are separated from a sample data consisting of people talking simultaneously in a room. Usually the problem is simplified by assuming no time delays and echoes. An important note is to consider that if  $N$  sources are present, at least  $N$  observations (i.e. microphones) are needed to get the original signals. Figure 4-2 shows that we have  $N$  sources and mix them with a mixing matrix to get observed signals.

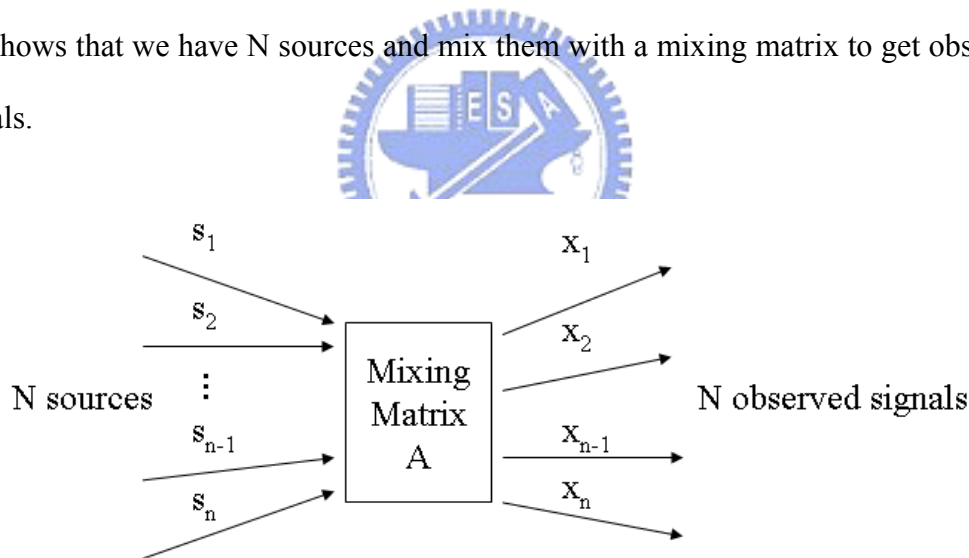


Fig.4- 2  $N$  mixed signals and  $N$  sources

The processing flow shown in Fig.4-1 of EEG signals is backwards. Its linear expression is represented in Fig.4-2, where  $A$  is described in 4-2. Where the  $s$  is unknown independent component and the element  $a$  is unknown mixing coefficient. The expression 4-1 can be rewritten as 4-3. This is called ICA model. After ICA, the

mixing matrix will be estimated; its inverse matrix is call unmixing matrix as described in expression 4-4. Hence the original sources can be obtained via Eq. 4-5.

$$\mathbf{x}_1 = \mathbf{a}_{11}\mathbf{s}_1 + \mathbf{a}_{12}\mathbf{s}_2 + \dots + \mathbf{a}_{1n-1}\mathbf{s}_{n-1} + \mathbf{a}_{1n}\mathbf{s}_n$$

$$\mathbf{x}_2 = \mathbf{a}_{21}\mathbf{s}_1 + \mathbf{a}_{22}\mathbf{s}_2 + \dots + \mathbf{a}_{2n-1}\mathbf{s}_{n-1} + \mathbf{a}_{2n}\mathbf{s}_n \dots\dots\dots$$

$$\dots \tag{4-1}$$

$$\mathbf{x}_{n-1} = \mathbf{a}_{n-1,1}\mathbf{s}_1 + \mathbf{a}_{n-1,2}\mathbf{s}_2 + \dots + \mathbf{a}_{n-1,n-1}\mathbf{s}_{n-1} + \mathbf{a}_{n-1,n}\mathbf{s}_n$$

$$\mathbf{x}_n = \mathbf{a}_{n,1}\mathbf{s}_1 + \mathbf{a}_{n,2}\mathbf{s}_2 + \dots + \mathbf{a}_{n,n-1}\mathbf{s}_{n-1} + \mathbf{a}_{n,n}\mathbf{s}_n$$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1,n-1} & a_{1,n} \\ a_{21} & a_{22} & \dots & a_{2,n-1} & a_{2,n} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \dots & a_{n-1,n-1} & a_{n-1,n} \\ a_{n,1} & a_{n,2} & \dots & a_{n,n-1} & a_{n,n} \end{bmatrix} \tag{4-2}$$

$$\mathbf{X} = \mathbf{A}\mathbf{S} \tag{4-3}$$

$$\mathbf{W} = \mathbf{A}^{-1} \tag{4-4}$$

$$\mathbf{A}^{-1} \mathbf{X} = \mathbf{A}^{-1} \mathbf{A}\mathbf{S} \rightarrow \mathbf{S} = \mathbf{W} \mathbf{X} \tag{4-5}$$

One approach with some information on the statistical properties of the signal  $\mathbf{S}$  is to use to estimate  $\mathbf{A}$  and to find solution of Eq. 4-5. The statistical method finds the independent components (aka factors, latent variables or sources) by maximizing the statistical independence of the estimated components. Non-Gaussianity, motivated by the central limit theorem, is one method for measuring the independence of the components with kurtosis or approximations of negentropy [49]. Mutual information

is another popular criterion for measuring statistical independence of signals. The kurtosis of  $y$  is classically defined by

$$\mathbf{kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2 \quad (4-6)$$

We assume that  $y$  is of unit variance, the expression can be simplified to  $E\{y^4\} - 3$ . For a Gaussian  $y$ , the  $E\{y^4\}$  equals  $3(E\{y^2\})^2$ . Thus, kurtosis is zero for a Gaussian random variables, kurtosis is nonzero. Kurtosis can be both negative and positive. Random variables that have a negative kurtosis are called sub Gaussian, and those with positive kurtosis are called super Gaussian. Super Gaussian has a spiky probability density function with heavy tail. For example, brain waves are super Gaussian variables while sinusoid and cosine waves are sub Gaussian variables.

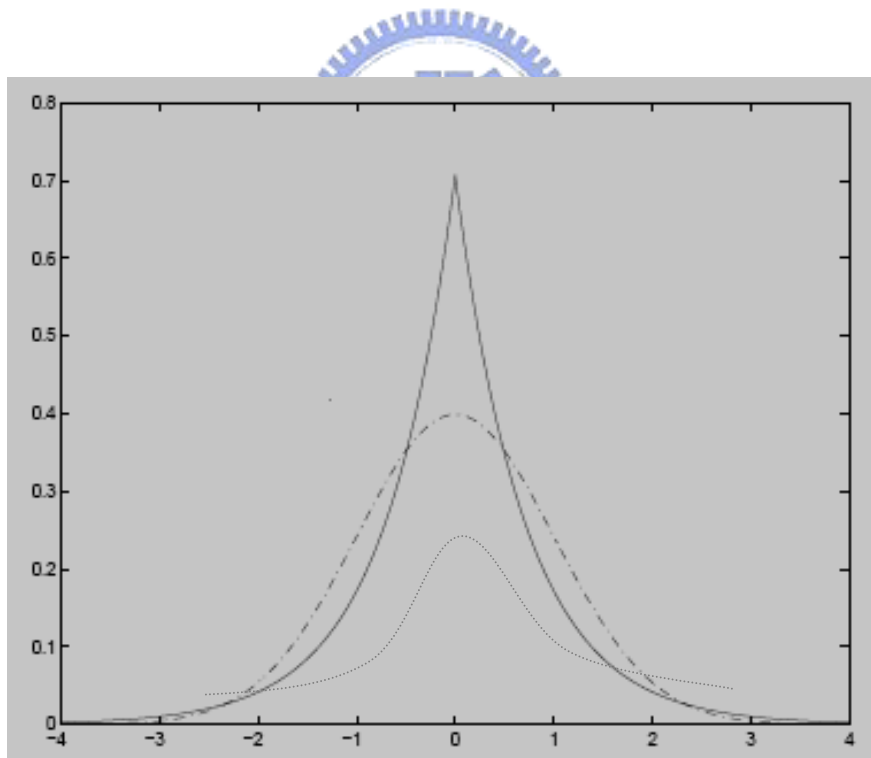


Fig.4- 3 The figure of typical super Gaussian, normal Gaussian, and sub Gaussian

A second important measure of nongaussianity is given by negentropy.

Negentropy is based on the information-theoretic quantity of entropy. Entropy  $H$  is defined for a discrete random variable  $Y$  as

$$H(Y) = -\sum P(Y = a_i) \log P(Y = a_i) \quad (4-7)$$

where  $a_i$  are the possible values of  $Y$ . The differential entropy is defined as

$$H(y) = -\int f(y) \log f(y) dy \quad (4-8)$$

Another often used version is called negentropy. Negentropy  $J$  is defined as follows

$$J(y) = H(y_{\text{gauss}}) - H(y) \quad (4-9)$$

The  $J$  can be simplified as

$$J(y) \approx \frac{1}{12} E\{y^3\}^2 + \frac{1}{48} \text{kurt}(y)^2 \quad (4-10)$$

or

$$J(y) \approx \sum_{i=1}^P k_i [E\{G_i(y)\} - E\{G_i(v)\}]^2, \quad (4-11)$$

where  $k_i$  are some positive constants, and  $v$  is a Gaussian variable of zero mean and unit variance. The variable  $y$  is assumed to be of zero mean and unit variance, and the functions  $G_i$  are some nonquadratic functions. Commonly used  $G$  are

$$G_1(u) = \frac{1}{a_1} \log(\cosh(a_1 u)) \quad (4-12)$$

and

$$G_2(u) = -\exp(-u^2/2) \quad (4-13)$$



and

$$\mathbf{G}_3 = \tanh(\mathbf{a}_1 \mathbf{u}), \quad (4-14)$$

where  $1 \leq a_1 \leq 2$  is some suitable constant.

Typical algorithms for ICA use centering, whitening and dimensionality reduction as preprocessing steps in order to simplify and reduce the complexity of the problem for the actual iterative algorithm. Without loss of generality, we can assume that both the mixture variables and the independent components have zero mean: If this is not true, then the observed signals can always be centered by subtracting the sample mean, which makes the model zero-mean. Whitening and dimension reduction can be achieved with principal component analysis or singular value decomposition. Whitening ensures that all dimensions are treated equally a priori before the algorithm is run. Algorithms for ICA include infomax [50], FastICA [51] and JADE [52], but there are many others also.

Most ICA methods are not able to extract the actual number of source signals, the order of neither the source signals, nor the signs or the scales of the sources. ICA is important to blind signal separation and has many practical applications. It is closely related to (or even a special case of) the search for a factorial code of the data, i. e., a new vector-valued representation of each data vector such that it gets uniquely encoded by the resulting code vector (loss-free coding), but the code components are statistically independent.

Similarly, we consider electrical recordings of brain activity as given by an EEG. The EEG data consists of recordings of electrical potentials in many different locations on the scalp. These potentials are presumably generated by mixing some

underlying components of brain activity. This situation is similar to the cocktail-party problem: we would like to find the original components of brain activity, but we only observe mixtures of the components. ICA can reveal interesting information on brain activity by giving access to its independent components.

As shown in Fig.4-4, the brain activity recorded at one point on the scalp is the mixture of electrical potentials from many different locations in the brain.

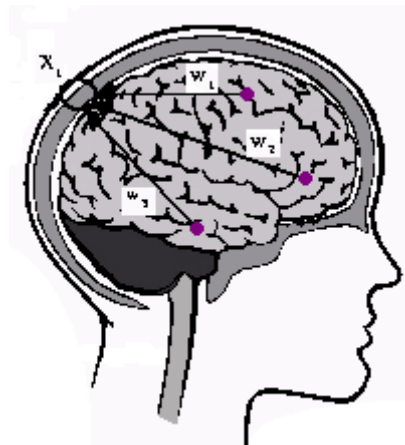


Fig.4-4 The brain activity recorded at one point

The general framework for independent component analysis was introduced by Herault and Jutten in 1986 [53] and was most clearly stated by P. Comon in 1994 [54]. In 1995, T. Bell and T. Sejnowski introduced a fast and efficient ICA algorithm based on infomax [50], a principle introduced by R. Linsker in 1992. In 1997 [55], S. Amari realized that the infomax ICA algorithm could be improved by using the natural gradient [56], which was independently discovered by J.F. Cardoso [52]. However, the original infomax ICA algorithm with sigmoidal nonlinearities was only suitable for super-Gaussian sources. T.W. Lee, in collaboration with M. Girolami, developed an efficient extended version of the infomax ICA algorithm that is suitable for general non-Gaussian signals [57].

The flow of Infomax ICA is shown in Fig.4-5. Centering the data can simplify the ICA algorithm, and the mean can be added back to the data. Whitening means that we remove any correlations in the data, i.e. the different channels are forced to be uncorrelated. Then initialize the weight, and after random permutation, find the maximization entropy output. If the weight change is smaller than the desired weight change then the training is stopped.

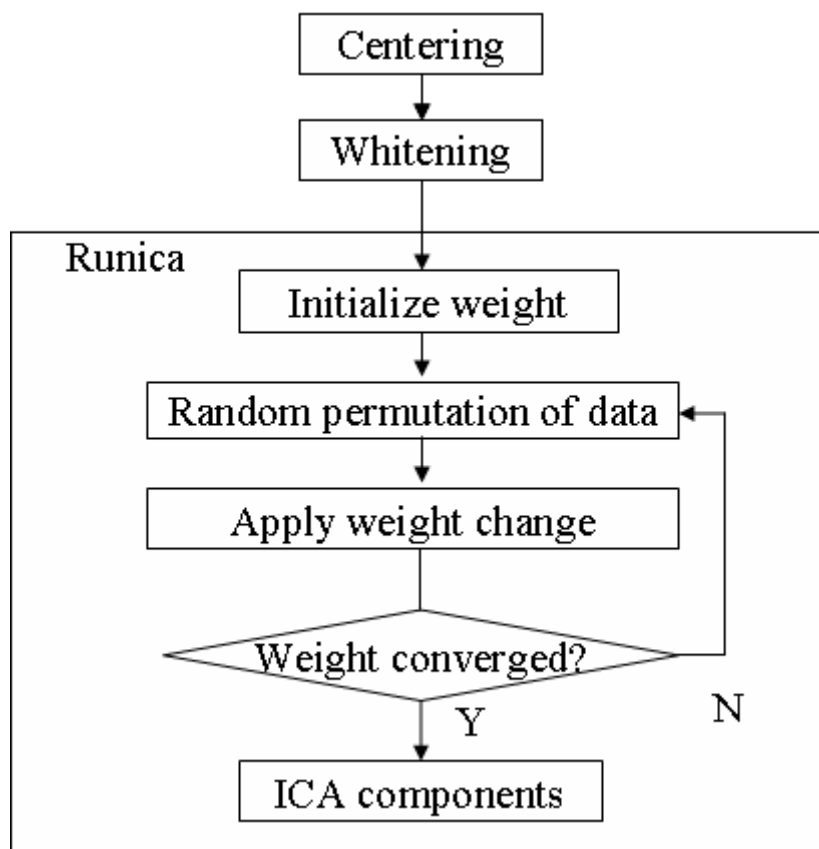


Fig.4- 5 Flow of ICA training

## 4.2 Rejecting Components with Standard Deviation

Here, we use the art of standard deviation to remove components from estimated components. To be briefed, the standard deviation is a measure of the spread of its values. The standard deviation of the original signal is very big. That is, we can't get much information in such noisy signals. After ICA, the unwanted components can be easily rejected by experienced EEG analysis researchers. However,

artificial rejection of unwanted components can only be done offline. With standard deviation, we can reject components with bigger standard deviation such as eye blinking judged by its standard deviation. As we can see in Fig.4-6, the observed signals are heavily contaminated by eye blinking. The standard deviation is also very big. After ICA, the estimated components are shown in Fig.4-7. Evidently, the eye blinking can be rejected due to its biggest standard deviation.

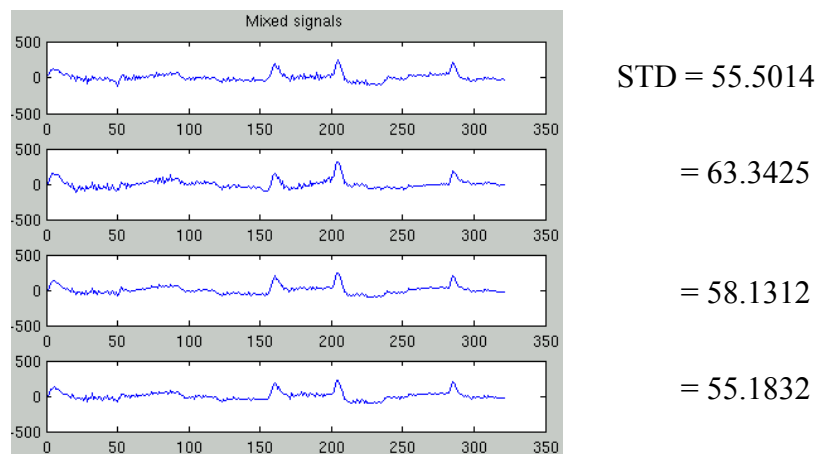


Fig.4- 6 The observed 4-channel mixtures signals

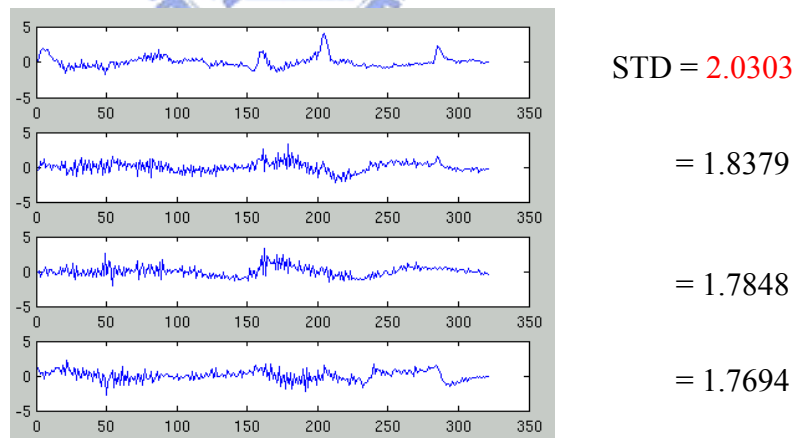


Fig.4- 7 The result of informax ICA, estimated by the signals in Fig.4-6

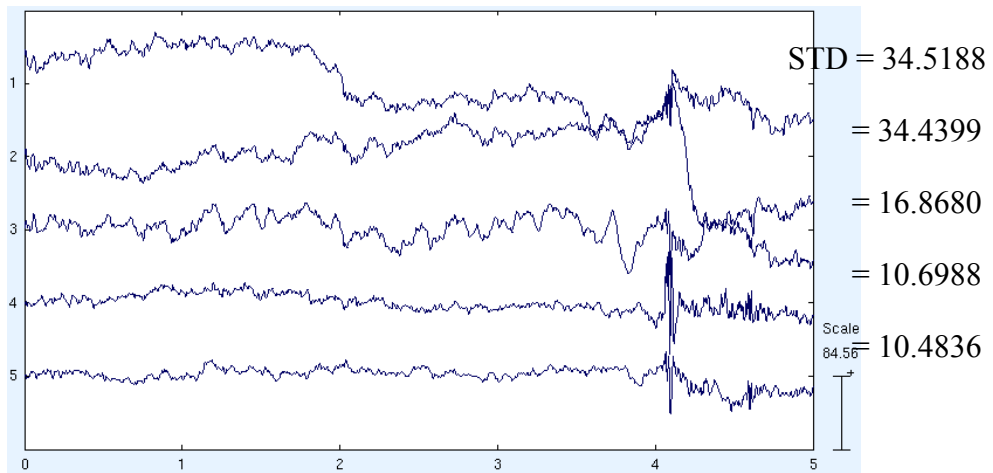


Fig.4- 8 The observed 5-channel mixtures signals

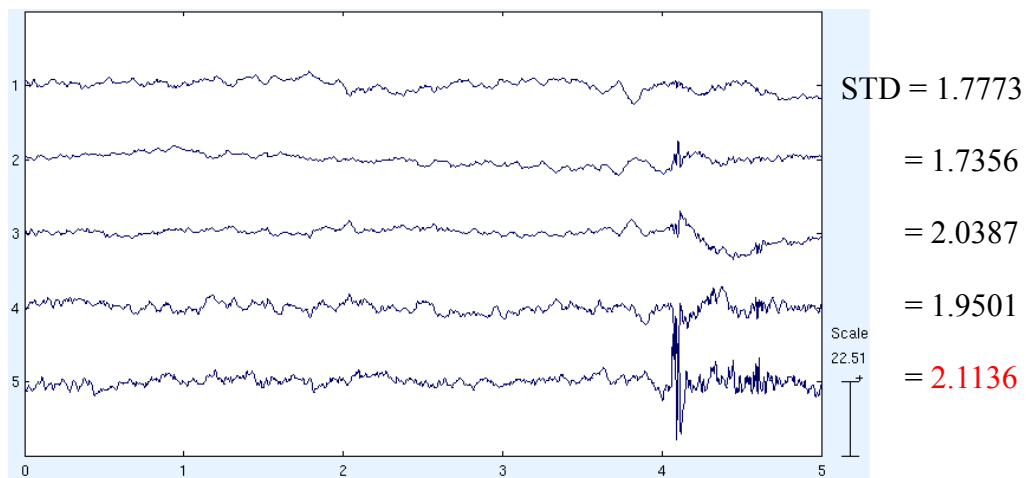


Fig.4- 9The result of informax ICA, estimated by the signals in Fig.4-8

As we can see in Fig.4-8, the observed signals are heavily contaminated by movement. The standard deviation is also very big. After ICA, the estimated components are shown in Fig.4-9. Evidently, the swing noise can be rejected with its biggest standard deviation.

## 4.3 Testing

### 4.3.1 Testing of Artificial Mixed Data

We have tested various waveforms to prove the performance of ICA. First of all, take Fig.4-10 for example, this is of course not realistic recorded signals, but

sufficient for this example. The original signal is shown in Fig.4-10 while the mixed signal is shown in Fig.4-11. Figure 4-12 gives the two signals estimated by fast ICA. Figure 4-13 shows the signals estimated by Information maximization ICA. As can be seen, these are very close to the original source signals.

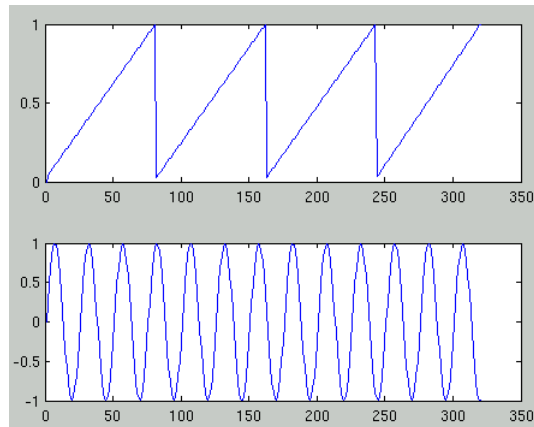


Fig.4- 10The original signals

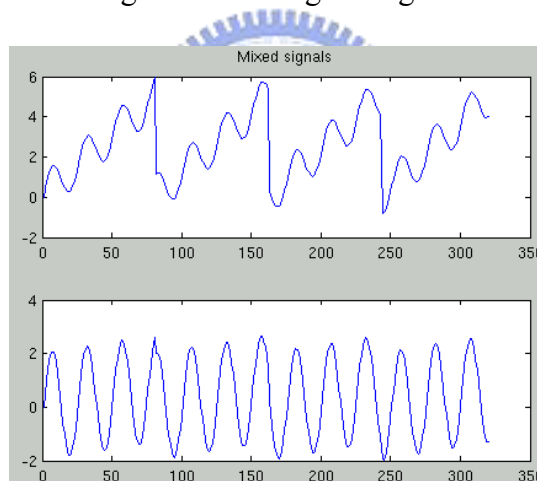


Fig.4- 11The observed mixtures of the source signals in Fig.4-10

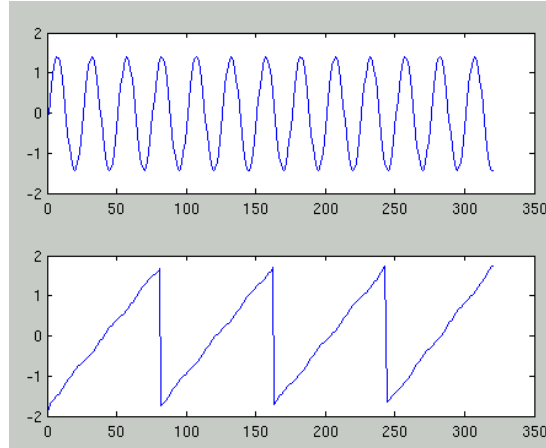


Fig.4- 12 The result of fast ICA, estimated using only the observed signals in Fig.4-11

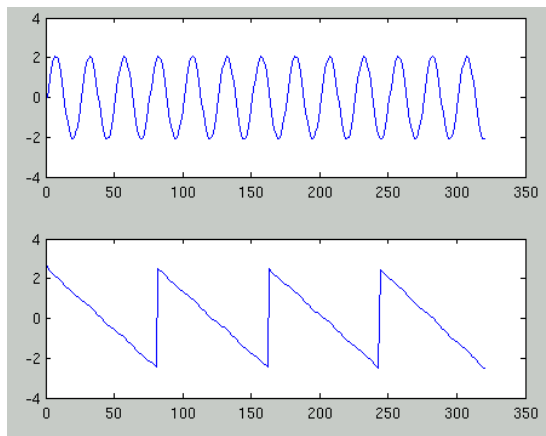


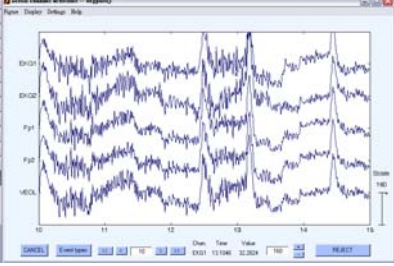
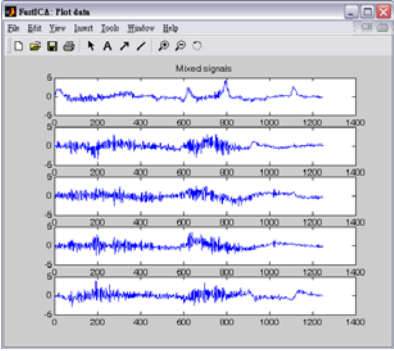
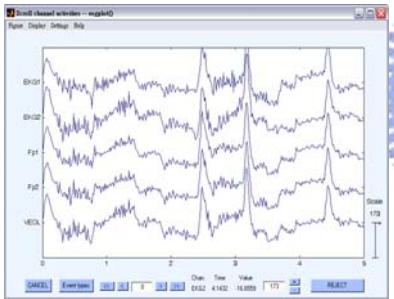
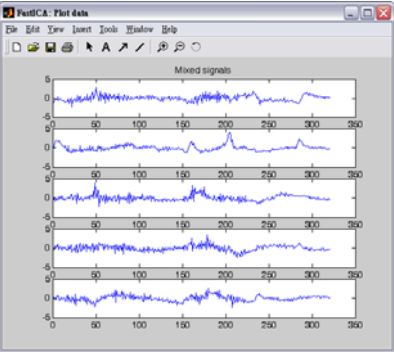
Fig.4- 13 The result of Information Maximization ICA, estimated using only the observed signals in Fig.4-11

### 4.3.2 Testing of Real EEG Data

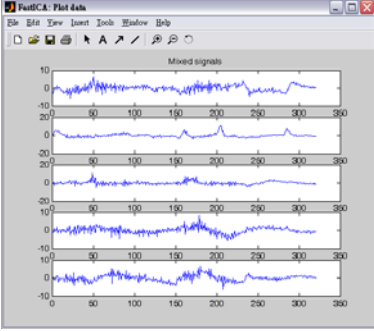
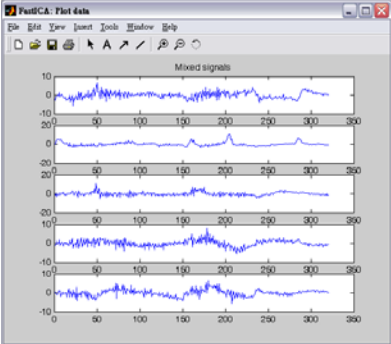
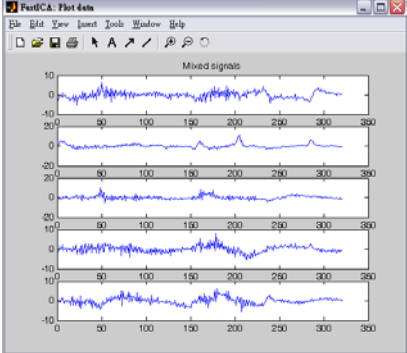
A set of ICA component after ICA is listed in the flowing table. With the modifying of ICA variables, we successfully speed up ICA. And the result shows good components after modified ICA.

Table 4- 1 The result of each ICA with different variables

Figure	Attributes	Execution time

 <p>The 320- point Raw EEG Data</p>	<p>Sampling rate: 250Hz</p>	<p>N/A</p>
 <p>The Estimated ICA Components</p>	<p>Sampling rate: 250Hz</p>	<p>142.04s</p>
 <p>The 320-point Raw EEG Data</p>	<p>Sampling rate: 64Hz</p>	<p>N/A</p>
 <p>The Estimated ICA Components</p>	<p><b>Extended</b>  Initial learning rate: 0.001  Block size: 10  Anneal step :0.98  Anneal deg : 60  Max steps: 512</p>	<p>43.22s</p>



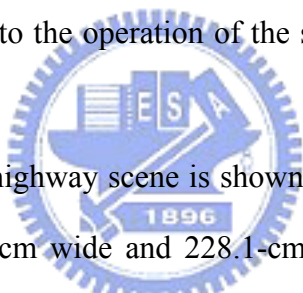
 <p>The Estimated ICA Component</p>	<p>Not extended</p> <p>Initial learning rate: 0.001</p> <p>Block size: 10</p> <p>Anneal step :0.98</p> <p>Anneal deg : 60</p> <p>Max steps: 512</p>	<p>27.61s</p>
 <p>The Estimated ICA Component</p>	<p>Not extended</p> <p>Initial learning rate: 0.001</p> <p>Block size: 29</p> <p>Anneal step :0.90</p> <p>Anneal deg : 60</p> <p>Max steps: 512</p> <p><b>Momentum: 0.4</b></p> <p>Verbose: off</p>	<p>12.51s</p>
 <p>The Estimated ICA Component</p>	<p>Not extended</p> <p>Initial learning rate: 0.001</p> <p>Block size: 29</p> <p>Anneal step :0.90</p> <p>Anneal deg : 60</p> <p>Max steps: 512</p> <p>Momentum: 0.4</p> <p>Verbose: on</p> <p>Stop at wchange less than 0.001</p>	<p>8.25s</p>

## Chapter 5 Experiment Designs and Results

### 5.1 The Experiment Design

#### 5.1.1 Experimental Setup

For the purpose of safety, a VR-based dynamic driving environment was developed for the drowsiness experiments to mimic realistic highway. The developed VR dynamical simulation system mainly consists of three elements: (1) a six-degree-of-freedom motion platform, (2) a real car, and (3) an interactive VR scene. The subjects are asked to sit inside the car on the platform with their hands holding the steering wheel to control the car in the VR scene. Seven projectors are used in the experiment to construct a 360-degree 3D scene as shown in Fig. 5-1. The movements of the platform are according to the operation of the subject and the condition of the road surface.



The VR-based four-lane highway scene is shown in Fig. 5-2. It is projected on a 120°-surround screen (304.1-cm wide and 228.1-cm high), which is 350 cm away from the driving cabin. The four lanes from left to right are separated by a median stripe. The distance from the left side to the right side of the road is equally divided into 256 points (digitized into values 0–255), where the width of each lane and the car is 60 and 32 units, respectively. The refresh rate of highway scene was set properly to emulate a car driving at a fixed speed of 100 km/hr on the highway. The car is randomly drifted (triggered from the WorldToolKit (WTK) program and the on-set time is recorded) away from the center of the cruising lane to mimic the consequences of a non-ideal road surface. The subject's driving error is defined as the deviation between the center of the vehicle and the center of the cruising (third) lane and it was continuously and simultaneously measured by the WTK program and recorded in the physiological measurement system accompanying with EEG/EOG/ECG physiological

signals.

It has been shown that human fatigue or drowsiness most commonly occurs late at night and during the afternoon. During these periods, alertness deficits would most likely take place in 1-h monotonous working [32], [33]. Thus, we conducted all driving experiments in the early afternoon hours after lunch to maximize the opportunities to collect data during which subject driving performance became intermittent. All the subjects were instructed to keep the car at the center of the cruising lane by controlling the steering wheel. For each session, the subject started with a min calibration procedure and then was asked to drive the car continuously for 45 min. The EEG/EOG/ECG data and the driving errors were measured and recorded simultaneously. We had collected successfully EEG data of 5 subjects (ages from 20 to 35 yr) participated in the VR-based driving task. We select participants who had two or more micro-sleep episodes based on the measured response time which is shown in Fig.5-3 and confirmed by video recordings in at least two driving sessions for further analysis. Based on these criteria, five subjects were selected for further on-line algorithm testing and verification.



Fig.5-1 A 360-degree 3D VR dynamic driving environment.



Fig.5- 2 The VR-based four-lane highway scene

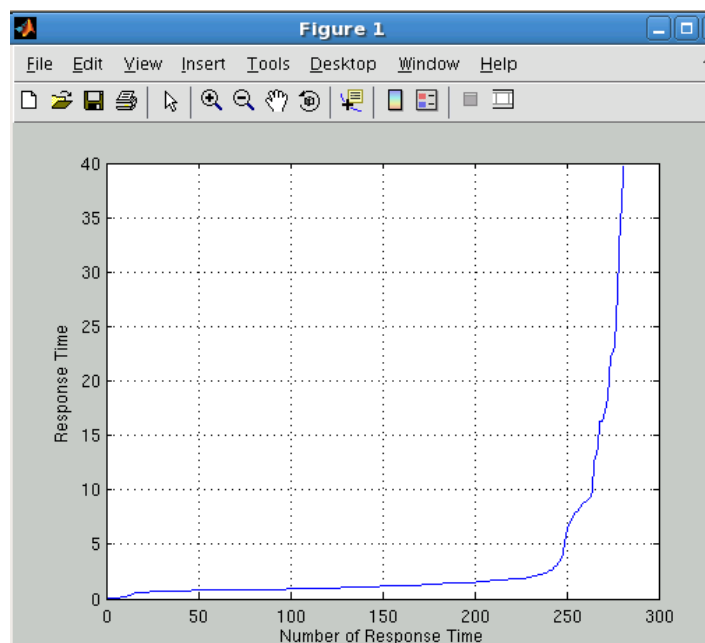


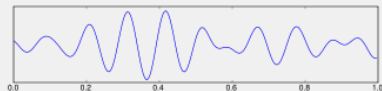
Fig.5- 3 The Response Time in One Session

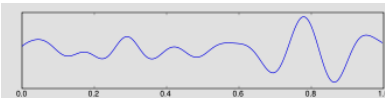
## 5.2 Drowsiness Detection Algorithm

There are four major types of continuous rhythmic sinusoidal EEG activity. They are recognized as alpha (8-12Hz), beta (above 12Hz), delta (below 4Hz) and theta (4-8Hz) and are listed in Table 5-1:

Table 5- 1 Characteristics of EEG bands

Types	Band range	Description

<p><math>\gamma</math> (gamma)</p>	<p>30Hz~</p> 	<p>Gamma rhythms may be involved in higher mental activity, including perception, problem solving, fear, and consciousness.</p>
<p><math>\beta</math> (beta)</p>	<p>13 – 30Hz</p> 	<p>Beta with low amplitude beta with multiple and varying frequencies is often associated with active, busy or anxious thinking and active concentration. Rhythmic beta with a dominant set of frequencies is associated with various pathologies and drug effects, especially benzodiazepines.</p>
<p><math>\alpha</math> (alpha)</p>	<p>8 – 12 Hz</p> 	<p>Alpha is characteristic of a relaxed, alert state of consciousness. For alpha rhythms to arise, usually the eyes need to be closed. Alpha attenuates with drowsiness and open eyes, and typically come from the occipital (visual) cortex. An alpha-like normal variant called mu is sometimes seen over the motor cortex (central scalp) and attenuates with movement, or rather with the intention to move.</p>
<p><math>\theta</math></p>	<p>4 - 8 Hz</p>	<p>Theta is associated with drowsiness,</p>

<b>(theta)</b>		<p>childhood, adolescence and young adulthood. This EEG frequency can sometimes be produced by hyperventilation. Theta waves can be seen during hypnagogic states such as trances, hypnosis, deep day dreams, lucid dreaming and light sleep and the preconscious state just upon waking, and just before falling asleep.</p>
<b>δ (delta)</b>	~ 4 Hz	<p>Delta is often associated with the very young and certain encephalopathies and underlying lesions. It is seen in stage 3 and 4 sleep.</p>

As the characteristic of EEG activity described above, there is an important phenomenon found by the team of brain research center (BRC), NCTU (National Chao Tung University) while recording EEG from forehead. That is, if a person is mild drowsiness, the alpha wave will tend to be superior in EEG activity, and its power will increase time after time. After that, if the person tends to fall asleep, the theta wave will tend to be superior in EEG activity, and the power of alpha will decrease while the theta will still increase time after time. With the phenomenon, the algorithm can be mapped in to a table listed below.

Table 5- 2 Criterion of Drowsiness

<b>State of EEG band</b>	<b>State of consciousness</b>
<b>alpha ↑ theta↑</b>	<b>Mild drowsiness</b>

<b>alpha ↓ theta↑</b>	<b>Deep drowsiness</b>
<b>alpha ↑ theta↓</b>	<b>Conscious</b>
<b>alpha ↓ theta↓</b>	<b>Conscious</b>

For the criterion of drowsiness is based on the changes of EEG activity in the frequency domain. We use the short-time Fourier transform (STFT), or alternatively short-term Fourier transform, which is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. Here we have tested the short-time Fourier transform on the embedded processor we use. It has produced good results while processing both lower frequency sinusoid (theta band from 4 to 8 Hz) and higher frequency sinusoid (alpha band from 8 to 12 Hz) of the frequency band we desired(4 to 12 Hz). As shown in Fig.5-4, the power of 4Hz and 5Hz sinusoid showed in Fig.5-5 result in peak of 4Hz and 5Hz. Also, in Fig.5-6, the power of 10Hz and 12Hz sinusoid showed in Fig.5-7 result in peak of 10Hz and 12Hz.



Fig.5- 4 The original signal of 4Hz & 5Hz sinusoid

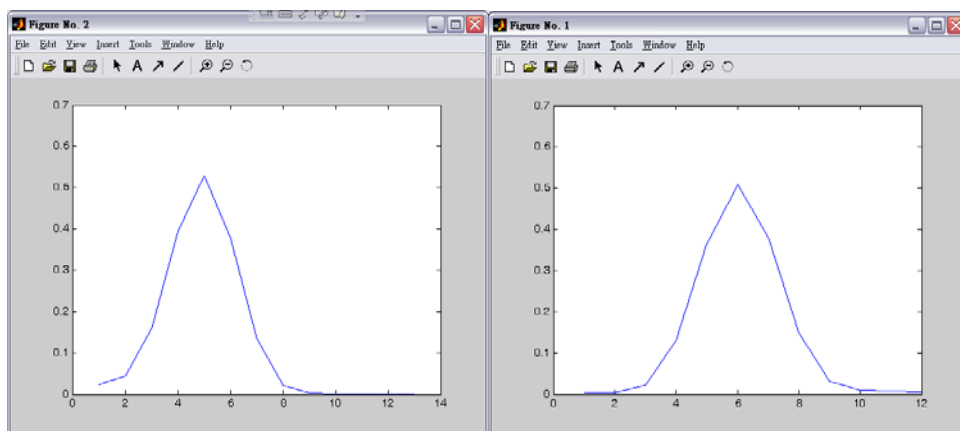


Fig.5- 5 The power spectrum of 4Hz & 5Hz sinusoid



Fig.5- 6 The original signal of 10Hz & 12Hz sinusoid

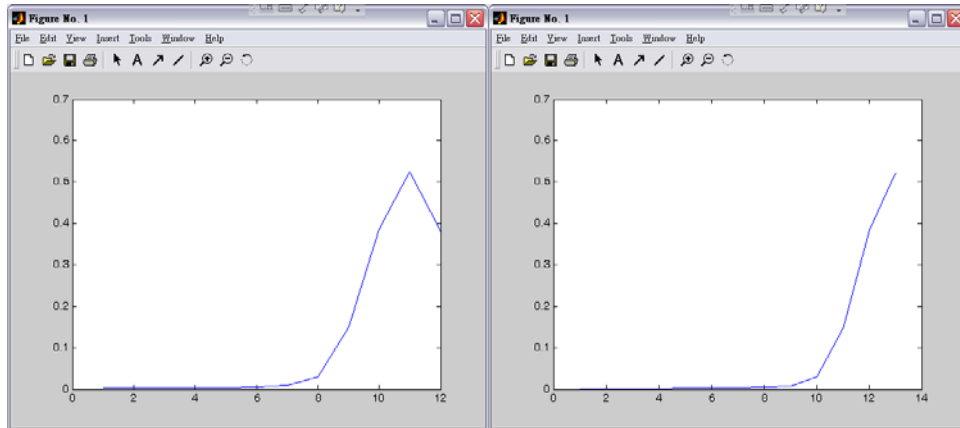


Fig.5- 7 The power spectrum of 10Hz & 12Hz sinusoid

The flowchart of our algorithm is described in Fig.5-8. First, the 4-channel EEG data is re-sampled to sampling rate 64Hz. After gathering 512 points of EEG data, the data was fed into ICA. After that, some estimated components are rejected with the criterion of standard deviation described before. We select the last 192 points of ICA component from rest components and use FFT to estimate the power of EEG data. Finally with the online drowsiness detection algorithm, we recognized the drowsiness index. The detail of the FFT procedure is shown in Fig.5-9. The 192-point EEG data of each channel is overlapped with 64-point update for each sec. Then the 192 points are divided into 32 points sub-windows with overlap 24 points. Then The first 16 points and the last 16 points of 64-point FFT were padded with zero.



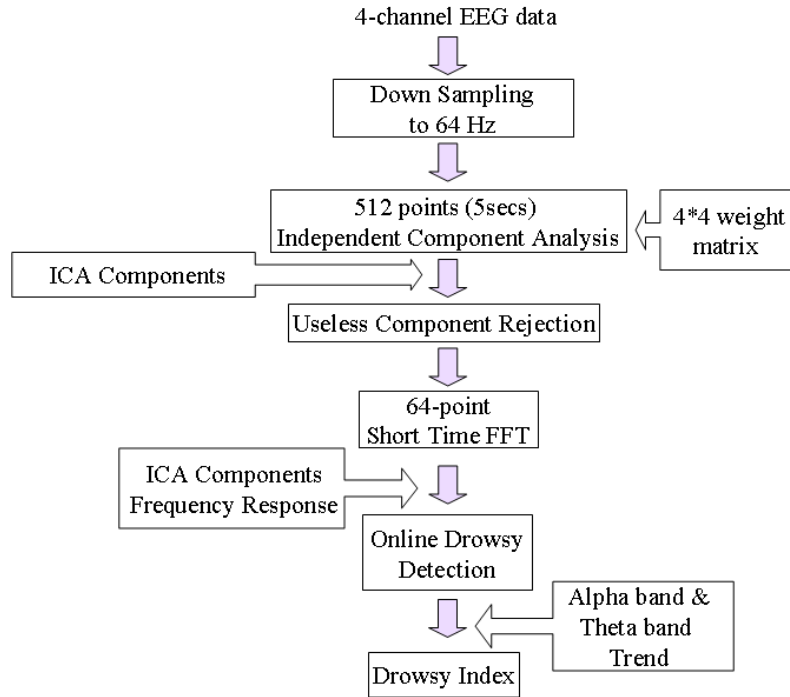


Fig.5- 8 The flowchart of our algorithm

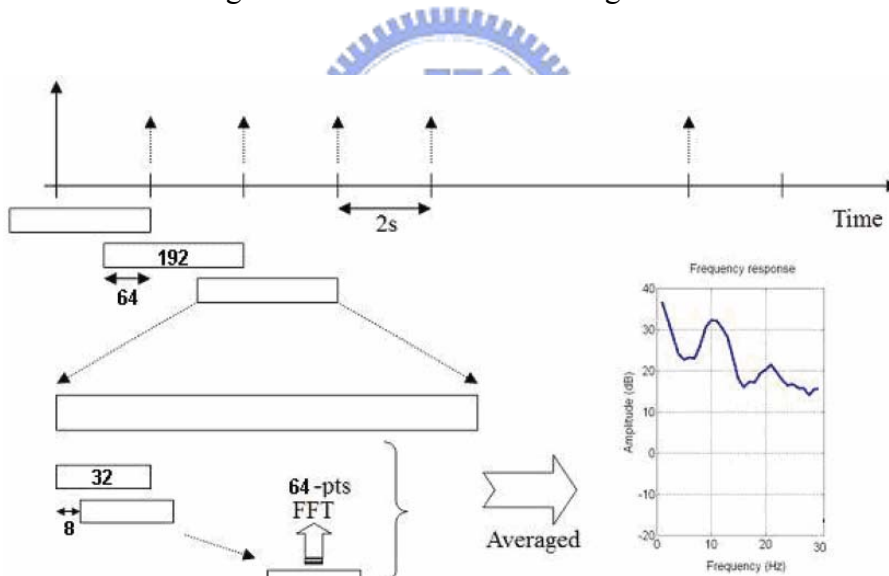


Fig.5- 9 The procedure of FFT

After we take 1~13 Hz of the decibel value of FFT and find the trend of alpha and theta band power, we can estimate drowsiness index with a 20 seconds time window. The flow of from FFT to estimation of drowsiness index is shown in Fig.5-10. The moving window size we use is 20sec. Since each time we update 2 seconds. We calculate the whole slope of ten 2-sec power when each time we update

the alpha and theta power value. Choosing the moving window of 20 seconds is the middle-of-the-road policy here because whether the window is too big or too small isn't good for drowsiness index estimation. If the window is too big, the change of drowsiness index will be too slow. On the contrary, if the window is too small, the change of drowsiness index will be too soon.

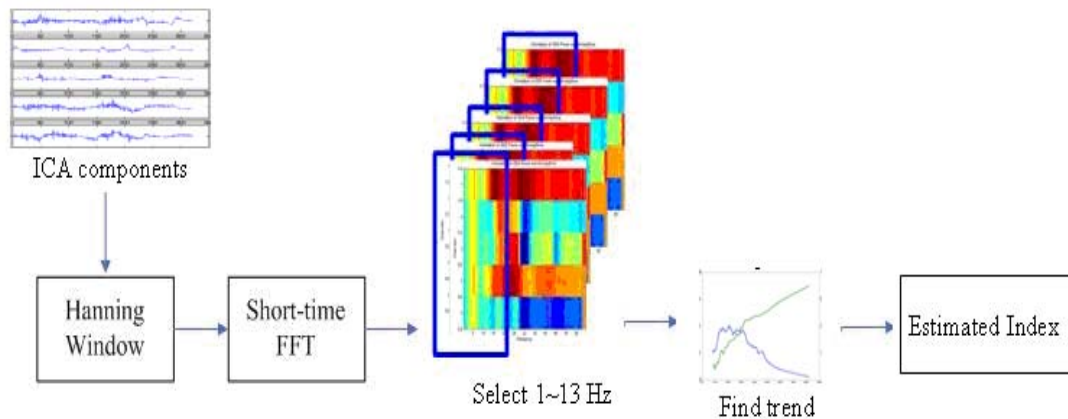


Fig.5- 10 The flow of drowsiness estimation algorithm

## 5.3 The Results

### 5.3.1 Result of 4 Channel Online EEG Analysis System

A demonstration of our system is shown in Fig.5-11. First, the preprocessing unit records the 4-channel EEG and sends wirelessly raw digital EEG to processing unit. Then after the drowsiness index is estimated, it sends result to a display unit through internet. The display unit could be PC or laptop and the displayed stuff is shown in Fig.5-12. The preprocessing can also be placed on to a hat as shown in Fig.5-13. This is convenient for users. Figure 5-14 shows the developed wireless BCI using RF. Figure 5-15 shows the developed wireless BCI using Bluetooth.

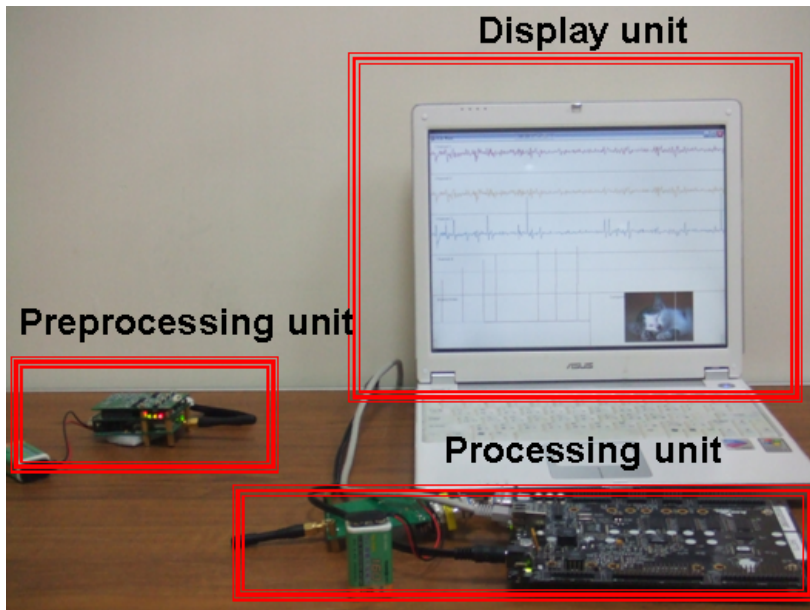


Fig.5- 11A Demonstration of Our System

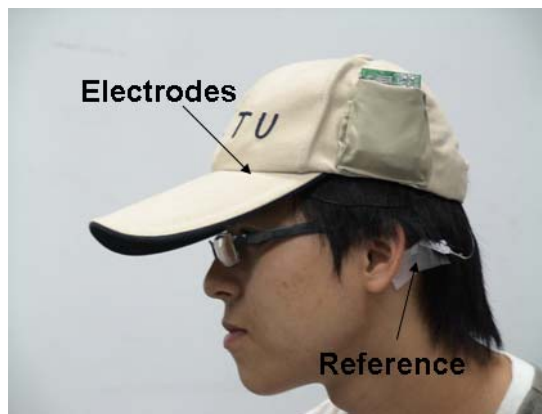


Fig.5- 12 A Demonstration of EEG Hat

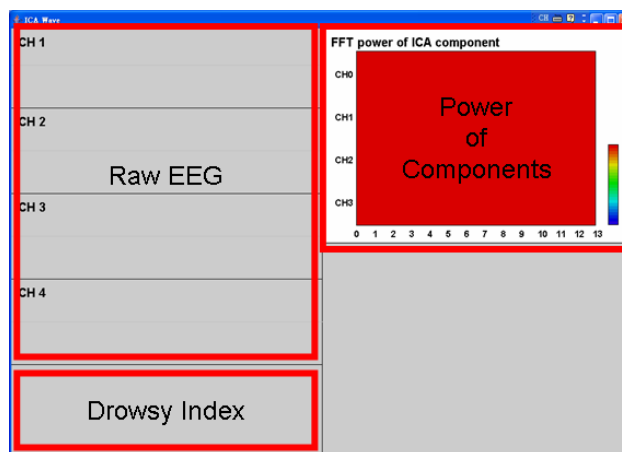


Fig.5- 13 The Result of GUI



Fig.5- 14 wireless BCI using RF



Fig.5- 15 wireless BCI using Bluetooth



## Chapter 6 Conclusions and Future Works

We propose an ICA-based embedded wireless BCI (Brain Computer Interface) system for real-time drowsiness detection to improve the imperfections of offline analysis. The embedded wireless BCI has been implemented. It includes three functional blocks: wireless transmission and EEG recording, real-time transmission online ICA, and online drowsiness detection algorithm. This BCI system achieves two major targets. One is that the wireless overcomes the problem of long distance and inconvenient of connection lines; another is that the modified online ICA and drowsiness detection algorithms are implemented on the embedded DSP board to give warning to drivers in real-time.

After that, the performance of the embedded wireless BCI system can still be improved in time, transmission and the problem of algorithm stability as follows.

- ◆ The hardware of Bf533 is provided with hardware interrupt and DMA (Direct Memory Access), this speed the time handing peripheral.
- ◆ The transmission protocol can be replaced with high speed Bluetooth.
- ◆ The selection of components now is artificial. The rejection of estimated components is not completed correct. Most important of all is the selection of components.

## References

- [1] M. Falkenstein, J. Hohnsbein, J. Hoormann, and L. Blanke, “Effects of crossmodal divided attention on late ERP components. II. Error processing in choice reaction tasks,” *Electroencephalography and Clinical Neurophysiology*, vol. 78, pp. 447-55, 1991.
- [2] S. Makeig, M. Westerfield, T.P. Jung, S. Enghoff, J. Townsend, E. Courchesne, and T. J. Sejnowski, “Electroencephalographic Sources of Visual Evoked Responses,” *Science*, vol. 295, pp. 690-94, Jan. 25, 2002.
- [3] M.D. Serruya, N.G. Hatsopoulos, L. Paninski, M.R. Fellows, J.P. Donoghue, “Instant neural control of a movement signal,” *Nature*, vol. 416, pp. 141–142, 2002.
- [4] D.M. Taylor, S.I. Tillery, A.B. Schwartz, “Direct cortical control of 3D neuroprosthetic devices,” *Science*, vol. 296, pp. 1829–1832, 2002.
- [5] J.K. Chapin, K.A. Moxon, R.S. Markowitz, M.A.L. Nicolelis, “Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex,” *Nature Neuroscience*, vol. 2, pp. 664–670, 1999.
- [6] J. Wessberg, C.R. Stambaugh, J.D. Kralik, P.D. Beck, M. Laubach, “Realtime prediction of hand trajectory by ensembles of cortical neurons in primates,” *Nature*, vol. 408, pp. 361–365, 2000.
- [7] 連怡仲, “數位腦波機系統之設計與研製”, 國立台灣大學, 碩士論文, 民國八十六年。
- [8] 杜益昌, “多通道腦波機系統之設計與裝置”, 國立台灣大學, 碩士論文, 民國八十七年。
- [9] 林伯星, “數位腦波機系統之人機介面設計與研發”, 國立台灣大學, 碩士論文, 民國八十七年。

- [10] 蔡進寶，“腦波紀錄儀設計與製作”，國立交通大學，碩士論文，民國八十九年。
- [11] 楊勝文，“可攜式智慧型腦波記錄器”，國立台灣大學，碩士論文，民國九十年。
- [12] 黃名斌，“USB 介面之模組化腦波記錄儀”，國立中原大學，碩士論文，民國九十一年。
- [13] 吳炎法，“研製無線化多通道腦波系統並運NAB 方法進行腦波分類之研究”，國立台北科技大學，碩士論文，民國九十二年。
- [14] M. Cheng, X. Gao, S.K. Gao, and D.F. Xu, “Design and Implementation of a Brain-Computer Interface With High Transfer Rates,” *IEEE Transaction biomedical Engineering*, vol. 49, pp. 1181-1186, 2002.
- [15] X.R, Gao, D.F. Xu, M. Cheng, and S.K. Gao, “A BCI-Based Environmental Controller for the Motion-Disabled,” *IEEE Transaction Neural Systems and Rehabilitation Engineering*, vol. 11, pp. 137-140, June 2003.
- [16] B. Obermaier, “Design and implementation of an EEG based virtual keyboard using hidden Markov models”, *Ph.D. dissertation*, Technical University-Graz, Graz, Austria, 2001.
- [17] B. Obermaier, C. Neuper, and G. Pfurtscheller, “Information transfer rate in a 5 classes brain-computer interface,” *IEEE Transaction Rehabilitation Engineering*, vol. 9, pp. 283-288, 2001.
- [18] G. Edlinger, G. Krausz, F. Laundl, I. Niedermayer, C. Guger, “Architectures of Laboratory-PC and Mobile Pocket PC Brain-Computer Interfaces,” *Proceedings of the 2 International IEEE EMBS Conference on Neural Engineering*, Arlington, Virginia, March 16 - 19, 2005.
- [19] K. Ashwin Whitchurch, B. Han Ashok, R. Vinod Kumaar and K. Sarukesi, and K. Vijay Varadan, “Wireless system for long term EEG monitoring of Absence

Epilepsy,” *Biomedical Applications of Micro- and Nanoengineering, Proceedings of SPIE*, Vol. 4937, pp. 343-349, November 2002.

[20] Y.H. Nam, Z. Halm, Y.J. Chee, K.S. Park, “Development of Remote Diagnosis System Integrating Digital Telemetry for Medicine,” *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Hong Kong, China, October 29-November 1, 1998.

[21] W.C. Kao, W.H. Chen, C.K. Yu, C.M. Hong, S.Y. Lin, “A real-time system for portable homecare applications,” *Proceedings of IEEE International Symposium Consumer Electronics*, Macau, June 14-16, 2005.

[22] F. Lamberti, C. Demartini, “Low-Cost Home Monitoring Using a Java-Based Embedded Computer,” *Proceedings of the 4th Annual IEEE Conference on Information Technology Applications in Biomedicine*, UK, April 24-26, 2003.

[23] T.P. Jung, S. Makeig, C. Humphries, T.W. Lee, M.J. McKeown, V. Iragui, and T.J. Sejnowski, “Removing Electroencephalographic Artifacts by Blind Source Separation,” *Psychophysiology*, vol. 37, pp. 163-178, 2000.

[24] T.P. Jung, S. Makeig, W. Westerfield, J. Townsend, E. Courchesne, and T.J. Sejnowski, “Removal of eye activity artifacts from visual event-related potentials in normal and clinical subjects,” *Clinical Neurophysiology*, vol. 111, pp. 1745-1758, 2000.

[25] S. Ulrika, “Blink behaviour based drowsiness detection: method development and validation”, *Undergraduate dissertation*, Linköping University, Department of Biomedical Engineering, 2004.

[26] B. Otman Adam, B. Jean Pierre, K. Fakhreddine, D. Kristopher, “Drowsiness detection system”, *Intelligent Mechatronic Systems Incorporated*, 2004.

[27] S. Matsushita, A. Shiba, K. Nagashima, “A wearable fatigue monitoring system: application of human-computer interaction evaluation”, *Proceedings of*



*the 7th Australasian User interface conference*, Hobart, Australia, 2006.

- [28] C.T. Lin, R.C. Wu, S.F. Liang, W.H. Chao, Y.J. Chen, and T.P. Jung, “EEG-Based Drowsiness Estimation for Safety Driving Using Independent Component Analysis”, *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. 52, pp. 2726-2738, December, 2005.
- [29] C.T. Lin, L.W. Ko, I.F. Chung, T.Y. Huang, Y.C. Chen, T.P. Jung, and S.F. Liang, “Adaptive EEG-Based Alertness Estimation System by Using ICA-Based Fuzzy Neural Networks”, *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. 53, pp. 2469-76, November, 2006.
- [30] M. Jose. Carmena, A. Mikhail Lebedev, E. Roy. Crist, E. Joseph O'Doherty, M. David Santucci, F. Dragan Dimitrov, G. Parag Patil, S. Craig Henriquez, A. L. Miguel Nicolelis, “Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates,” *PLoS Biology*, vol. 1, pp. 193-208, 2003
- [31] J. Hendrix, “Fatal crash rates for tractor-trailers by time of day,” *Proceedings of the International Truck and Bus Safety Research and Policy Symposium*, Tennessee, USA, 2002
- [32] H. Ueno, M. Kaneda, and M. Tsukino, “Development of drowsiness detection system,” *Proceedings of Vehicle Navigation and Information Systems Conference*, Yokohama, Japan, September, 1994.
- [33] 黃冠智, “即時 DSP 嵌入式 Linux 系統開發及其在腦波訊號處理之應用”, 國立交通大學, 碩士論文, 民國九十三年
- [34] 高士政, “以雙核心處理器為基礎之即時嵌入式無線腦波訊號感測與處理系統及其在駕駛員瞌睡偵測之應用”, 國立交通大學, 碩士論文, 民國九十四年。
- [35] 謝弘義, “具備多工排程功能之無線嵌入式腦機介面系統及其在即時汽車駕駛員疲勞狀態偵測與提醒之應用”, 國立交通大學, 碩士論文, 民國九十四

年。

- [36] Ancher technology inc., Wireless UART RF3100 module specification. Ancher technology inc., Taipei County Taiwan R.O.C. [Online]. Available: <http://www.ancher.com.tw/rf3100.html>
- [37] Ancher technology inc., Wireless UART RF3105 module specification. Ancher technology inc., Taipei County Taiwan R.O.C. [Online]. Available: <http://www.ancher.com.tw/rf3105.html>
- [38] Q. Li, C. Yao, “Real-Time Concepts for Embedded Systems,” *CMP Books*, 2003.
- [39] Getting Started With Blackfin Processors Rev2.0, Analog Devices, Inc. Sep 2005.
- [40] ADSP-BF533 Data Sheet Rev. D, Analog Devices, Inc., Sep 2006.
- [41] ADSP-BF533 STAMP v1.2, Analog Devices, Inc.
- [42] Blackfin Linux Docs, Open Source Software on the Blackfin Processor, Blackfin Linux Docs. [Online]. Available: <http://docs.blackfin.uclinux.org/doku.php?id>
- [43] Linux™ Blackfin, Welcome to the Blackfin Linux Project, Linux™ Blackfin. [Online]. Available: <http://blackfin.uclinux.org/>
- [44] ICA, Independent component analysis from Wikipedia, the free encyclopedia, Independent component analysis, Wikipedia. [Online]. Available: [http://en.wikipedia.org/wiki/Independent\\_component\\_analysis](http://en.wikipedia.org/wiki/Independent_component_analysis)
- [45] A. Hyvärinen and E. Oja, “Independent Component Analysis: A Tutorial,” *Neural Networks*, 13(4-5):411-430, 2000.
- [46] Lookup table, Lookup table from Wikipedia, the free encyclopedia, Lookup table, Wikipedia. [Online]. Available: [http://en.wikipedia.org/wiki/Lookup\\_table](http://en.wikipedia.org/wiki/Lookup_table)
- [47] Loop unwinding, Loop unwinding from Wikipedia, the free encyclopedia, Loop unwinding, Wikipedia. [Online]. Available: [http://en.wikipedia.org/wiki/Loop\\_unrolling](http://en.wikipedia.org/wiki/Loop_unrolling)

- [48] C. Guger, W. Harkam, C. Hertnaes, G. Pfurtscheller, "Prosthetic Control by an EEG-based Brain-Computer Interface (BCI)", *Proceedings of AAATE 5th European Conference for the Advancement of Assistive Technology*. Düsseldorf, Germany, November 3-6, 1999.
- [49] Negentropy, Negentropy from Wikipedia, the free encyclopedia, Negentropy, Wikipedia. [Online]. Available: <http://en.wikipedia.org/wiki/Negentropy>
- [50] A. J. Bell and T. J. Sejnowski, "An information maximization approach to blind separation and blind deconvolution," *Neural Computation*, Vol. 7, pp. 1129-1159, 1995.
- [51] J.F. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, Vol. 11, pp. 157-192, 1999.
- [52] A. Hyvärinen, "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis" *IEEE Transactions on Neural Networks*, Vol. 10, pp. 626-634, 1999.
- [53] C. Jutten, J. Herault, "Blind separation of sources, Part 1: an adaptive algorithm based on neuromimetic architecture," *Signal Processing*, Vol. 24, pp.1-10, 1991.
- [54] P. Comon, C. Jutten, J. Herault, "Blind separation of sources, Part 2: problems statement," *Signal Processing*, Vol. 24, pp.11-20, 1991.
- [55] R. Linsker, "Improved local learning rule for information maximization and related applications," *Neural Networks*, Vol. 18, pp.261-265, 2005.
- [56] S. Amari, T.P. Chen, A. Cichocki, "Stability analysis of learning algorithms for blind source separation," *Neural Networks*, Vol.10, pp.1345-1351, 1997.
- [57] T.W. Lee, M. Girolami, T. J. Sejnowski, "Independent Component Analysis using an Extended Infomax Algorithm for Mixed Sub-Gaussian and Super-Gaussian Sources," *Neural Computation*, Vol. 11, pp. 417-441, 1999.