國立臺灣大學工學院機械工程學系

碩士論文

Department of Mechanical Engineering
College of Engineering
National Taiwan University

Master Thesis

可配合眼球運動之人工眼伺服系統

Eye Motion Interpretation and Artificial Eye Control

陳薏竹

Yi-Zhu Chen

指導教授：顏家鈺 博士

Advisor: Jia-Yush Yen, Ph.D.

中華民國 99 年 6 月

June, 2010


國立臺灣大學
人愛國愛・學勵品教

國立臺灣大學碩士學位論文
口試委員會審定書

可配合眼球運動之人工眼伺服系統

Eye Motion Interpretation and

Artificial Eye Control

本論文係陳薏竹君（R97522811）在國立臺灣大學機械工程學研究所完成之碩士學位論文，於民國 99 年 6 月 30 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

（指導教授）

系主任、所長


國立臺灣大學
人愛國愛・學勵品教

致謝

首先我要感謝顏家鈺老師這兩年來對於研究的內容以及做研究的方法所做的建議與指導，老師幽默的個性以及開放的態度，讓實驗室的大家能夠在一個愉快的研究環境下互相成長，另外也要謝謝口試委員陳文翔醫師、王一中醫師以及戴子安副教授在計畫執行一年多來對於研究的建議與協助，對於這本論文的指導。

我要特別感謝 tito 學長，除了在研究上的建議與協助外還有對我的包容與勉勵，從他身上學習到許多我欠缺的特質與觀念，感謝俊儒學長在影像處理以及許多方面的幫忙，由衷的感謝並祝福 tito 和俊儒學長未來的研究路途一切順利。

感謝實驗室各位學長姐的協助與陪伴，謝謝熊、俊賢、政儒、郭老大、阿良、俊彥、俞婷學姐、玄宗、雅琴、凡哲、晨瑋、承錫、思龍、小倩、念真以及志堅，給予我們許多的建議以及鼓勵。

還要謝謝一起奮鬥兩年的好夥伴們：游泳、柏儒、Pablo、一折還有汪大師，一起為論文奮鬥、一起出遊，度過了許多歡樂的時光。感謝 Jimmy 在論文上的幫忙，還要其他碩一的學弟妹們：左庭、紫微、中原、庭軒，帶給實驗室一股新的朝氣。

最後謝謝我的家人，一直給予包容與鼓勵並陪伴在身邊，讓我在沮喪時有勇氣重新出發。

感謝大家，並祝大家都能完成心目中對自己的期盼。


國立臺灣大學
人愛國愛・學勵品教

可配合眼球運動之人工眼伺服系統

陳薏竹

國立台灣大學機械工程研究所

摘要

眼球追蹤方面的研究已經發展許多年，除了持續應用在科學研究上，例如討論人眼對視覺刺激的傾向或眼球注視方向反應出的心理狀態等，也積極開發其應用層面，比如做為電腦周邊的輸入裝置或記錄眼睛所注視的影像。市面上目前已發展出多種的眼球追蹤器供實驗研究或各種應用。然而，多數眼球追蹤裝置都有相同的缺點：繁複的校準過程以及準確度容易受頭部轉動影響，以因此至今尚未能夠廣泛應用。

在本論文中，作者設計並實現了一套眼球追蹤裝置系統，能夠將攝影機隨眼球轉動並聚焦在同一物體上，期許未來能夠應用在控制人工眼球的方向以及對焦，此系統使用了一組伺服馬達做為制動器、一組微處理器做為控制器，並選擇一台攝影機做為擷取眼睛影像的輸入裝置，經過影像處理的運算後能夠將系統的兩台分別代表人眼以及未來人工眼球的攝影機聚焦在眼睛所視的同一點上，並經由幾何運算可得到眼睛所視物體相對於人眼之方位以及距離。實驗結果顯示，能夠依據所擷取到的眼球影像成功驅動攝影機轉向人眼注視之物體，又所估測的物體距離與實際距離誤差約在 10%以下。

關鍵字：眼球追蹤、人工眼、物件辨識


國立臺灣大學
愛國愛學勵品教人

Eye Motion Interpretation and Artificial Eye Control

Yi-Zhu Chen
Department of Mechanical Engineering
National Taiwan University

Abstract

Research in eye tracking technique has been developed for many years. Besides being continuously used in scientific experiments, such as discussing the tendency of human eye for visual stimulation or psychology state revealed from eye gaze direction, it's application aspects are also being actively developed. Some of its applications include: alternative input devices in human-computer interface or recording of the eye's gazed object. There are many commercially available developed eye tracking devices for scientific experiments or other applications. However, most of the eye tracking devices have the same defects: complicated calibration procedures and the accuracy is easily affected by head motion. Therefore, it hasn't been widely used up to now.

In this thesis, we have designed and developed an eye tracking system consisting of cameras that follow the eye's movement and focus on the same target. It is expected to be applied in an artificial eye system controlled by a real eye. The system uses servo motors as actuators, microcontrollers as controllers and a camera to capture the eye's image. After this image is processed, the two cameras, each represents the real human eye and the artificial eye, are able to focus on a specific point, the same as the human eye's gazing point. With some geometrical operations, the object's distance

and direction away from human eye is also obtained. As shown in experimental results, the camera representing the artificial eye can be successfully turned to the eye's gazed object according to the captured eye image and the estimated object's distance has an error of about 10%.

Keywords: eye tracking, artificial eye, object recognition

Table of Contents


國立臺灣大學
博學篤行・愛國愛人

List of Figures

List of Tables


國立臺灣大學
人愛國愛學勵品教

Chapter 1 Introduction

1.1 Research Motivation and Purpose

Human organs are quiet complicated and generally difficult to repair once the damage is done. The human eye is one of the most complex organs. Its fine structure includes cornea, retina, lens, iris and pupil. Each part is absolutely essential for each particular function. Even the most precise camera cannot accurately simulate the operation of the human eye. This project is meant to produce an artificial eye that can replace the damaged eye. To further understand the eye motion's principle and be able to turn the artificial eye to the same direction of the real eye and be able to focus on the same target as the real eye, an eye tracking system is designed.

In this thesis, it is desired to move two cameras which represent the good human eye and the artificial eye, respectively, and are controlled by the good eye's movement. In order to meet this demand, a structure with three cameras was designed. It is able to react rapidly to the eye's movement. Also, with some computation using the turning angles of the cameras, the focus distance can be determined. If the tracking system is capable of working in a highly efficient way, besides from being used as an artificial eye, it can have a wide range of applications.

1.2 Prior Art (Paper Survey)

Eye gaze tracking technique has been developed for a long time. Early eye gaze tracking devices were used in scientific experiments in controlled environments. Though, it was developed towards a human-computer interface for many years. To be a general interactive for general users, its intrusive technique for eye gaze tracking is not acceptable anymore and its usability has limited by two major problems which are

the troublesome calibration procedure and restrictions of head motion. Fortunately, in recent years, several techniques have been developed for remote eye gaze tracking with simple calibration and free head motion.

Non-intrusive techniques for eye gaze tracking are mostly vision based by using camera to capture the image of the eye with other optical devices to measure the eye position. After surveyed several papers, many eye trackers use infrared (IR) light source to enhance the contrast between pupil and iris in order to segment the pupil, just like Fig. 1-1. Because IR is not visible by human eye, the light doesn't distract the user but still can be detected by cameras. The accuracy of gaze detection depends on the pupil detection. Besides the detection of pupil position, there are several measurements used in other techniques to estimate the direction of gaze, such as the skin potential [1], relative position of Purkinje images [2] (i.e. reflections created at different layers of the eye structure, see Fig. 1-2) and so on. All these measurements, just like pupil position, are finally translated to eye orientation.

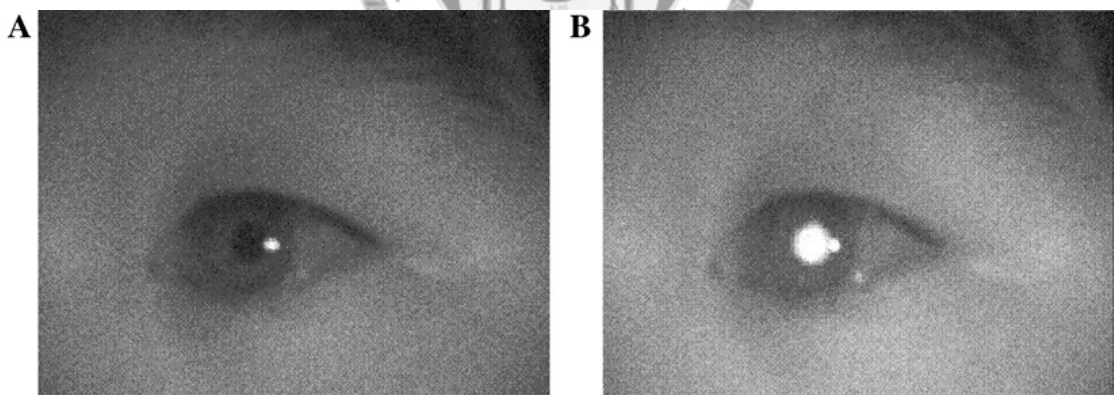


Fig. 1-1 Dark and bright pupil images [3]

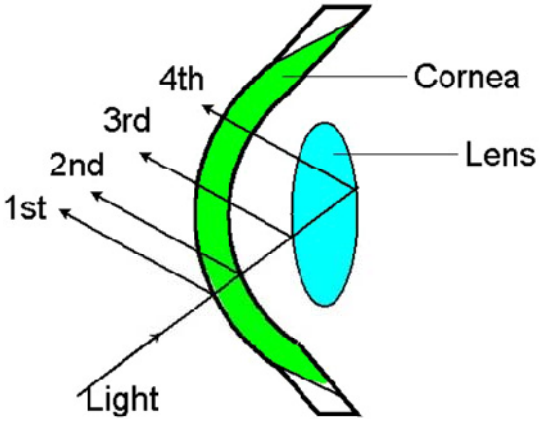


Fig. 1-2 Purkinje images [3]

Many remote eye gaze trackers are base on the corneal reflection technique [4][5][6][7][8][9]. Fig. 1-3 shows a sketch of the pupil corneal reflection technique setup. Just as other eye gaze tracker, it also needs an IR source to generate and detect the first Purkinje image (i.e. corneal reflection) which is the brightest and the easiest to be detected. Suppose the eye is a sphere and rotate around its center. Since the IR source and camera are fixed, the corneal reflection position doesn't change while the eye rotates, therefore it can be used as a reference point. The center of the pupil acquired by the mentioned method and this corneal reflection position defines a vector in the image. Then calibration procedure is needed to compute the mapping from the vector to the computer screen coordinates. The corneal reflection to be a reference point allows small head motion because the reflection follows the head motion. However the calibration is sensitive to the movements along the optical axis as showed in several papers' experiment results.

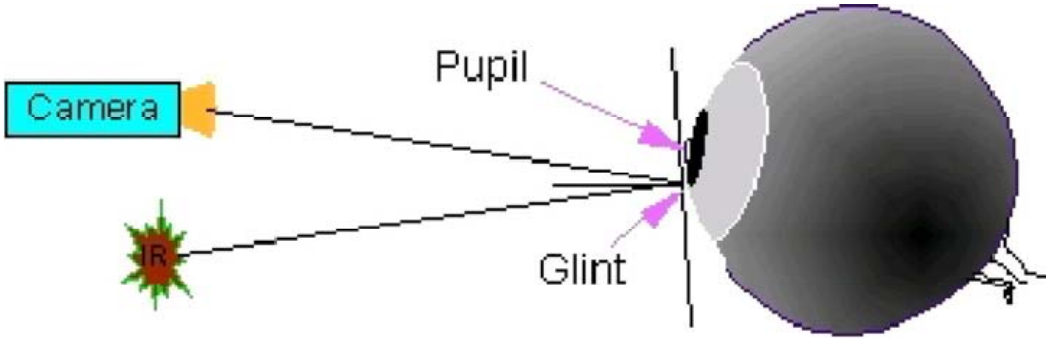


Fig. 1-3 Pupil-corneal reflection technique [3]

For not only being used in laboratory, the calibration procedure of the remote eye gaze trackers has to be more simplified and more robust. The head motion also needs to be unrestricted.

1.3 Thesis work and Contribution

In this thesis, we have designed an eye tracking system for a next generation artificial eye. It can be controlled by the eye movement and turning the artificial eye to focus on the same target as the real eye is. Unlike other aforementioned techniques, the method doesn't need a calibration procedure and allows head motion. The whole system includes three cameras, two microcontrollers and four servo motors. The estimated object's distance is close to the real object's distance as showed in experiment results.

In Chapter 2, the design of the eye tracking system is introduced, including the hardware used in this system and the manufacturer. The latter half of Chapter 2 is going to introduce the controller and several modules of the controller. Further, the control procedure of this system is showed.

In Chapter 3, we will first introduce the whole concept of the system and where the image processing is used and what for. Then, explains each one's details and theorem.

Next, Chapter 4 shows the experiments we took. The purpose of the experiments and the design of the experiment are also included. As the experiments results show, this eye tracking system is capable of tracing the gaze point of the human eye.

Conclusions and Future works are in Chapter 5, along with some possible modification of the mechanism and some applications in the future.

Chapter 2 Mechanism, Actuator and Controller

2.1 Mechanism Design

In this section, the design of the structure and the details of the hardware are introduced. The mechanism consists of a complete eye tracking system capable of measuring the focal distance. In order to reduce the effect of head movement, the mechanism was mounted on a helmet and fixed with a screw. The helmet supports the mechanism securely, which ensures the repeatability of experiments. At first, the reason of such a design is discussed, followed by a discussion of the encountered problems.

Four servo motors are used as the actuators of this system and three cameras are fixed on the mechanism. The whole idea is to create a model which can support two cameras moving independently with two-degrees–of-freedom but finally focused on the same object and controlled by just one human eye. Each part of the system's mechanism is discussed along with its functionality in the following section.

2.1.1Mechanism Design

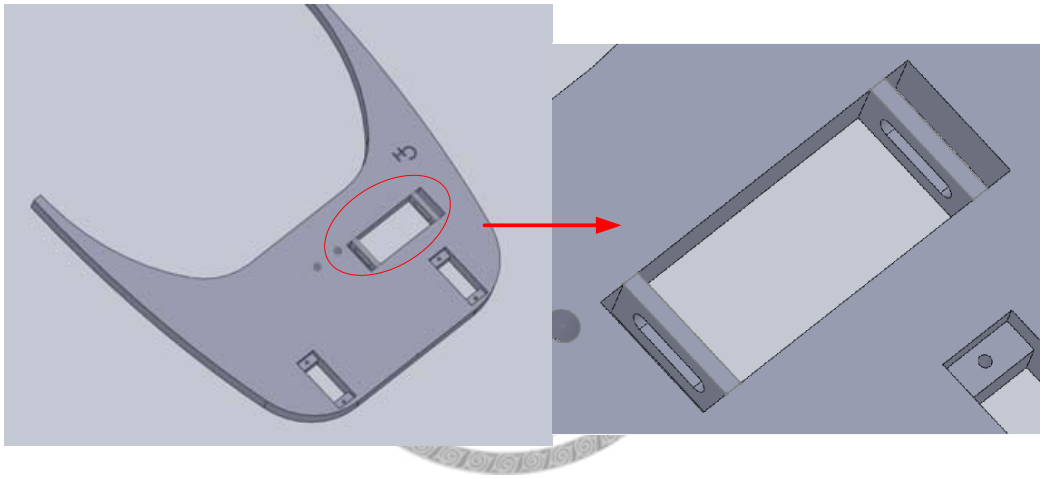
As mentioned before, the major goal of this project is being able to track the human eye and control the two cameras to the desired direction. The most important of all is that this project is mainly designed for someone who lost one of his eyes unlike the other eye tracking systems which is usually designed for someone with two good eyes. In this project, the damaged eye is assumed to be the right eye. Therefore the system is designed according to the idea that it is supposed to be controlled by the movement of the remaining left eye. The software SolidWork is used to design the mechanism and because the designed components here are relatively small and complicated to manufacture, all of the components are created with Rapid Prototyping

(RP). This method doesn't need a fixture or a mold to be able to manufacture. The concept of RP is to cut the module into slices in a computer and a laser source follows the path on each layer to build the module layer by layer. The thickness of the slice is the main factor of the roughness of the work piece. Since it is completely different from traditional manufacture methods, the problem of feeding doesn't have to be considered when designing the module. Because of this feature, it is especially suited for small and complicated work pieces and that is why this method was chosen to realize our mechanism.

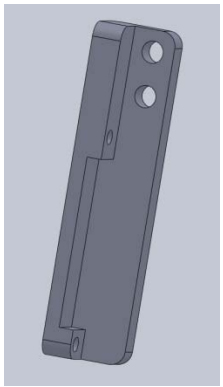
Back to the mechanism design, in order to capture the image of the eye's movement, a fixed-focus camera, the ACV-566F introduced in a later section, was chosen. For better image processing results, having a much clearly focused image is desired, thus, the focal length of the camera is important. The lens of the camera has been changed for the purpose of having a smaller focal length so that the size of the mechanism can be smaller. Considering the flexibility of the mechanism, the ACV-566F camera is originally designed to be fixed with screws and nuts on a groove, therefore, it can be moved along the groove and be adjusted to the best distance from the eye in order to choose the best angle of inclination, as shown in Fig. 2-1(a). In addition to the freedom in the distance, as showed in Fig. 2-1(b), the supporting part also provides another freedom in height by having two different positions for the screw. All of these arrangements are to find the best position and angle to capture the image of the eye's movement. However, the created mechanism doesn't function as ideal as the designed idea. This wide range in flexibility causes difficulties in stability. To solve this problem, a temporary solution is used by fixing the best position and angle. Some modifications to this version must be done in the next version to improve its performance. These modifications are going to be discussed at the end of the section.

In order to use the system as easily as possible, a fixed relative position between head and the mechanism is desired. Under an ideal situation, the head movement does not affect the results of the eye tracking system. In other words, the subject would have unrestricted freedom of movement and a more stable image from the eye could be gotten. For this reason, the mechanism is designed to be attached to the helmet so that it can be more stable in its position. The main plate of the mechanism is designed according to the shape of the helmet, so the mechanism can be firmly fixed on the helmet. Besides, an L-shaped component is screwed onto the helmet and the mechanism as showed in Fig. 2-2.

a)



b)



c)

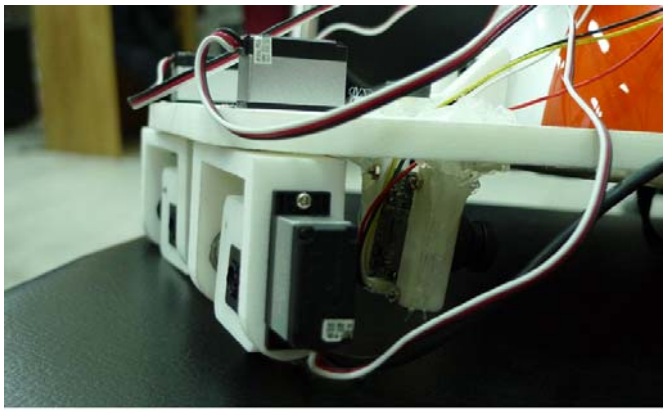


Fig. 2-1 Mechanism structure (a) the groove on the main plate of the framework (b) supporting part for the ACV-566F (c) ACV-566F connected on the model

a) b)

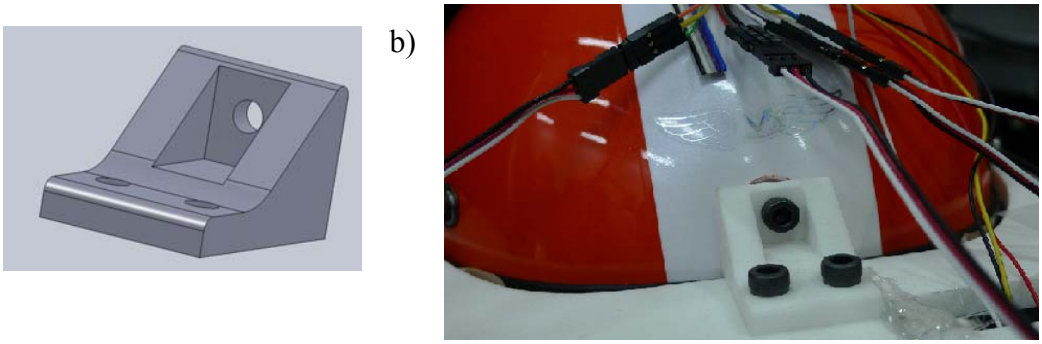


Fig. 2-2 (a) L-shaped component (b) L-shaped component connected the helmet and the model

In order to provide two degrees of freedom for the motion of the other two cameras and be able to move independently, four motors have to be used. After some research, instead of using a stepping motor as the actuator, a servo motor was picked. The reason why the servo motor was chosen is that, in addition to the fact that the current angular position of the motor can be easily known, it can provide a relatively high torque. So the Align DS410M was chosen because of its relatively small size and high torque. Its specifications, features and control method are going to be discussed in 2.1.2. Four servo rudder pieces with different shapes and one servo horn were given by Align motor to transmit the motion from the motor to the modules. One of them was chosen and used in this mechanism as shown in Fig. 2-3.

Fig. 2-3 The servo rudder piece used in the model

From Fig. 2-3, it can be seen that the two motors can separately control the module to the specific angular position in two directions. In order to make the camera

be successfully driven by the motors, another component was designed.

Fig. 2-4 shows the diagram in SolidWork of the component in which the camera can fit in. The servo rudder piece is also shown. In Fig. 2-4, the red arrow points out where the servo rudder piece is designed to fit in; the orange arrow indicates where the wires of the camera can pass through; the blue arrow points out a shaft which matches the bearing put in the other component (what other component?). In that way, the component can freely rotate with the rudder piece.

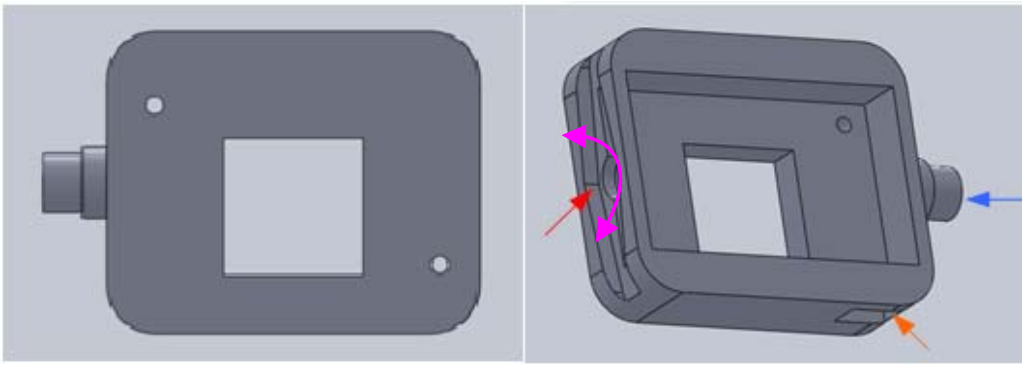


Fig. 2-4 The camera embedded component

Fig. 2-5 shows the assembly part which is composed of two components and each of them provides one free rotating-dimension. Therefore, the desired function of two degrees of freedom in rotation can be achieved. The camera embedded here is not the one mentioned before. Fig. 2-5 shows a front view and a rear view of the camera. The camera is Microsoft LifeCam HD-5000 and its specification and features are discussed in a later section. The most important feature is the function of autofocus because it is similar to the reaction of the human eye. This is the main reason why it was chosen.

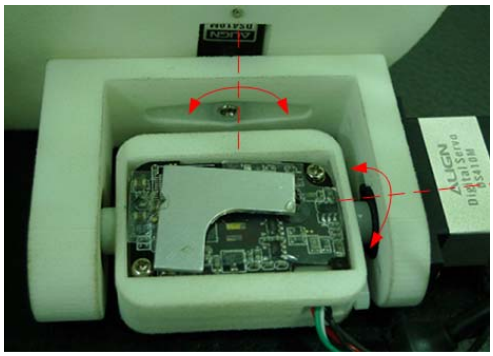
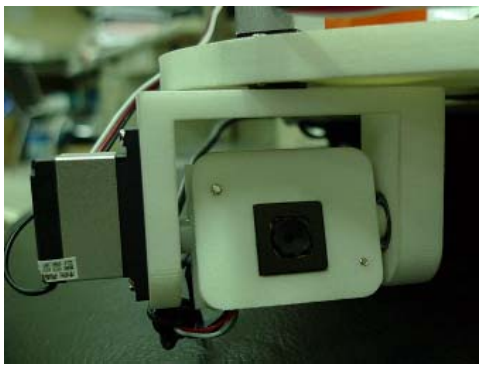
a)  b) 

Fig. 2-5 Assembly part with two degrees of freedom in rotation

The assembly part showed in Fig. 2-5 is suspended on the right side of the plate and another similar one but reversed is suspended on the left side. The reason why it has to be reversed is to avoid collision with each other during operation. Fig. 2-6 shows the diagram in SolidWork of the assembly part with all the components together. In Fig. 2-7, the mechanism is showed.

At the end of this section, some disadvantages and modifications are discussed. In addition to the stability problem of the supporting component of the ACV-566F mentioned before, the whole mechanism is not light enough. Because of this, the mechanism may incline forward and make the experiment's execution more difficult. This problem can improve by pulling all the wires towards the back of the helmet. Finally, the major concern is that the mechanism design becomes an obstacle along the human eye's sightline. After some experiments, the defect of the mechanism design had been revealed and the influence is not so serious. However, these disadvantages still have to be modified in the next version. Some ideas about modified method are discussed in future work.

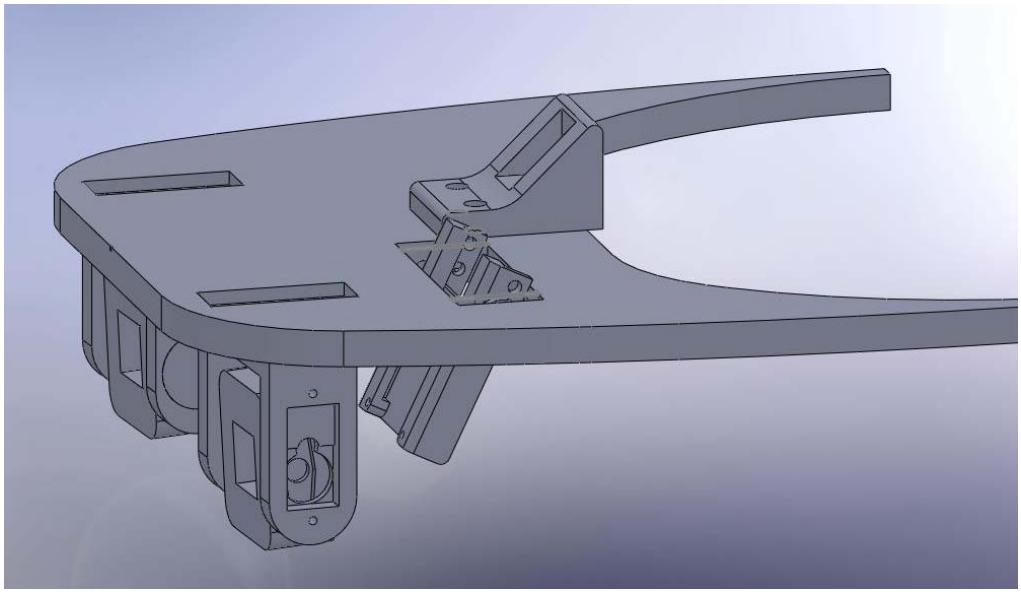


Fig. 2-6 Assembly part of the entire mechanism

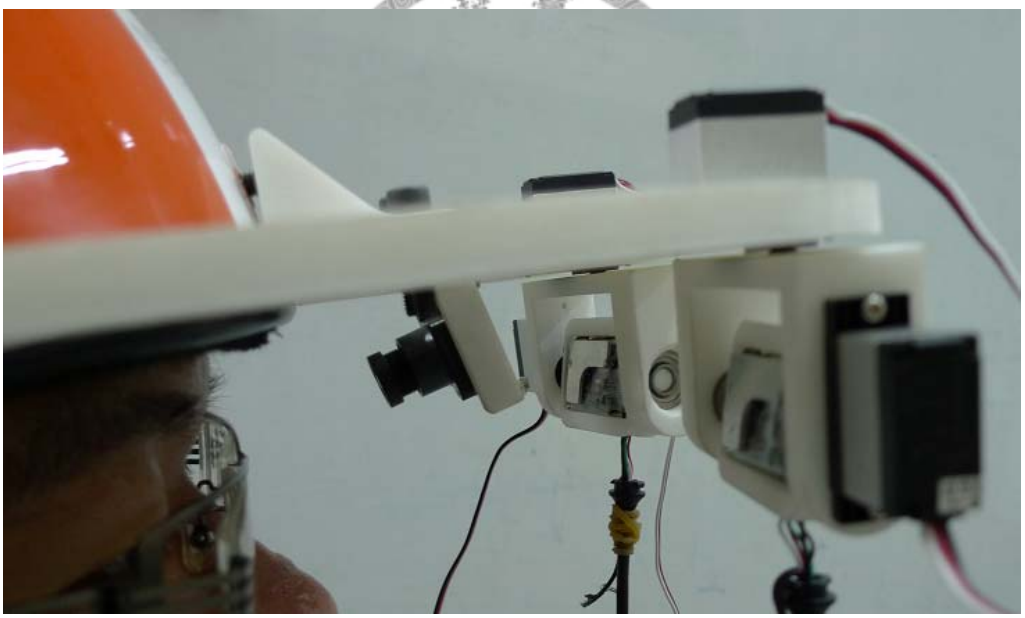


Fig. 2-7 Mechanism in practical operation

2.1.2Align DS410M Servo Motor

As mentioned before, a servo motor is used as the actuating system. The reason why this motor was chosen is that after being compared with most of the servo motors, the DS410M is smaller and lighter. Fig. 2-8 shows how the servo motor looks like. More important, its position can be controlled in a relatively easy and precise way.

After carefully looking at the specifications of the motor, the torque can be 1.8 kg-cm at a driving voltage of 4.8V and can provide a torque of 2.2 kg-cm at a driving voltage of 6.0V. Such a high torque is certainly capable of driving the entire system. The speed of response is up to sixty degrees per 0.09 second, which is sufficient for the movement of the eyeball. The specifications of the motor are shown in Table 2-1.

Table 2-1 DS410M servo motor specifications [15]

Servo Motor specifications	
Model	Align DS410M
Size	22.8 x 12 x 25.4mm
Weight	13.3g
Speed	0.13sec/60°(4.8V) 0.09sec/60°(6.0V)
Torque	1.8kg.cm(4.8V) 2.2kg.cm(6.0V)
Resolution	0.5degree

From Table 2-1, it can be seen that the motor has a relatively small size and a relatively light weight. Because a small artificial eye is desired, this motor was chosen.

Fig. 2-8 DS410 Servo Motor [15]

Different from a stepping motor, the servo motor needs a Pulse Width Modulation (PWM) signal as the actuating signal. The angle of a servo motor depends on the duty

cycle of the PWM signal it receives. In Fig. 2-9, the general concept of a PWM signal can be seen. The “Frame Time” is the duration of one entire cycle which includes the HIGH and LOW parts. The length of the Frame Time is the width of each signal that the servo motor can receive. For the DS410M, its Frame Time is 20ms. The “Pulse Width” or “Duty Cycle” is defined as the time that the signal is in HIGH state. Therefore, giving the specific width of the Duty Cycle determines the angle of the servo motor.

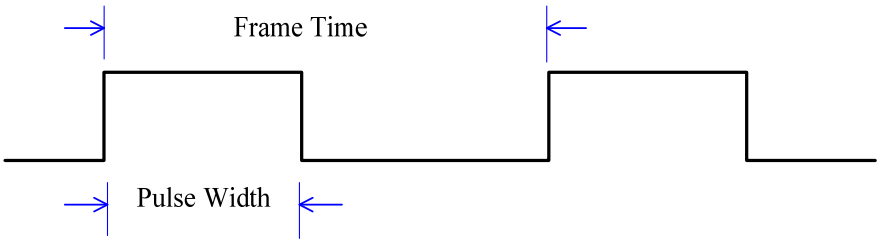


Fig. 2-9 Pulse Width of the PWM signal for servo motor

The used motor Align DS410M has an angle range of -70° to 70°. The maximum duty cycle that the servo motor can receive is of 2.2 ms and the minimum is of 0.8ms. Here, the motor was controlled to a resolution of 1°.

In the next section, the controller is introduced, the PIC18F4580 by Microchip. Microchip PIC18F4580 has four different timers, and by making use of these four timers, the desired waveform can be created. Compared to the timers’ high resolution offered by PIC, the resolution used here is relatively low. For a resolution of 1°, the signal has to be controlled precisely to a width of duty cycle of 10μs.

To achieve the control of the duty cycle to an exact timing, the function “interrupt” is used. Interrupt is usually applied whenever an exact timing is required. Its precise feature ensures that the interrupt can occur in a period of 10μs, which means that the position of the servo motor can be controlled within 1°. Because other calculations need time to be done and they cannot fit within 10μs, two different timers with different interrupts are used.

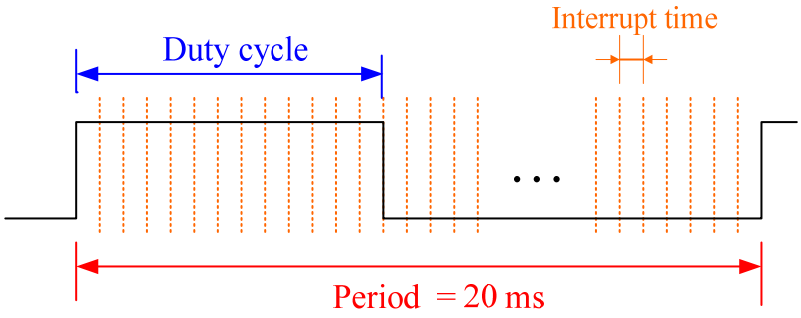


Fig. 2-10 Control of Duty Cycle through interrupt

In Fig. 2-10, the PWM signal given to the servo motor is showed. The orange lines displayed represent each time an interrupt occurs. The whole cycle was separated into many small parts by interrupts. Here, only one of the timers was used to create the interrupt. But as mentioned before, some calculations and transmissions needed to be done during each cycle, and therefore, the time until the next interrupt was not enough to complete all the work. The implementation of two timers to create two interrupts was able to solve this problem.

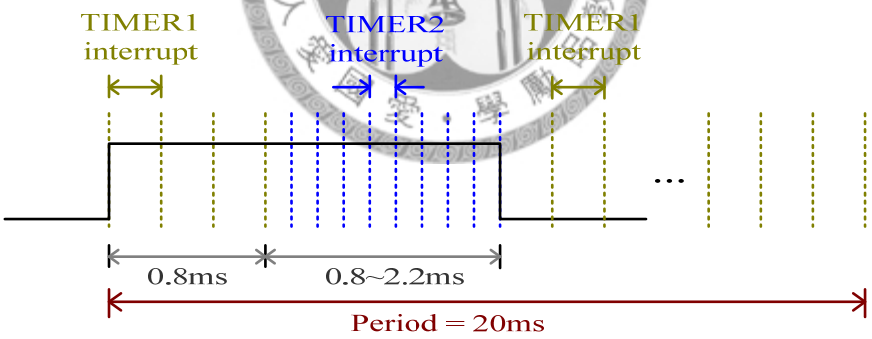


Fig. 2-11 Control of duty cycle through two different interrupts

Since the minimum duty cycle that the servo motor can accept is of 0.8ms and the maximum duty cycle it can accept is of 2.2ms, the whole PWM cycle can be separated into two parts, one with longer interrupt period, one with shorter interrupt period. For this purpose, two timers are used to control it, TIMER1 and TIMER2. First, TIMER1 is used for controlling interrupts that take longer to happen, e.g. 0.2ms. The first 0.8ms of the duty cycle is controlled by TIMER1 and when it reaches 0.8ms,

it changes to TIMER2. Fig. 2-11 shows how these two timers control the PWM signal. The time between 0.8 and 2.2ms is split denser than the rest of the total width of the signal. In this way, a better control is achieved. There are no calculations during the time between 0.8 and 2.2ms; therefore, the interrupts of TIMER2 can happen exactly, maintaining its high precision without worrying about the delay caused by calculations. All the calculations are put in TIMER1, which has a longer period between each interrupt.

With this control method, the motor can be successfully controlled using the PWM signal produced by PIC18F4580. Therefore, through the designed mechanism, two motors are able to change the direction of the camera as desired.

2.1.3 Cameras

The three cameras used in this system are very important parts of the whole tracking system. The data coming from the cameras is used as the input of image processing or either the output. As mentioned previously, two of them are the same and are able to auto focus. Because the camera with fixed focal length is not supposed to move, from now on, the camera is called the fixed-camera. The other two cameras represent the good eye and the artificial eye respectively.

First, the fixed-camera is going to be introduced. Fig. 2-12 shows the outer look and inner look of the ACV-566F[17] and some of its specifications are showed in Table 2-2. The features of this camera are its wide range of picture angle, relatively low luminance and high resolution. It was originally designed as a monitor used in elevators or at stairwells where light is limited. The characteristics exactly fulfill the needs since light is covered a lot by the mechanism at the position of the ACV-566F.

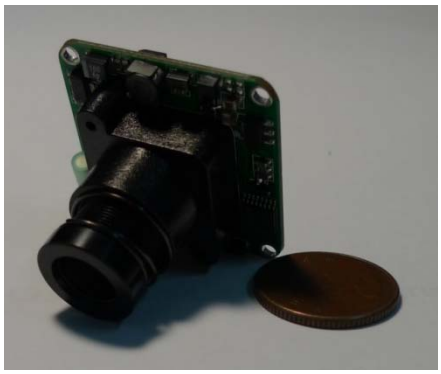
a)  b) 

Fig. 2-12 ACV-566F a) outer look[17] b) inner look

Table 2-2 ACV-566F camera specifications

Low-Lux., Intermediate-Res. camera specifications	
Model	ACV-566F
Chip	1/3” SONY Color CCD image sensor
Size	36(W)*36(H)*15.9(D) mm
Picture angle	92.6°
Weight	53g
Voltage	DC12V
Lens specification	F3.6mm/F2.0
Horizontal resolution	420 TV Lines
Min-luminance.	0.5 Lux./F2.0
S/N ratio	48dB

The current focal length of the lens is about 2.5cm. Fig. 2-13(a) shows the image taken by ACV-566F with the target at its focal length and Fig. 2-13(b) shows the image taken at a distance of about 4 cm. It is easy to see the difference between these two images. In order to obtain a relatively clear image, the best distance from the eye and the camera is 2.5cm. However, if the camera is put that close to the human eye, the field of sight becomes almost shaded. On the contrary, lens with larger focal length, although is going to be less shaded, but the corresponding mechanism

becomes bigger. After considering all this information, a decision was made. Instead of strictly demanding the distance between the eye and the camera to match the focal distance, the camera is designed to be put at about 6.5cm away from the eye. Fig. 2-14 shows the image taken at that distance. The quality of the image is acceptable and the effect of image processing is not affected. Consequently, the design of the mechanism is acceptable.

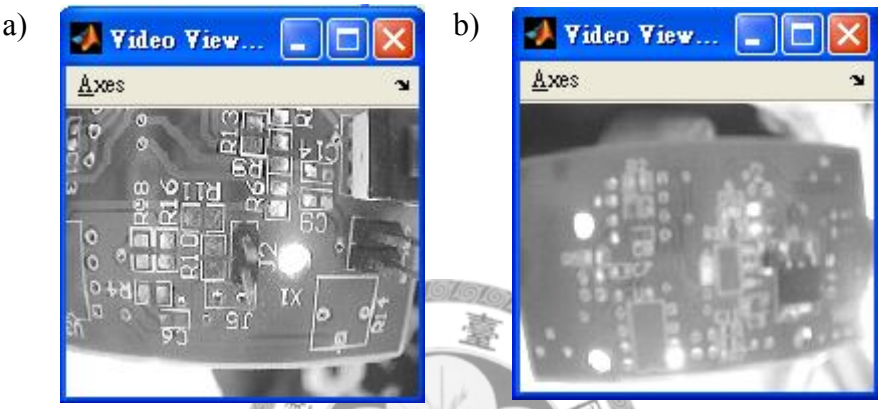


Fig. 2-13 Image captured by Camera ACV-566F (a) focused (b) unfocused

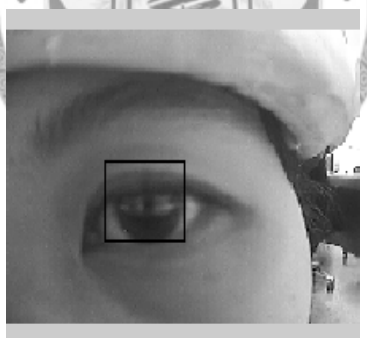


Fig. 2-14 Image of eye taken in 6.5cm by ACV-566F

After discussing about the ACV-566F, the Microsoft LifeCam HD-5000[16] is introduced, which has the function of autofocus. Fig. 2-15 shows its outer look and the inner look. Besides its feature of autofocus, its relatively small size and weight are also the reasons why was chosen. HD-5000 was originally a webcam with a microphone which has been taken off.

a) b)

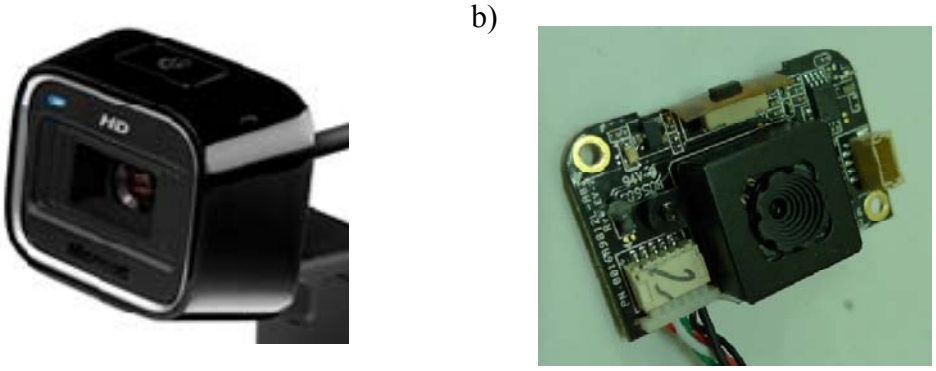


Fig. 2-15 Microsoft LifeCam HD-5000 [16] a) outer look b) inner look

Table 2-3 Microsoft LifeCam HD-5000 specifications

LifeCam specification	
Product Name	Microsoft LifeCam HD-5000
Size	40.8(W)*37.8(H)*10.9(D)
Weight	96.5g
Sensor	CMOS sensor technology
Resolution	Motion Video:1280*720pixel
Imaging Rate	Up to 30frames per second
Field of View	66° diagonal field of view
Imaging Features	▪ Auto focus, range from 4’’ to infinity ▪ 16:9 widescreen

Both ACV-566F and LifeCam HD-5000 have much higher resolutions than the ones actually used in the following experiments. However, the resolutions of these cameras are all set to the lowest resolution, 176×144, which is restricted by the calculation abilities of the hardware. As a result, the error of the experiments is affected in some way.

2.2 Controller

In this section, the controller PIC18F4580 by Microchip will be introduced,. It is a microcontroller that has many functions in it and is suitable for all of our purposes.

After finishing the image processing in the computer, the information will be transmitted to the MCU through serial communication. Once the microcontroller receives the data, it starts to create the corresponding PWM signal for the servo motor and to copy the data to the other MCU at the same time. The interrupts and timers can be put into use to create the precise PWM signals. On the other hand, PIC18F4580 has another function for communication between microcontrollers called CAN-BUS.

CAN-BUS is the abbreviation of Controller Area Network [18], which is a computer network protocol and bus standard designed to allow microcontrollers and devices to communicate with each other and without a host computer. The microcontroller, PIC18F4580, supports this protocol, which is used in order to create the communication protocol between the microcontrollers.

In addition to CAN-BUS, another communication protocol is also used here: Universal Synchronous Asynchronous Receiver Transmitter (USART). USART is also a common way of communication between the microcontroller and PC. More details will be discussed in the following sections.

2.2.1Introduction to PIC18F4580

Microcontroller means to put the functions of a Center Processing Unit (CPU), on a semiconductor chip. Since the number of external pins of the microcontroller is limited, most of its pins are programmable.

The PIC series of microcontroller is developed by Microchip. The Microchip PIC18F4580 microcontroller is a 40-pin enhanced flash microcontroller with ECAN Technology, 10-bit A/D converter and nanoWatt technology. It also has many

power-managed modes, for example: Run Mode, Idle Mode, Sleep Mode, allowing the users to have a wide range of selections depending on the functions the user desires.

Another interesting feature is that it has a flexible oscillator structure. It has four crystal modes, being able to go up to 40MHz. It has a 4x Phase Lock Loop (PLL), which is available for crystal and internal oscillators. It has two external RC modes, up to 4MHz, and two external clock modes, up to 40MHz. Except for the external clock, it also has an internal oscillator block, which has 8 selectable frequencies, from 31 kHz to 8MHz and provides a complete range of clock speeds from 31 kHz to 32MHz when used with PLL.

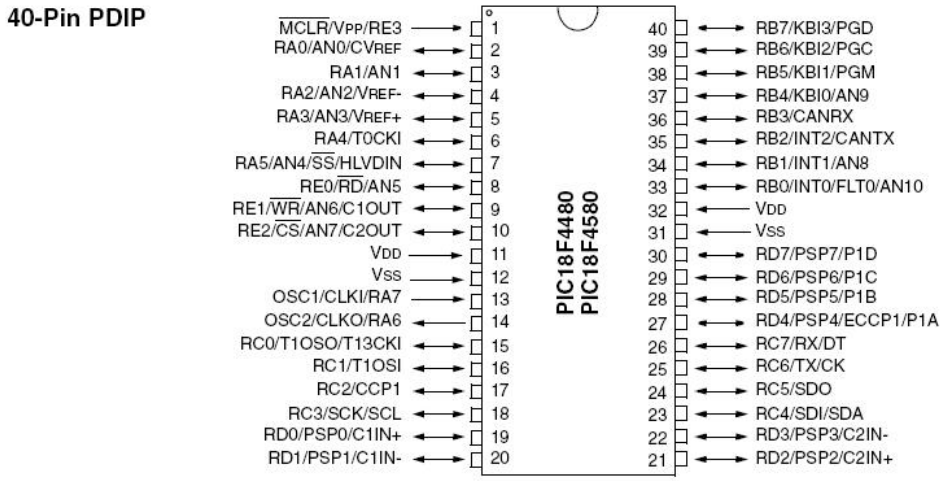


Fig. 2-16 PIC18F4580 Pin distribution [11]

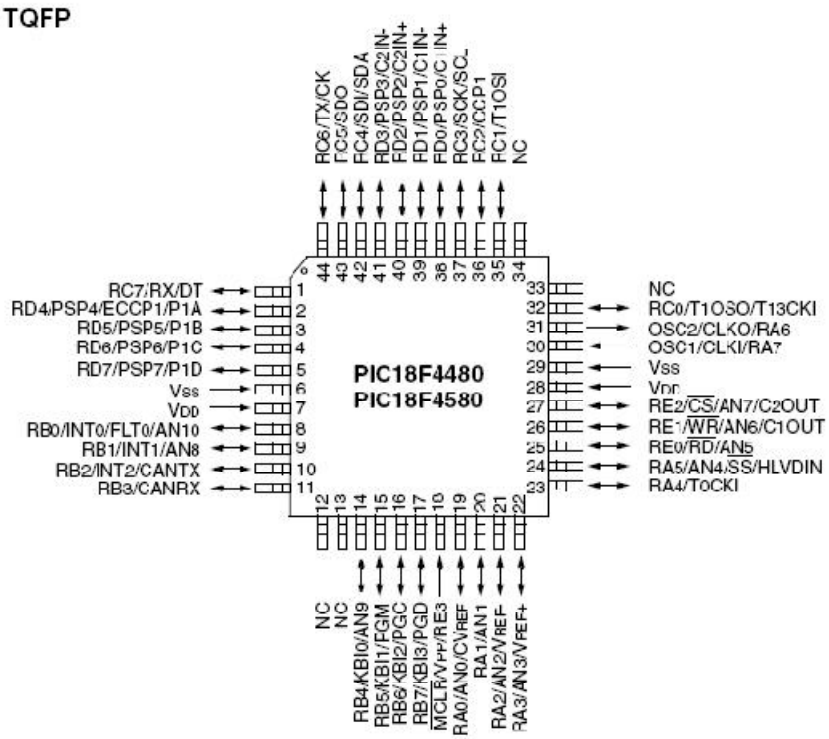


Fig. 2-17 TQFP layout of a PIC18F4580 microcontroller [11]

Fig. 2-16 and Fig. 2-17 show the pin distribution of the PIC18F4580 microcontroller. It has a total of 40 pins (44 pins in the TQFP layout, 4 pins having no connections) and it provides the user a wide range of features. Some of the features are as follows.

- 8-bit microcontroller
- Flash Program Memory Type
- 32Kbytes of Program Memory Size
- 10MIPS of CUP Speed
- 1536RAM
- 256bytes of Data EEPROM
- 36 I/O grouped in 4 groups
- Enhanced CAN protocol module
- Internal Oscillator

- Self Programming
- 40 MHz Max. Speed
- LIN USART
- 4 Timers Module
- Interrupt feature
- Capture/Compare/PWM modules
- Enhanced Universal Synchronous Receiver Transmitter (EUSART)
- 10-bit Analog-to-Digital Converter

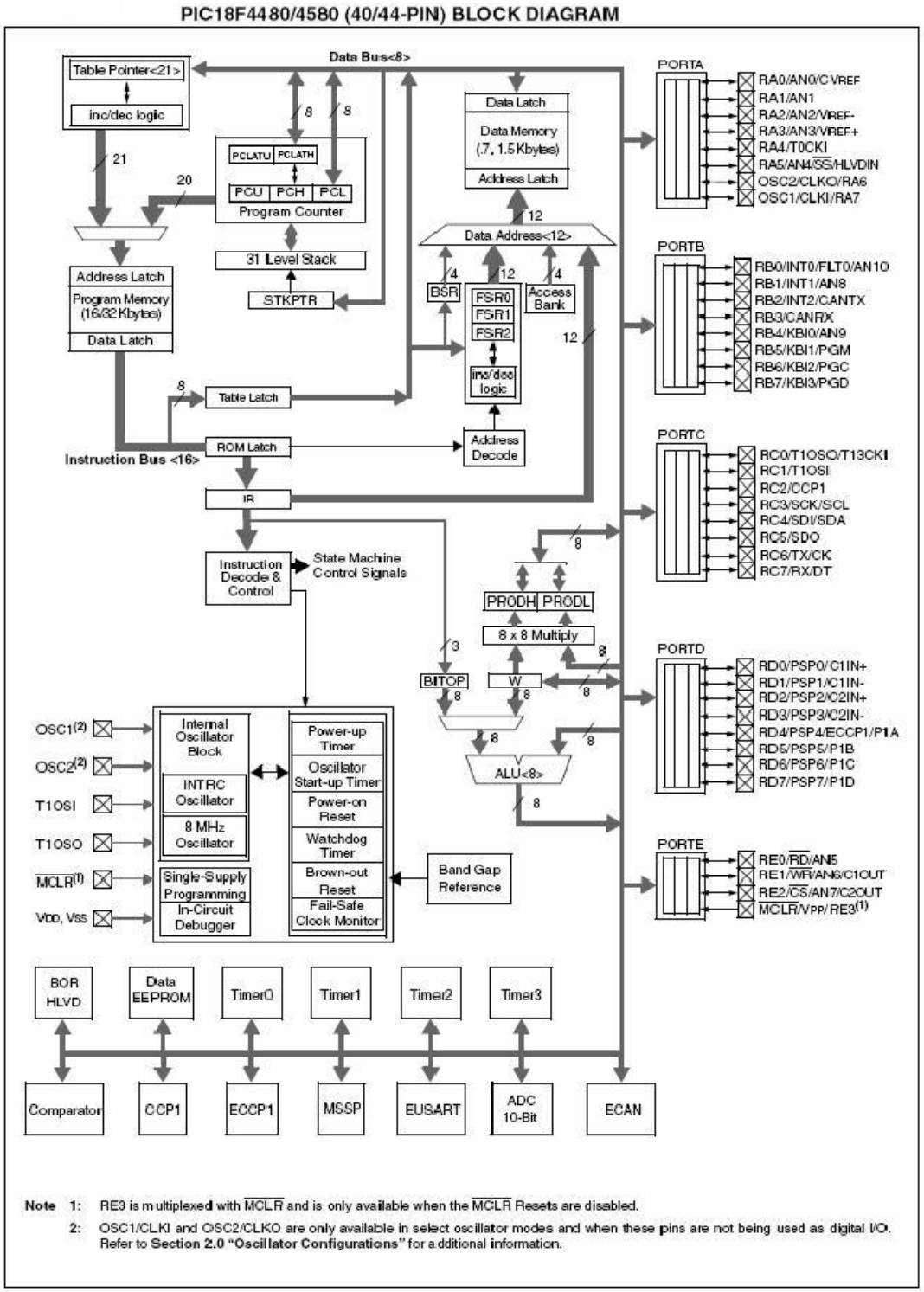


Fig. 2-18 PIC18F4580 Block Diagram [11]

Fig. 2-18 shows a block diagram of the PIC18F4580. The basic features are all presented in this block diagram. For more details, refer to the datasheet of the microcontroller [11].

Table 2-4 Device Features [11]

DEVICE FEATURES

Features	PIC18F4580
Operating Frequency	DC – 40 MHz
Program Memory (Bytes)	32768
Program Memory (Instructions)	16384
Data Memory (Bytes)	1536
Data EEPROM Memory (Bytes)	256
Interrupt Sources	20
I/O Ports	Ports A, B, C, D, E
Timers	4
Capture/Compare/PWM Modules	1
Enhanced Capture/ Compare/PWM Modules	1
ECAN Module	1
Serial Communications	MSSP, Enhanced USART
Parallel Communications (PSP)	Yes
10-Bit Analog-to-Digital Module	11 Input Channels
Comparators	2
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable High/ Low-Voltage Detect	Yes
Programmable Brown-out Reset	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set Enabled
Packages	40-pin PDIP 44-pin QFN 44-pin TQFP

Asides from the block diagram presented in Fig. 2-18, Table 2-4 presents the main features of the microcontroller.

2.2.2Introduction to MPLAB, ICD2, PICKit2

This section introduces the tools that Microchip provides for development of the controller. These tools include MPLAB IDE, ICD2 or PICKit2, and APP001 Rev.2 development bard. The different tools presented here are for the purpose of providing a better development environment.

MPLAB Integrated Development Environment (MPLAB IDE) is a free, integrated toolset to develop code for embedded microcontrollers. It has a built-in simulator and debugger and it can work together with the debuggers/programmers provided by Microchip. Therefore, even if the hardware is not finished, users can begin testing the code with the simulator, a software program that simulates the execution of the microcontroller.

MPLAB supports programming in Assembly language, but if users want to develop the project in an different way, MPLAB also supports the C code with its own C compiler. Writing the program in C language to develop the PIC microcontroller is much easier to read and to program. MPLAB also supports a more flexible way, in-line assembly language, in other words, the program can directly call the assembly language. As discussed above, the Microchip provides flexibility in developing the program.

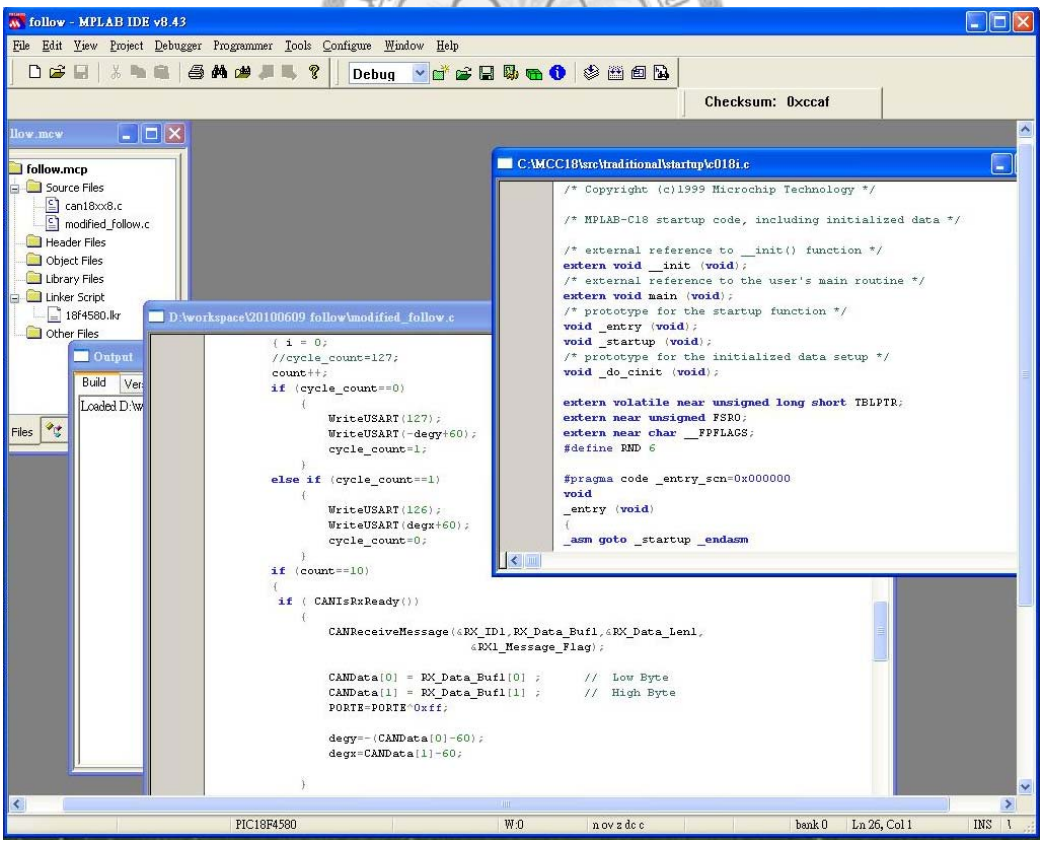


Fig. 2-19 MPLAB Interface

Fig. 2-19 shows the interface of the MPLAB IDE program. It provides a superior

debugging environment. Users can pick some windows to open depending on the things they need to watch. In the figure, it can be seen that it has a “Watch” window, in which the symbols or different registers can be added and its current value can be watched. Because MPLAB IDE is an integrated development platform, programmers can write and revise the source code simultaneously. The MPLAB IDE works also as a compiler and text editor. The C codes can be written inside the program for easy programming.

Microchip also has debugger/programmer devices, like MPLAB ICD2 and the PICKit2. MPLAB ICD2 is a low cost, real-time debugger and programmer for selected PIC microcontroller units. Using Microchip Technology’s proprietary in-circuit debug functions, programs can be downloaded, executed in real time and examined in detail with the debug functions of the MPLAB. Set watch variables and breakpoints from symbolic labels in C or assembly source code, the current values of the variables can be seen. MPLAB ICD2 can also be used as a development programmer for supported MCUs. Fig. 2-20 shows a picture of the MPLAB ICD2 by Microchip.

Fig. 2-20 MPLAB ICD2

The PICKit2 development tool showed in Fig. 2-21 works similarly to the MPLAB ICD2. It is also a low-cost development debugger/programmer tool. It

supports 8-bit, 16-bit and 32-bit microcontroller. It also enables in-circuit debugger for easy programming for the user.

Fig. 2-21 Microchip PICKit2

Either one of these tools, connected to the user's computer and a development board (or target board) that has a microcontroller on it, can allow the user to debug and program the target easily. There is a development/target board by Microchip named APP001 Ver.2 which can be used as a development board. With all these tools together, a connection to the in-circuit development tool can be established; here PICKit2 is used as the debugger. The debugger is connected to the development tool with the microcontroller on it. After connecting, the working with the program can start.

2.2.3USART

In this section, the Universal Synchronous Asynchronous Receiver Transmitter is introduced. USART is a serial I/O module that can communicate with other hardware. In computer systems, the most well known USART interface is RS232. The datasheet of PIC18F4580 has mentioned that the USART module can typically be used in RS-232 and RS-485 systems. RS-485 is developed based on the RS-232 protocol and defects like the communication distance length are improved. Baud rate is enhanced

and other serial communication functions are added.

To establish communication between the PC and PIC18F4580, since RS-232 is the original connector provided by PC, RS-232 is selected to be used. RS-232 has some restrictions in communication, such as, the maximum Baud rate, which is 20Kbits per second with a transmission distance of about only 15 meters. Table 2-5 shows the relationship between Baud rate and the Maximum cable length.

Table 2-5 RS-232 Baud rate vs. cable length

Baud rate	Max. cable length (ft)
19200	50
9600	500
4800	1000
2400	3000

The equipment at the far end of the connection is named the DTE device (Data Terminal Equipment), usually a computer or a terminal. Equipment at the near end of the connection is named the DCE device (Data communication Equipment). The cable linking the DTE and DCE devices has two kinds of connectors. Fig. 2-22 shows the connector and the pin definition is showed in Table 2-6.

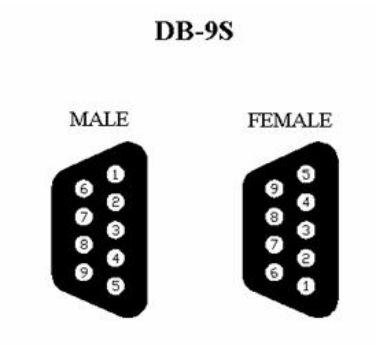


Fig. 2-22 RS-232 connectors

Table 2-6 DB-9 pin definition

DB-9			
1	Received Line Signal Detect	6	DCE Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	DTE Ready	9	Ring indicator
5	Signal Ground		

Signal name implies the data transmit direction, such as Transmit Data and Receive Data. The Transmitted Data at the DTE side becomes Received Data at the DCE side. Fig. 2-23 shows the conventional usage of signal names in DB-9 and the arrows represent the direction of transmission.

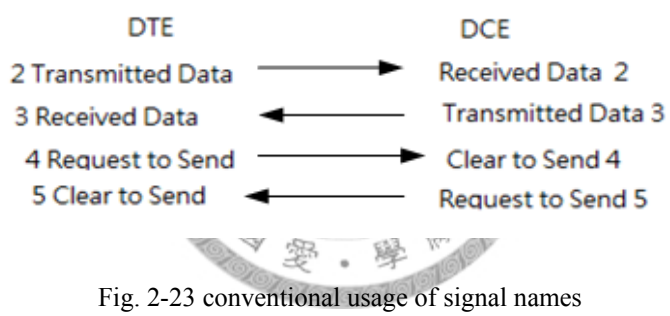


Fig. 2-23 conventional usage of signal names

The RS-232 defines a communications method where information is sent bit by bit on a channel. This means, the sending process can start at any time. Therefore, if the data can start at any time, there is a problem for the receiver to know which bit is the first one. To overcome this problem, RS-232 standard defines a data frame which consists of one start bit, data bits, one parity bit and stop bits. Once the start bit is received by the receiver, the receiver recognizes and understands that the information starts from this bit. The format form of the data is shown in Fig. 2-24. The bits directly following the start bit are the data bits. After the data bits, come the parity bit, which is used to examine the transmission accuracy. Finally, at the end of the data

frame is the stop bit which can have different lengths. In fact, it is not a real bit but a minimum period of time at the end of each data frame in which the line must be idle or in mark state.. A stop bit length of 1 is acceptable for all data sizes.

The signal level of the RS-232 pins has two states. A high bit, as logical “1”, or Mark state, is identified by a negative voltage from -15V to -3 V and a low bit, as logical “0” called Space state is identified by a positive voltage from +3V to +15V. The voltage level from -3V to +3 V will be viewed as undetermined. Fig. 2-24 has an example of this signal standard.

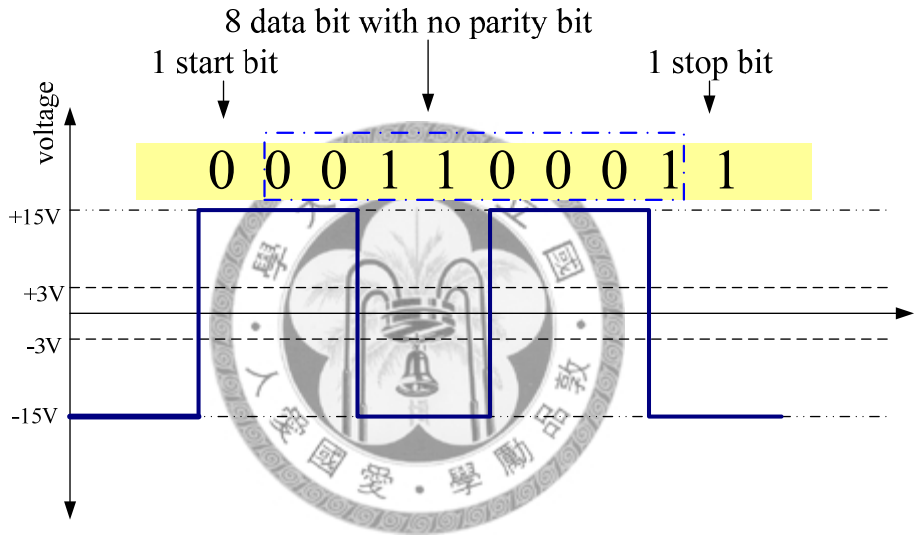


Fig. 2-24 RS-232 Standard Signal example

To transmit data correctly, DTE and DCE must be programmed to use the same transmission speed which is known as baud rate, indicating the number of bits per second for the RS-232 device to transmit and receive the data. The datasheet of PIC18F4580 mentions the setting method of the baud rate and the corresponding error rate. The USART protocol uses certain buffers to store the data whether are going to be sent or just received. Whenever the data is successfully transmitted the buffer will be cleared, and ready for the next data.

Because of the different voltage levels between RS-232 and the standard logic used in PIC18F4580, there is a voltage level shift circuit used to interface these two

systems. Fig. 2-26 shows the most common circuit for this purpose which includes a voltage level shifter MAX232. Fig. 2-25 also shows the layout of the chip and the block diagram of the MAX232. With this circuit, the PC can successfully communicate through RS-232 with the PIC18F4580.

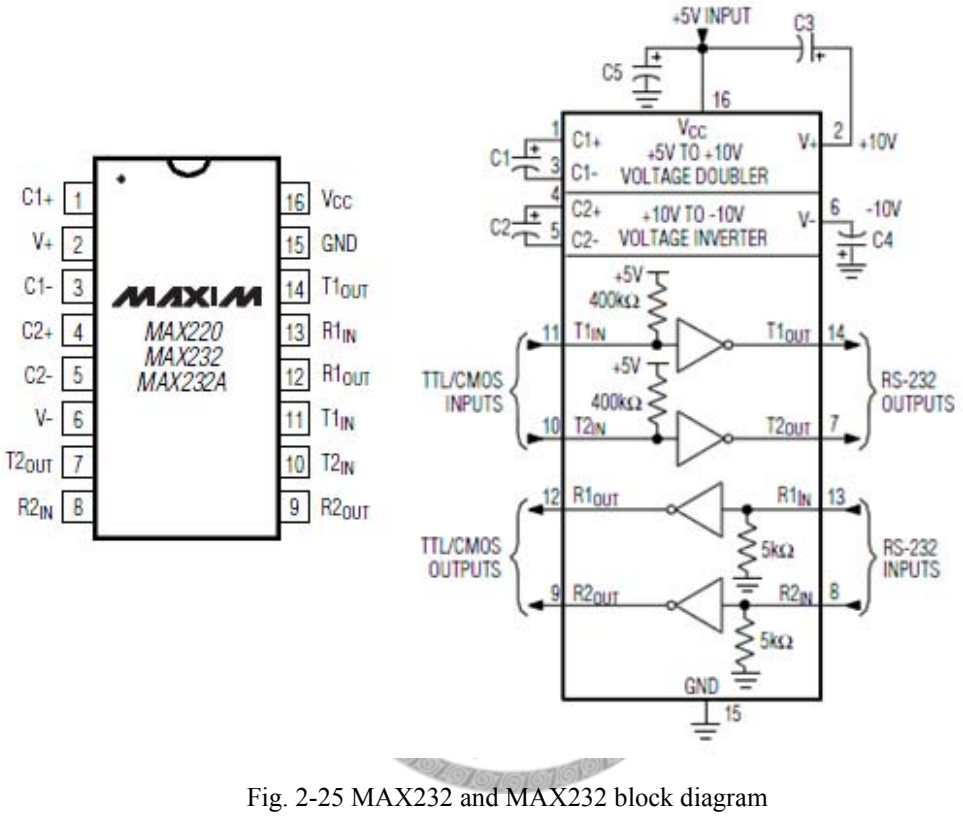


Fig. 2-25 MAX232 and MAX232 block diagram

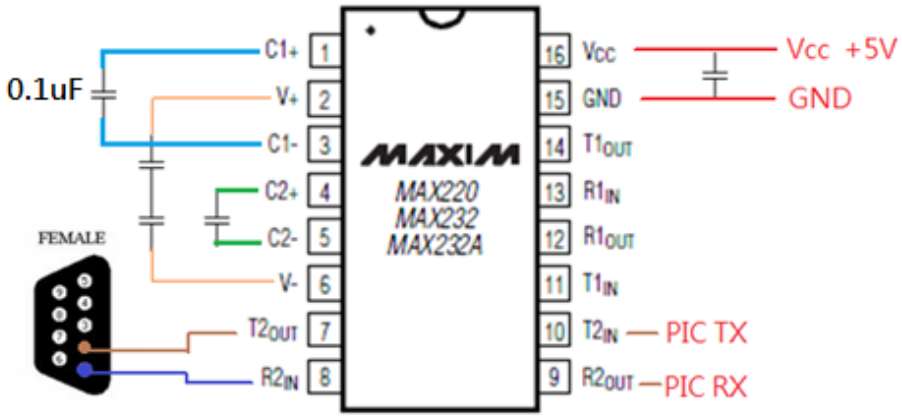


Fig. 2-26 voltage level shift circuit

USART is responsible for transmitting data between MCU and PC. The goal is to transmit the image processing results from the PC, giving instructions on the value of the needed duty cycle. One PIC is in charge of one camera's motion, or in other words, two motors. As discussed in 2.1, after the image of the eye is acquired by the fixed-camera and fed into image processing, the computer calculates the current position of the eye and determines the angular position to which the camera should turn to. This angular position is sent through USART to the microcontroller. The application of USART can be seen in section 3.4.

2.2.4CAN-BUS

In this section, the Controller Area Network Protocol is introduced. CAN BUS was originally designed for automotive applications but because of its convenience and flexibility, it is also use in other areas. It is a computer network protocol and bus standard designed to allow microcontrollers and devices to communicate with each other and without a host computer. PIC18F4580 microcontroller also has this function and allows communication between controllers. Later in this section, the CAN BUS features of the microcontroller are introduced. The CAN BUS protocol can transmit a maximum of 8 Bytes of data each time.

Table 2-7 shows the relation between the Bit Rate and Bus Length. Here it can be seen that CAN BUS is able to transfer data to up to 1000 meters of distance. Compared to USART, the transmission speed is much faster, which makes this protocol really useful.

Table 2-7 CAN BUS protocol Bit Rate vs. Bus Length

Bit Rate (Kbit/s)	Bus Length (m)
1000	40
125	500
50	1000

As mentioned earlier, CAN BUS doesn't need a host computer to be able to transmit. Every microcontroller connected to this BUS is called a node. Every node can transmit and receive messages from other nodes, but not simultaneously. This has the advantage that for example, in the application of automobiles, sensors, controllers, and other units from different manufacturers, or companies, can be connected in the bus and still be able to communicate between each other. This BUS supports at least 32 nodes connected to it, each of them being able to receive and transfer.

The CAN BUS featured in the microcontroller has four different configuration modes: Normal, Listen Only, Sleep and Loopback. The Normal mode is the one used for normal communication between each of the nods and allows the node to participate in the communication. In Listen Only mode, it cannot send any message, only read. The Sleep mode is the mode in which the node doesn't participate in the communication and it is a power-saving mode. The last one, the Loopback mode, is a more unusual mode. Since it is a self-test mode, it's useful for debugging and developing phases. No CAN transceiver is needed on the bus.

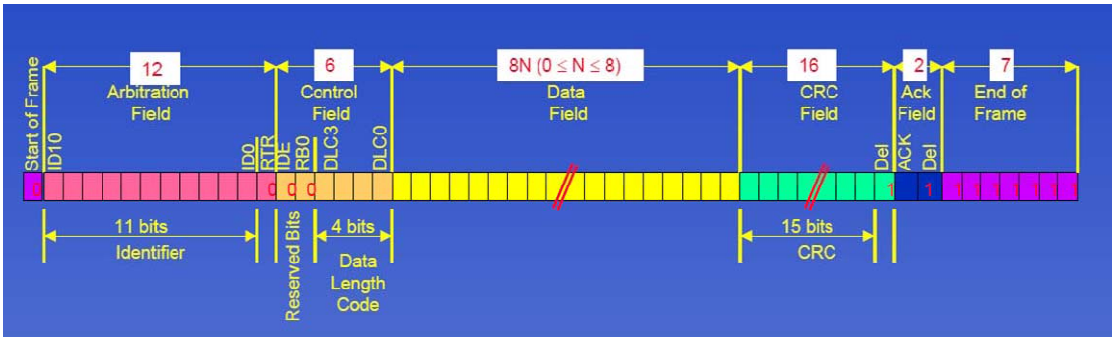


Fig. 2-27 CAN BUS Dataframe

Fig. 2-27 shows the data frame of the CAN BUS protocol. From this figure, it can be seen that every time a data message is sent, there is a start bit at the beginning to indicate that after it comes a set of information. After the start bit, 11 bits, defined as the identifier are sent. These bits allow the other microcontrollers connected to the bus to know the identification of the message, so as to be capable of accepting or rejecting the message when they read this message. After the identifier, there is a set of 4 bits that indicates the length of the data. As mentioned before, the maximum data length that CAN BUS can transmit and receive is 8 bytes. This is where it indicates how long the following message is. After these bits comes the message itself. After the message itself, there's a set of 15 bits that allow us to know if the message is correctly sent and received. It is called CRC field, which contains the errors that happened and can be used for knowing the problem. After the CRC field, the acknowledge field comes. Because the CAN BUS module will continue sending message until the message has been taken, the acknowledge bits are necessary to understand whether the message was already received by someone on the bus yet. Finally, at the end of the data frame is the End of Frame bits, letting the nodes know that the message ends here.

The CAN BUS protocol, just like other communication protocols, uses buffers to temporarily store the data that is going to be sent or received. The microcontroller has 3 transmit buffers and 2 receive buffers. When a node wants to send a message, it will first check if there's any empty transmit buffer. When all of the transmit buffers are full, it has to wait until any of them becomes empty, then, put the message into the buffer waiting to be sent. Whenever the message has been successfully transmitted, the buffer originally containing the message will be cleared, so that other messages waiting to be sent can be put into the buffer. Unlike the other communication protocol, CAN BUS has a special regulation for receiving data. It uses masks and filters to

identify the message whether to receive it or reject it. If the message fulfills the conditions, it is read and put into the receive buffer.

Because there are so many nodes on the bus, the node has to be able to recognize if the message is for it or not. In order to achieve this, the CAN BUS protocol has filters and masks. The CAN BUS in this microcontroller has 6 filters and 2 masks. Fig. 2-28 shows the relation between transmit buffers, receive buffers and filters and masks. After going through the transmit buffers, message enters the CAN BUS, but going first through a chip which changes the values of the signal to the standards of CAN BUS. The layout of the chip and the block diagram is shown in Fig. 2-29 and Fig. 2-30. CAN BUS protocol needs two lines to connect all the nodes, denoted as CANH and CANL, as showed in Fig. 2-33. With these two lines, all the nodes can communicate with each other.

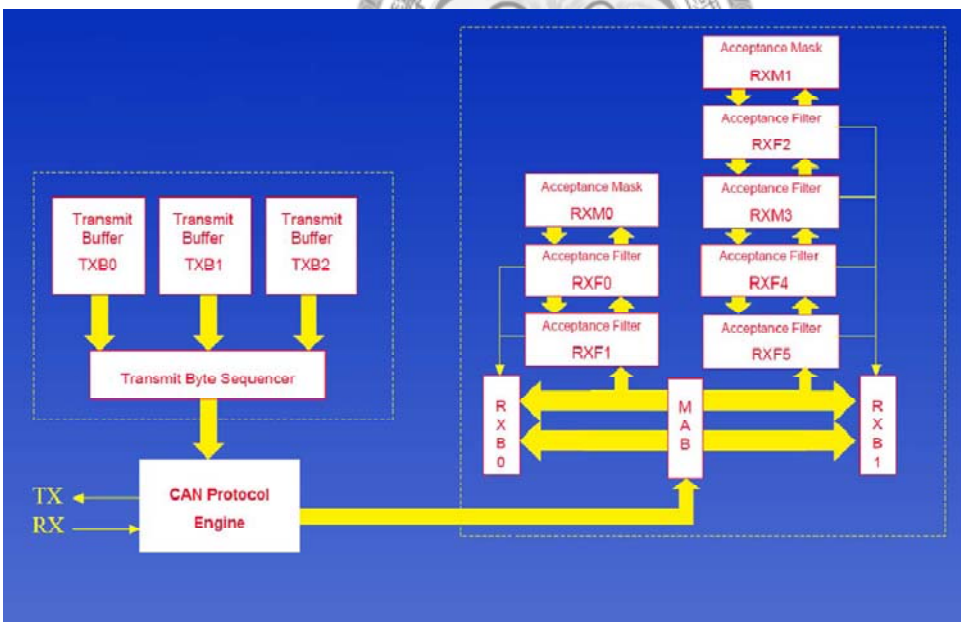


Fig. 2-28 Filters and Masks diagram

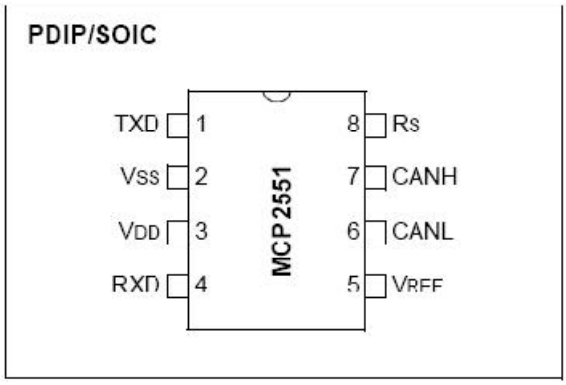


Fig. 2-29 MCP2551

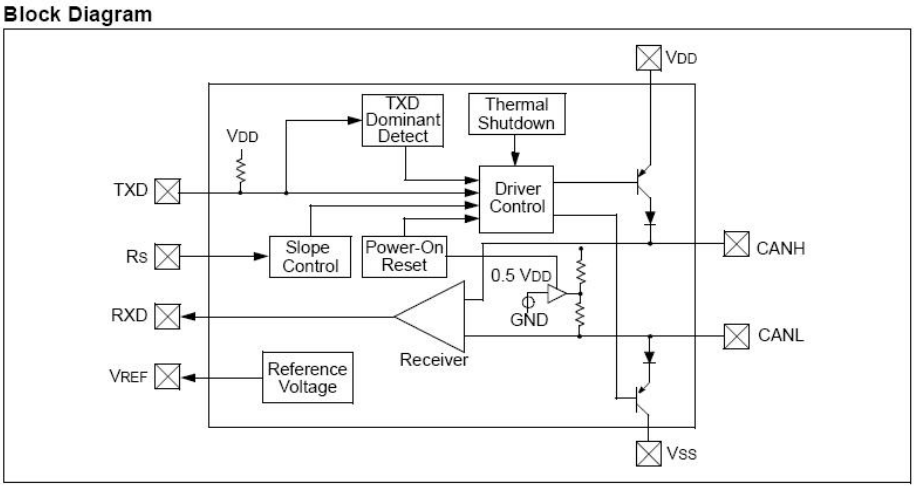


Fig. 2-30 MCP2551 block diagram

To understand the regulations about how each node decides to take the message or not, the relation between masks and filters will be explained. Mask determines if the filter is to be compared with the message identifier. If the mask bit n is 0, then that the message bit n will be accepted no matter what the filter is. If the mask bit n is 1, then it has to further check the filter bit n and the identifier bit n if they fulfill the condition. When the filter bit n and the message ID bit n matches, the message is received. These rules provide a flexible way to set a very wide range of filters and masks, letting us decide which message to receive and which to reject. Table 2-8 clearly shows the relation between filter, mask and the identifier and how they decide to accept or reject the message.

Table 2-8 Masks and Filters for the CAN BUS protocol

Filter/Mask Truth Table			
Mask bit n	Filter bit n	Identifier bit n	Accept/Reject
0	X	X	Accept
1	0	0	Accept
1	0	1	Reject
1	1	0	Reject
1	1	1	Accept

Because CAN BUS protocol is relatively complicated to set up, Microchip provides a library, AN738 [12], for users to set up the CAN BUS protocol. With this library, users can easily set up the protocol by some functions which have been already written.

In this eye tracking system, CANBUS is applied for transmitting information between two microcontrollers whenever the eye has obviously changed its gazing target. In that situation, the left camera could certainly follow the eye movement at any time since its motion is connected to the eye movement. But for the right camera, it is desired to avoid the target from not being showed up in the image of the right camera. If this situation happens, the results of image processing will be incorrect. To solve this problem, whenever the left camera changes its angular position above a predefined value, the right camera has to follow it by receiving the new angular position through CAN BUS.

2.2.5Printed Circuit Board (Controller Board)

The needed features of the microcontroller are taken out, such as Input and Output pins, RS-232 and CAN-BUS. Our own circuit is layered, making the whole board as small as possible. Fig. 2-31 shows a picture of the layered PCB.

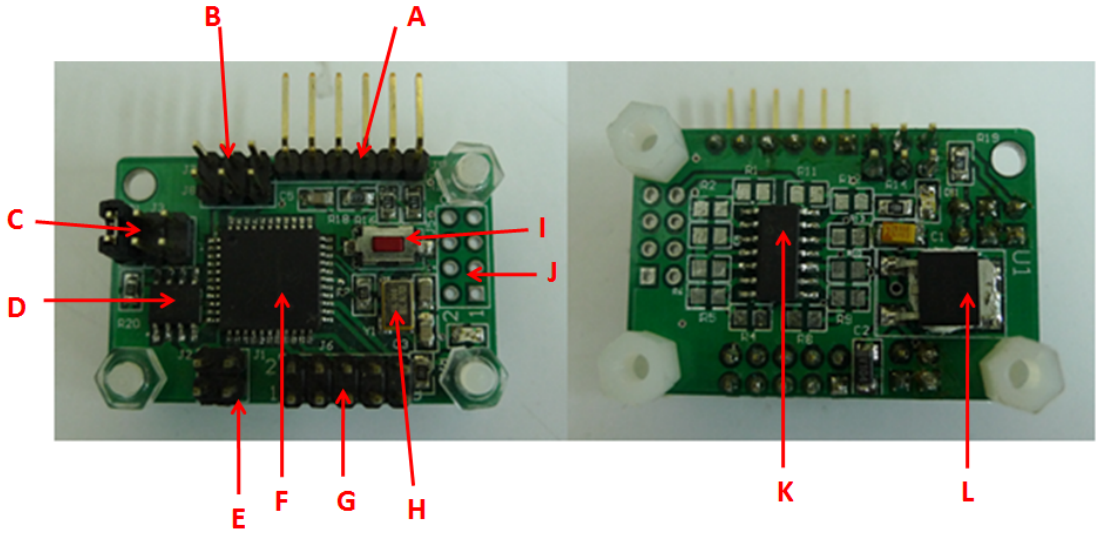


Fig. 2-31 Printed Circuit Board

Section A is the 6-pins for the debugger connector. As mentioned before, either the MPLAB ICD2 or the PICKit2 can be used for debugging/programming. Section B is used for the UART connector (RS-232). Fig. 2-32 shows the Jumpers 7 and 8 that provide the required pins: RX, TX and GND. Section C is the section used for CAN BUS communication. CAN BUS protocol needs two lines to communicate, denoted as CANH and CANL, Fig. 2-33 shows the pins needed for the CAN BUS module. Section D is a MCP2551 chip that changes the values of the signal to the standard of CAN BUS. Section E is the power input of the board. Section F is the PIC18F4580 itself. Section G is a set of Digital inputs/outputs. It contains ten pins that can be used as inputs or outputs, and 8 of them can be also used as analog inputs for the A/D converter. Here, a 10MHz crystal oscillator is used, that is, when using HSPLL settings, the user can reach 40MHz. Section I is a Reset Button. Section J and Section K are both used for the Operational Amplifier. Section L is a voltage regulator L7805 for producing a stable 5V voltage for the board.

Fig. 2-32 UART connector

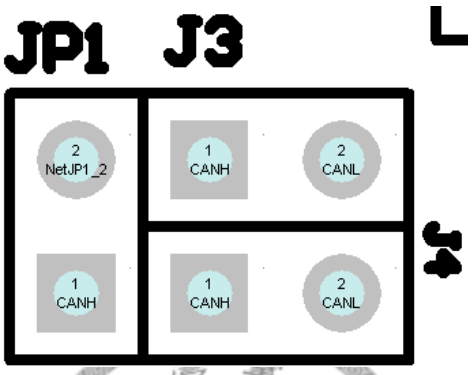


Fig. 2-33 CAN BUS connector

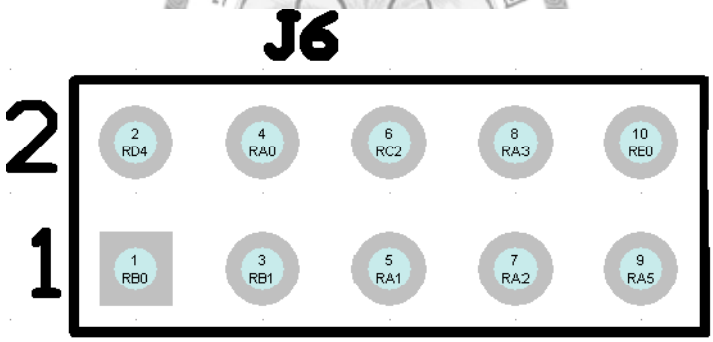


Fig. 2-34 I/O & Analog input

Chapter 3 Image Processing with Simulink

With Similink software, the image processing needed in this system can be easily done. The two major parts are color models transformation and correlation. The former one is used to transform the data transmitted from the camera into the format we want and also simplify the operation. The latter one is one of the template matching methods responsible for finding the most similar part of the image to the template.

3.1 Color Models Transformation

Color model is also called color space or color system. Since the oral description of color is no longer precise enough for reproducing color and the judgment of color varies from people and also affected by the environment illumination and other factors, a set of general accepted measurement of color has to be built. In that way, the color can be represented in a more scientific way [14]. In practice, colors are broken down into three portions. In the most common color model, colors are seen as variable combinations of the so-called primary colors RGB (red, green, blue). The amounts of red, green, and blue needed to form any particular color are called the tristimulus values and are denoted X, Y, and Z, respectively. A color is then specified by its trichromatic coefficients, defined as

$$x = \frac{X}{X + Y + Z} \tag{3.1}$$

$$y = \frac{Y}{X + Y + Z} \tag{3.2}$$

$$z = \frac{Z}{X + Y + Z} \tag{3.3}$$

Therefore, x, y, and z are in the range [0, 1], and with the relation that

$$x+y+z=1 \tag{3.4}$$

Based on RGB model, the color space can be viewed as the cube shown in Fig. 3-1, in which RGB primary values are at three corners; black is the origin; and white is at the corner at point (1, 1, 1), farthest from the origin. In this model, points along the main diagonal of equal RGB values have gray values. For convenience, the values of R, G, B have already been normalized using the equations shown before so that the cube in Fig. 3-1 is a unit cube.

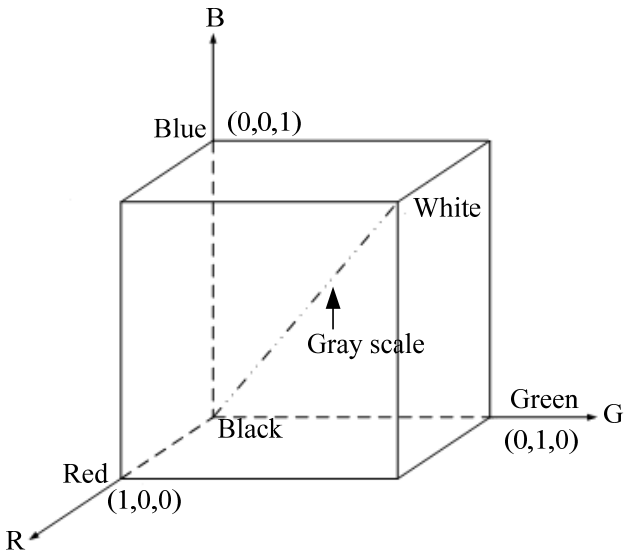


Fig. 3-1 Schematic of the RGB color cube

Color images represented in the RGB color model consist of three component images which combine on the screen to produce a composite color image. Fig. 3-2 shows the concept of the combination of the three component images. In addition to the RGB color model, there are still other color models, such as CMY (Cyan-Magenta-Yellow) and HSI (Hue-Saturation-Intensity).

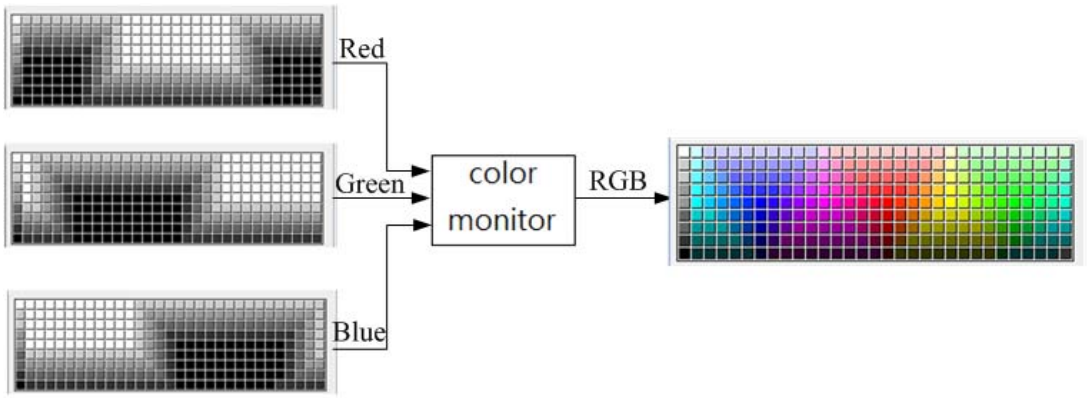


Fig. 3-2 Generating the color image by three component images

Hue represents dominant color as perceived by an observer. That is, when we tell what the color of the object is, we are referring to its hue. Saturation refers to the relative purity. The pure spectrum colors are fully saturated instead of added with white light. Hue and Saturation taken together are called chromaticity. The RGB, CMY, and other color models are not as intuitive for humans as HSI is. When humans view a color object, the color is characterized by its brightness and chromaticity. Therefore, we can get a conclusion that RGB is ideal for generating color, though HSI is ideal for developing image processing algorithms. As Fig. 3-3 shows, usually an angle of 0° from the red axis indicates 0 hue, and the hue increases counterclockwise from there and range [0, 360]. The saturation is the distance from the vertical axis. The vertical axis is defined as intensity which is the most important component of the HSI. Therefore, the HSI color model is usually showed based on circular color planes. The vertical axis of the color point indicates its intensity, the length of the vector from the origin to the point is the saturation, and the angle this vector makes with the red axis is defined as hue.

Each color space can be converted to the other. As mentioned before, the sample image taken by the camera is a color image and defined in RGB color space. However, the image processing used here doesn't need color information except the intensity which is sufficient for describing an object's contour, since it is enough for correlation.

Therefore, to make the operation simpler and much faster, the RGB color model was transferred to HSI and only the intensity information preserved [13].

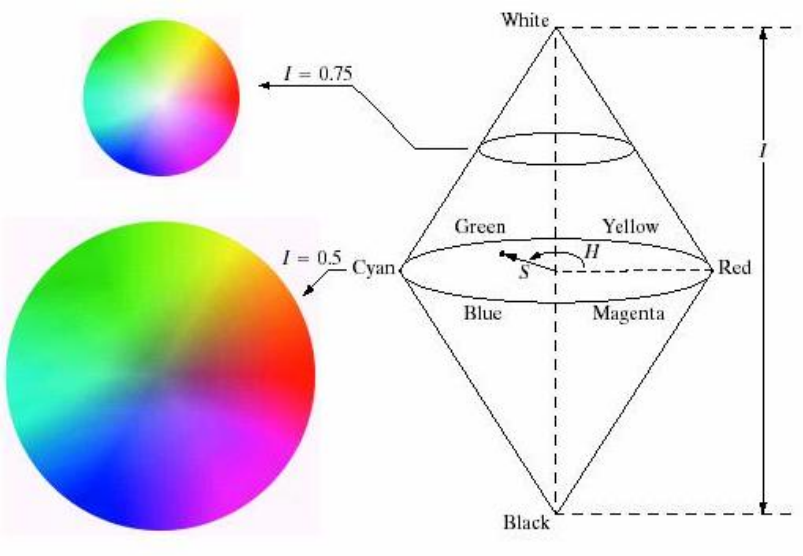


Fig. 3-3 The HSI color model based on circular color planes

Connected to the camera, the sample image comes in RGB color format and their H, S, I values are obtained using the equations below. Notice that the RGB values in these equations have been normalized to the range [0, 1]. Hue can be normalized to the range [0, 1] by dividing by 360°. The other two HSI components are already in this range since the given RGB values are in the interval [0, 1]. The toolbox of image processing provided by Simulink can help do the transformation, and make the operation easier.

$$H = \begin{cases} \theta & \text{if B} \le \text{G} \\ 360 - \theta & \text{if B} > \text{G} \end{cases} \tag{3.5}$$

with

$$\theta = \cos^{-1}\left\{ \frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}} \right\} \tag{3.6}$$

$$S = 1 - \frac{3}{(R+G+B)}\left[\min(R,G,B)\right] \tag{3.7}$$

$$I = \frac{1}{3}(R+G+B) \tag{3.8}$$

3.2 Matching by correlation

Object recognition is a great issue in image processing field and has a wide range of application. It has many ways to get the purpose. In this thesis, one of the methods called correlation is used. Correlation is the process of moving a filter mask over the image and computing the sum of products at each location [14]. The correlation of a mask $w(x,y)$ if size $Mb \times Nb$, with an image $f(x,y)$ may be expressed in the form [19]

$$C(x,y) = \sum_{s=0}^{(Ma-1)} \sum_{t=0}^{(Na-1)} w(s,t) f(x+s, y+t) \tag{3.9}$$

Where $0 \le i < Ma+Mb-1$ and $0 \le j < Na+Nb-1$. However, the preceding equation is sensitive to scale changes in f and w. Instead, the following normalized correlation coefficient is usually used [14].

$$\gamma(x,y) = \frac{\sum_s \sum_t \left[w(s,t) - \overline{w} \right] \sum_s \sum_t \left[f(x+s, y+t) - \overline{f}(x+s, y+t) \right]}{\left\{ \sum_s \sum_t \left[w(s,t) - \overline{w} \right]^2 \sum_s \sum_t \left[f(x+s, y+t) - \overline{f}(x+s, y+t) \right]^2 \right\}^{\frac{1}{2}}} \tag{3.10}$$

$\overline{w}$ is the average value of the mask, and $\overline{f}$ is the average value of f in the region coincident with w. Often, w is referred to as a template, and correlation process is referred to as template matching. After being normalized, γ has values in the range [-1, 1] and no longer changes its scale via the amplitude of w and f. The maximum value of γ occurs when the normalized w and the corresponding normalized region in f are identical. In other words, the maximum correlation coefficient indicates the best possible match between the image and the template.

Fig. 3-4 illustrates the mechanics of the procedure. The border around image

$f(x,y)$ is the padding necessary for the situation when the center of w is on the border of f. In template matching, the values of correlation when the center of the template is on the border of the image are concerned, so the padding is limited to half the mask width. By moving the center of the template, it can be made that the center of w visits every pixel in f.

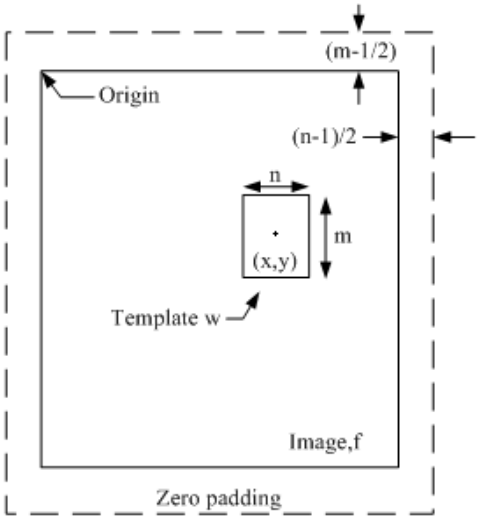


Fig. 3-4 The mechanics of template matching

At the end of the procedure, $\gamma(x,y)$ is obtained, and then we look for the maximum in $\gamma(x,y)$ to find where the best match occurred. It is possible to find multiple locations in $\gamma(x,y)$ with the same maximum value.

All this procedure can be worked in Simulink using its toolbox [19]. The inner block diagram of the subsystem with purple frame showed in Fig. 3-8 and Fig. 3-13 is below. Looking at the figure, In1 of 2-D Correlation block is the image while the In2 is the template. Each of them has to be a matrix with dimensions of $M \times \mathrm{N} \times 1$. Therefore, using the intensity of the image as the correlation input is reasonable and appropriate. After the 2-D Correlation, we need a Maximum block to find where the maximum of the output matrix is. This indictaes the position of the best possible

match (i.e., the coordinate of the maximum value).

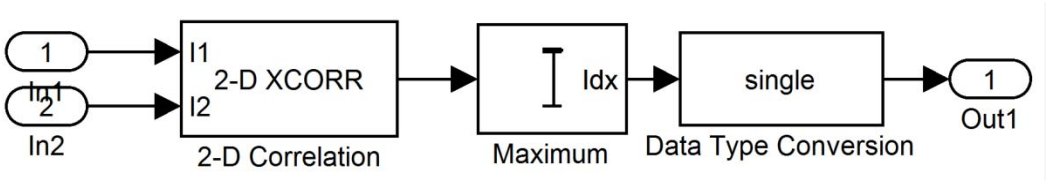
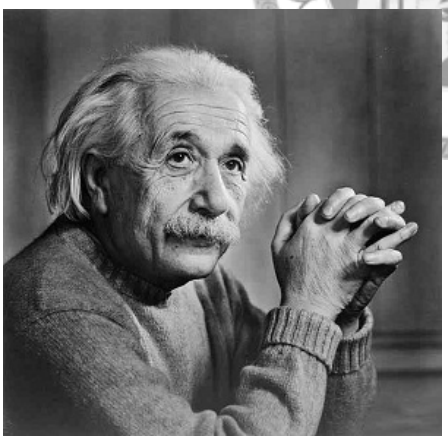


Fig. 3-5 correlation subsystem

Fig. 3-6(a) shows a famous image of Einstein with a size of 284×280 pixels. As an example of correlation, the right eye of Einstein is selected to be the template It is a small sub image (17×13) from Fig. 3-6(a). Fig. 3-6(c) shows the result of computing the correlation coefficient. Intensity in this image is proportional to correlation value in the range [0, 1]. In other words, the brighter the point is the better match the location is. The brightest point of the correlation image is clearly visible near Einstein's right eye. Fig. 3-6(d) shows the image with a white plus at the location of the maximum correlation coefficient which corresponds closely at the location of the right eye in Fig. 3-6(a).

a)



b)

c)

d)

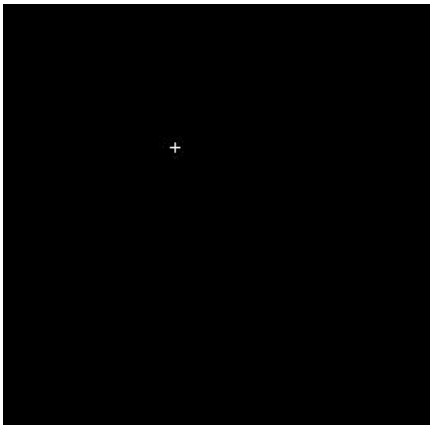


Fig. 3-6 (a) Image of Einstein (b) Template of Einstein's right eye

(c) Correlation coefficient shown as an image (d) Location of the best match.

3.3 Communication with MCU

Instrument Control Toolbox from Simulink allows computer to communicate with other devices. Once the Simulink finishes the image processing and has the coordinates where the best match occurred, it transferrs the result through RS232. All the parameters of the serial transmission can be set in the dialog box. However, the data length that can be transferred by RS232 is only 8bits, and the first bit usually stands for whether the data is positive or negative. To prevent miss-transmissions, some transformations have been done. Fig. 3-7 shows the process of data transforming subsystem which is the content of the Create Usart output block in Fig. 3-8 and Fig. 3-13. The coordinate of the best match position is transferred as a 2×1 matrix. After demuxing the matrix, it splits into two parts, one is the y-index, and the other one is the x-index. However, the range of y and x is determined by the camera's resolution. To make it more flexible and be able to accept higher resolutions, the value of y or x are divided into two parts with the relation of

$$\begin{aligned} y &= 8\times y_1 + y_2 \\ x &= 8\times x_1 + x_2 \end{aligned} \tag{3.11}$$

Therefore, there is little chance that data exceeds 127, that is, the eighth bit is hardly used and neither confronts the dilemma that data turns into negative. From Fig. 3-7, it can be seen that a set of data is consisted of five numbers. The final number is set to be a constant 127 which is regarded as a head data so that when MCU receives the number 127, it can recognize that the following four numbers are $y1, y2, x1, x2$, respectively.

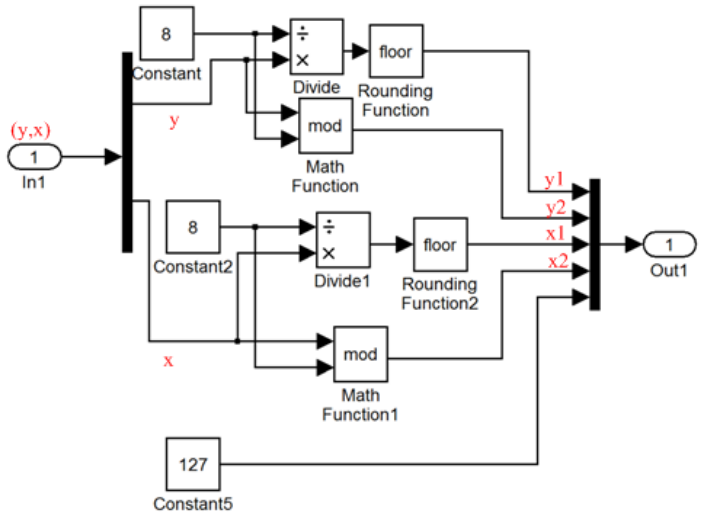


Fig. 3-7 Creating the data for USART transmission

3.4 System architecture

After introducing all the hardware and the image processing methods, this section shows how they work together in this eye tracking system.

As mentioned before, the goal of the project is to turn the two cameras to the direction where the human eye is looking at, and furthermore, to focus on the same object. It can be achieved by Simulink software by Matlab, which provides a Video and Image Processing Blockset and Instrument Control Toolbox. Some of the blocks from these two toolboxs are used here; therefore, users can simply build a new model by these blocks instead of through programming. The entire image processing procedure is developed in two parts. The first part is showed in Fig. 3-8. The block "USB 2863 Device YUY2 176 x144 composite" can acquire a live image from the image acquisition device the introduced ACV-566F camera. This is used to bring the image data into the model. The block allows users to configure and preview the video image. In this case, the output port is selected to be one multidimensional signal instead of three output ports corresponding to the three color bands, red, green, and

blue. The image acquisition rate, which means the speed at which frames are delivered to this block during simulation is also specified in this block.

After the video image data is brought into the model, the output color model data is converted in order to simplify the operation. The amount of data will be reduced to one third compared to the previous data.. After that, the data and the template are sent to the correlation block as two matrices to find the most similar part in each sample image.

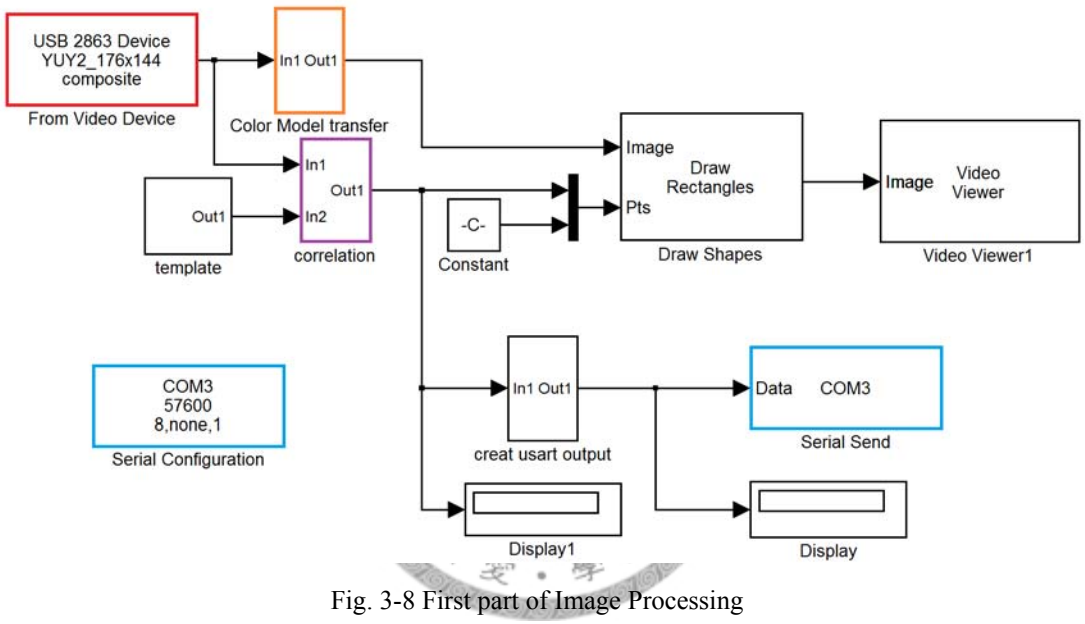


Fig. 3-8 First part of Image Processing

Since the model is planned to extract the eyeball from the image, the template is set to the figure as Fig. 3-9(d). Looking at Fig. 3-10, it is not hard to notice the light reflection in the black area of eyeball. Therefore, considering this phenomenon, the template is not a simple black circlelike Fig. 3-9(b). Instead, there is a black point in the light reflection area. According to the rolling angle of the eye and the ambient illumination at that time, the image may be a little different, but generally, it can be viewed as Fig. 3-9(d). The difference between Fig. 3-9(c) and Fig. 3-9(d) is the background, which is filled up with color white in the remaining areas in (c) and maintained transparent in (d). By checking out the values of white color and transparent color in Simulink, these two colors are different and after comparing their

effect on correlation, the transparent background is apparently better.

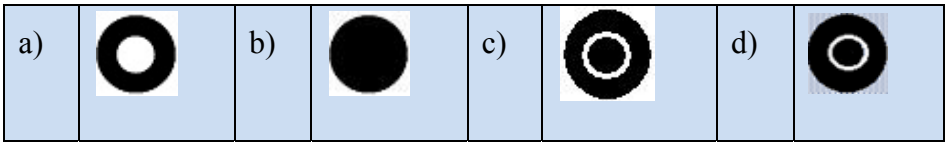


Fig. 3-9 Template samples in First part of Image Processing

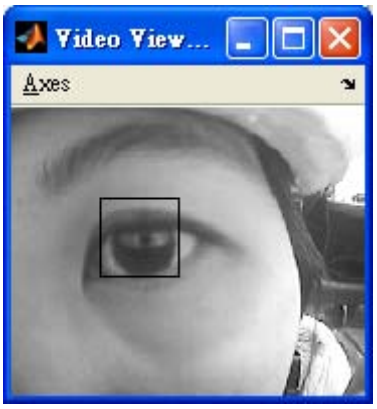


Fig. 3-10 Outcome Video Image from First part of Image Processing

After deciding the template, the correlation block in Simulink will help to do the correlation operation and export the result matrix. The correlation theorem is introduced in section3.2. As showed in Fig. 3-8, the maximum of the correlation result matrix can be found by another Simulink block. Briefly speaking, the maximum is exactly the most similar part with the template in each sample image, that is, the position of the eye at that time.

The first part of image processing has so far obtained the position of the eye in the real time image. The last step in image processing is to transfer the position of the eye to the microcontroller. A toolbox of serial transmission mentioned before supported by Simulink lets us simply specify the configurations.

When the first PIC receives the position of the eye from this first part of image processing, through some reasonable calculations, the first PIC produces the appropriate PWM signal to drive the camera to the right direction where the eye is supposed to be looking at. At the same time, the first PIC also transfers the data to the second PIC through CAN-BUS. In that way, the second PIC can produce the same

PWM signal as the first PIC to drive the second camera to the same direction. Up to this step, the two cameras rotate towards the same direction, but they do not focus the same target yet. The schematic diagram is shown below in Fig. 3-11.The arrows represent the directions of these two cameras.

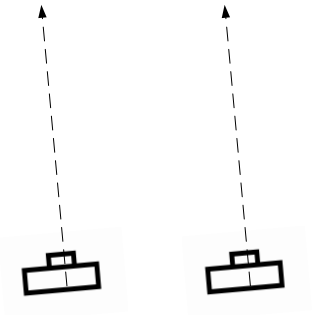


Fig. 3-11 Schematic Diagram of the eyesight direction after the first part of image processing

It's easy to understand that the images taken by the two cameras are definitely different. For example, Fig. 3-12 shows the images taken by the two cameras towards the same direction. Fig. 3-12(a) is from the left camera and Fig. 3-12(b) is from the right camera. The square in the center of Fig. 3-12(a) represents the center of the sight. Assuming that when we are looking at something, it is showed in the center of the sight. Therefore, the center area framed by a square in Fig. 3-12(a) can be viewed as the center of the sight, and the right camera is supposed to turn towards the same thing instead of only turning to the same direction.

a) b)

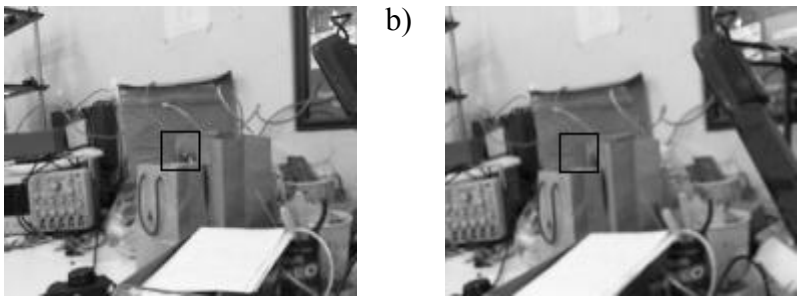


Fig. 3-12 The Images took by the two cameras positioned like Fig.3-4

To achieve this purpose, the second part of image processing is needed. The block diagram is shown in Fig. 3-13 is very similar to Fig. 3-8. The difference is the

template of correlation, which is a figure of an eye sample in Fig. 3-8 but a submatrix of the image from the left camera in Fig. 3-13. By selecting this template, the second part of image processing is going to be able to find the same target appearing in the other camera. Therefore, until the same target appears in these two cameras, the coordinates will be continuously transferred to the second PIC18F4580 in order to check the direction to which the right camera turned.

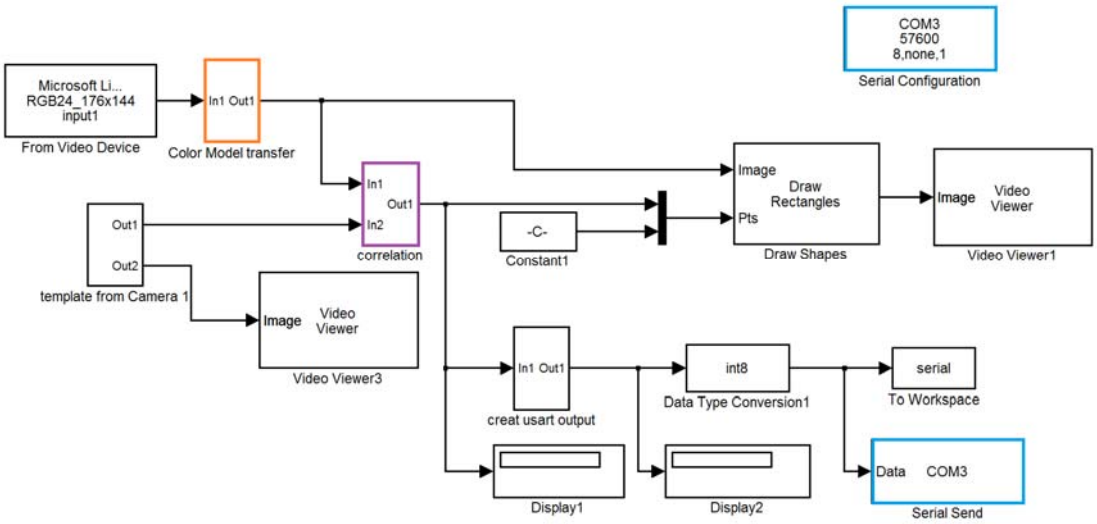


Fig. 3-13 Second part of Image Processing

Fig. 3-14 shows the concept of these two cameras' movement. From Fig. 3-14, it can be easily seen that the directions of these two cameras are no longer the same, but with the same target appearing in the center of the image taken by each camera. In that way, the two cameras can be viewed as the behavior of human eyes when focusing an object. Fig. 3-15 shows the images viewed from these two cameras positioned like in Fig. 3-14. Compared to Fig. 3-12, the square is moved to the center, meaning that the two cameras are focused on the same target, which is exactly the object framed by the square. Up to this step, the whole image processing was done and none of these two parts of image processing can be neglected. The first part is to make sure the object at the center of the first camera is also showed in the sight of the second camera. Thus, when carrying out the second part, the image processing can find the

object and turn the object to the center. Experiments and discussions are shown in Chapter 4.

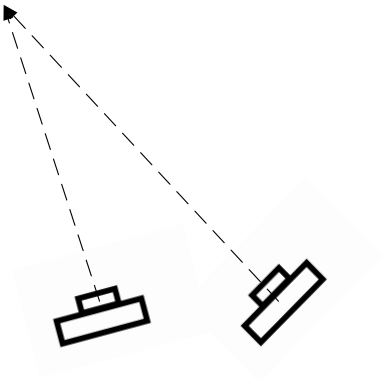


Fig. 3-14 Schematic Diagram (after the second part of image processing)

a) b)

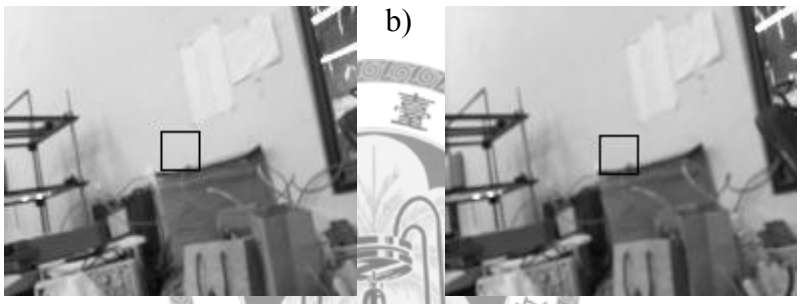


Fig. 3-15 The Images took by the two cameras positioned like Fig.3-7

After discussing the detailed image processing procedure, the control procedure of the whole system will be discussed. The flow chart showed in Fig. 3-16 is the control procedure of the entire eye tracking system. The words in blue identify the information transmitted while the words in red identify which communication way is used. The blocks marked Image Processing I and Image Processing II were already discussed before. At the beginning of the procedure, the image comes from the fixed-camera. This camera has the highest priority to control the entire system. After the image passes through Image Processing I, the position of eye has been found and sent to the first microcontroller. Once the new coordinates of the eye are received by the first PIC, it transfers the data into the corresponding angular position and uses the aforementioned way to produce the PWM signal to the servo motors, responsible for

controlling the rotational motion of the left camera. At the same time, it also transmits the computed angular position to the second microcontroller. As showed in Fig. 3-16, the second PIC has two inputs. It is important to decide which has a higher priority to determine the duty cycle of the PWM signal.

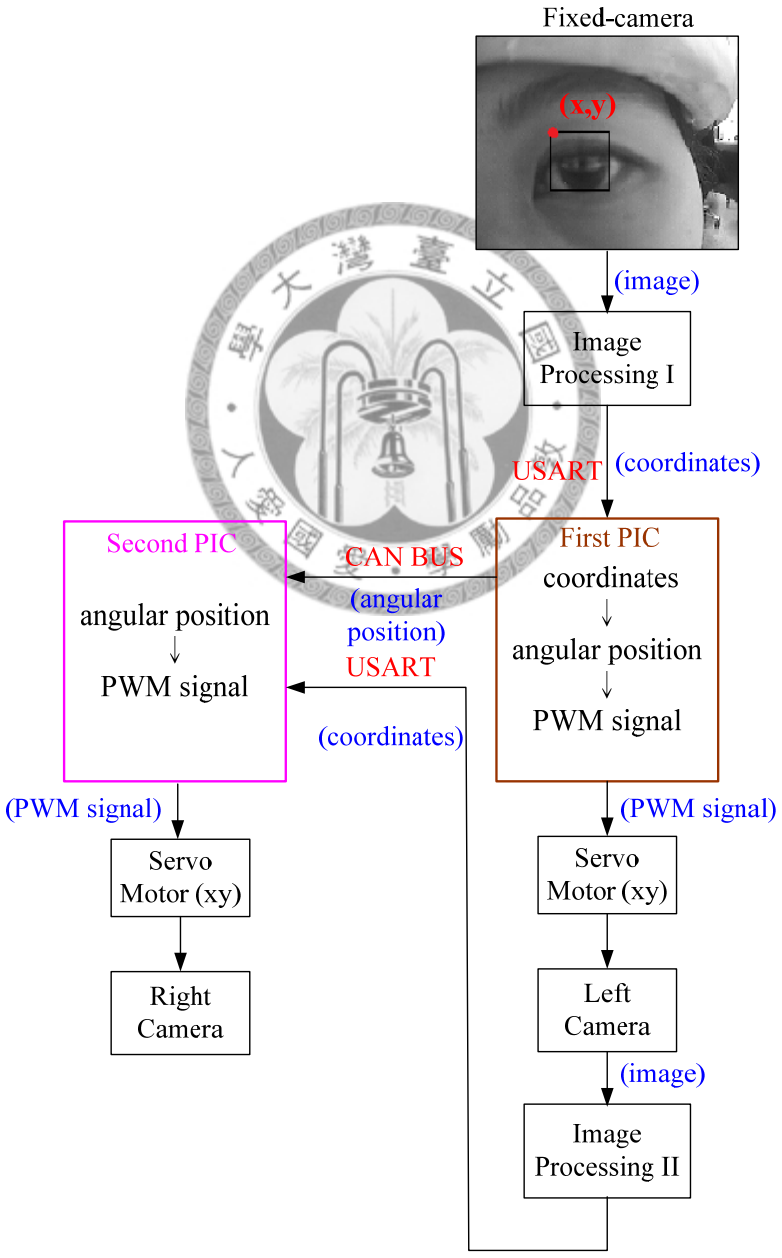


Fig. 3-16 Control Procedure of the entire system

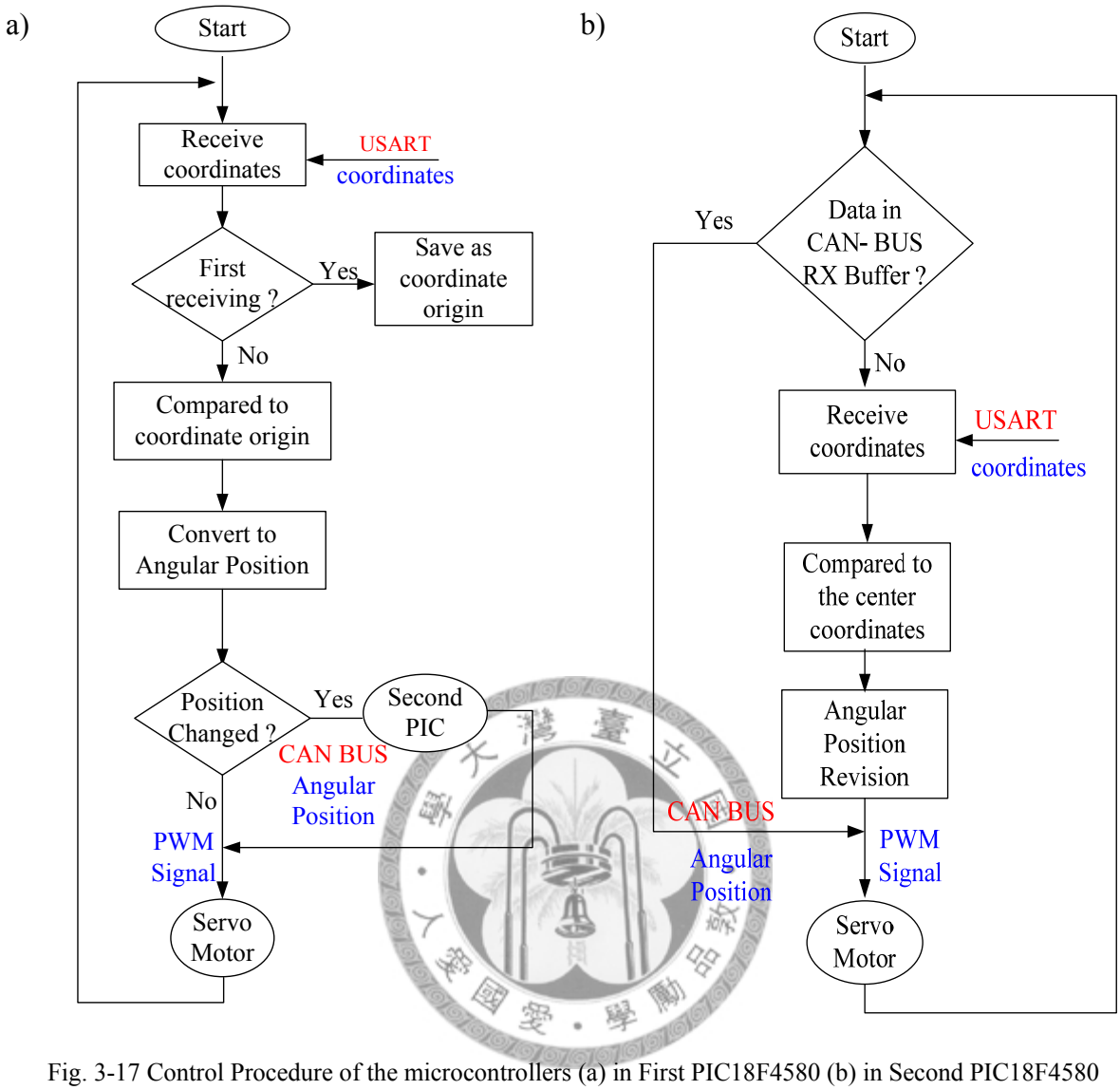


Fig. 3-17 Control Procedure of the microcontrollers (a) in First PIC18F4580 (b) in Second PIC18F4580

Fig. 3-17 shows the inside control procedure of the two microcontrollers. It clearly explains the priority decision of the two inputs in the second microcontroller. The arrangement is decided after considering the real motion of the human eyes. In actual situations, two eyes tend to turn to the same general direction no matter if the two eyes are focused or not. Therefore, whenever the first PIC18F4580 transmits the new angular position through CAN-BUS, the second PIC18F4580 should immediately produce the corresponding PWM signal in order to make the camera on the right follow the left one. After turning to the general direction, the camera on the right would feedback its image to the Image Processing II to revise the position of the

other camera. More detail is discussed in the next chapter; it only has to be understood that the right camera keeps revising its angular position until having the target showed in the center of the image is exactly the same as the one shown in the left camera.

Chapter 4 Experimental Results and Discussion

After introducing all the details of the eye tracking system, the system is operated in practice and tested. This chapter shows the experimental environment, how the experiments are conducted and finally, the experiments results are discussed. Based on the idea of getting distance between the gazing target and the eye, and expecting a correctly designed eye tracking system with high efficiency, the experiments are carried out with the purpose of attempting to get the distance of the gaze point using the experiment data.

The distance estimated by the experimental data has been compared to the real distances and results are discussed.

4.1 Fundamentals of Experiments

Before any experiment is carried out, some restrictions and instructions have to be mentioned and explained. Since the tracking system is mounted on a helmet to help fixing the relative position between the system and the human eye, the helmet needs to be worn identically every time for a better accuracy in experiment results, even though the original coordinates of the eye's position is reset every time when the tracking system starts.

For the purpose of estimating the distance from the tracking system to the target, the subject is told to look at a set of specific directions. After checking the eye movement can exactly control the first camera movement, the experiments are ready to be executed.

The angular position of the cameras are defined as in Fig. 4-1. Angular position was defined as zero degree when looking straight forward, no matter in horizontal or in vertical direction. Using the x-y coordinate system to represent the horizontal and vertical directions, respectively, when looking up or down, the angle was referred to

positive and negative degree in the vertical direction (i.e. y-direction). If turned to right or left, the angle was referred to positive or negative in the horizontal direction (i.e. x-direction). To clearly illustrate, Fig. 4-1(a) is the side view of the mechanism while Fig. 4-1(b) is the top view of only the module.

In order to get the current angular position of the servo motor, USART is used again to get the information of the second camera. Because the second camera goes through correlation processing, its angular position is not the same as the first camera. This is used to estimate the distance from target to the camera. The idea is that, the closer the target is the larger the horizontal angular position difference would be. Besides, the position in vertical direction should be quiet similar. The following experiments allows us verify these ideas. The feasibility and geometry is explained later.

To reduce the influence of transmission error, some prevention has been taken. Adding two different previously known numbers before each of the angular position data allows the receiver to recognize which one represents horizontal degree and which one represents vertical. In the following experiments, 127 and 126 are chosen, thus, a data formation in a set of four numbers is established. As soon as one of the four numbers is lost during transmission, this set is discarded.

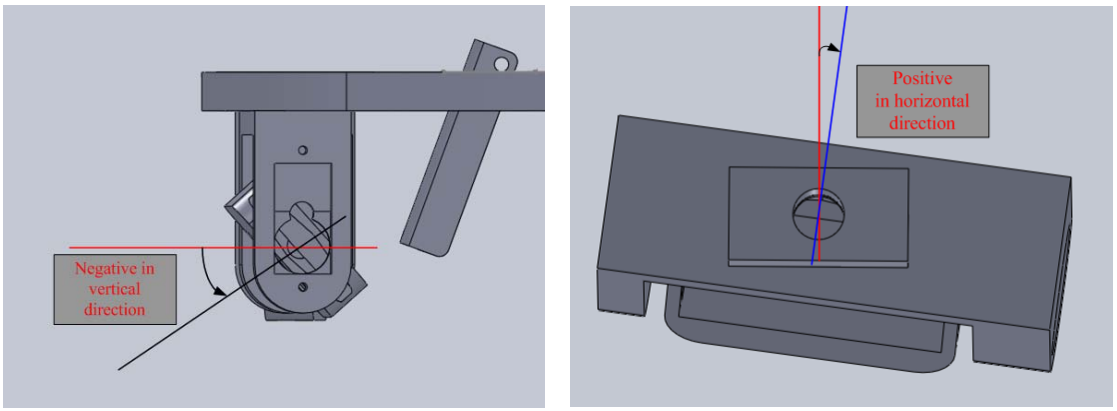


Fig. 4-1 Definition of the sign in two directions

4.2 1-D Experiments

4.2.1Gait Control

Experiments are carried out when the gazing point is along the x-axis just like horizontal saccade. To simulate the motion, the angular positions were given as

X direction angle： 0° -40° -30° -20° -10° 0° 10° 20° 30° 40°

Y direction angle： Maintained at zero degree

To see if the second camera followed the first camera, the trajectory was recorded in x and y direction separately. The range in the x-direction is because ±40° is the most similar to the human eye's range. Customarily, we don't attempt to overturn our eyes; instead, we would more likely turn our head.

After experiments that simulate the saccade in horizontal direction, experiments in vertical direction are also carried out with the arrangement of the following angular position series. A sample time of 20ms is used, which means that 50 samples are taken per second.

X direction angle： Maintained at zero degree

Y direction angle： 0°-30° -20° -10° 0° 10° 20° 30°

4.2.2Results and Discussion

- Test 1

As the experimental result shows in Fig. 4-2, the angle change of the right camera in y-direction is obviously small, nearly just chattering around zero. The result is pretty reasonable as expected. On the contrary, the angle in x-direction is more interesting. Take a notice that whenever the left camera is set to turn left (i.e.in negative angle region), the right camera has almost the same angle or an even bigger angle in order to show the same target in the center of the image. However, if the left camera turns right, the angle does not turn the same anymore. Take a look at ±40°,

where this phenomenon is most evident. This phenomenon can be clearly explained using Fig. 4-3. As the left camera turns the same angle θ to the left or to the right, the right camera will turn to different angles α and β towards the same direction of the left camera. This figure shows that α, β and θ have a relation of $\alpha > \theta > \beta$. Therefore, the experiment result is reasonable.

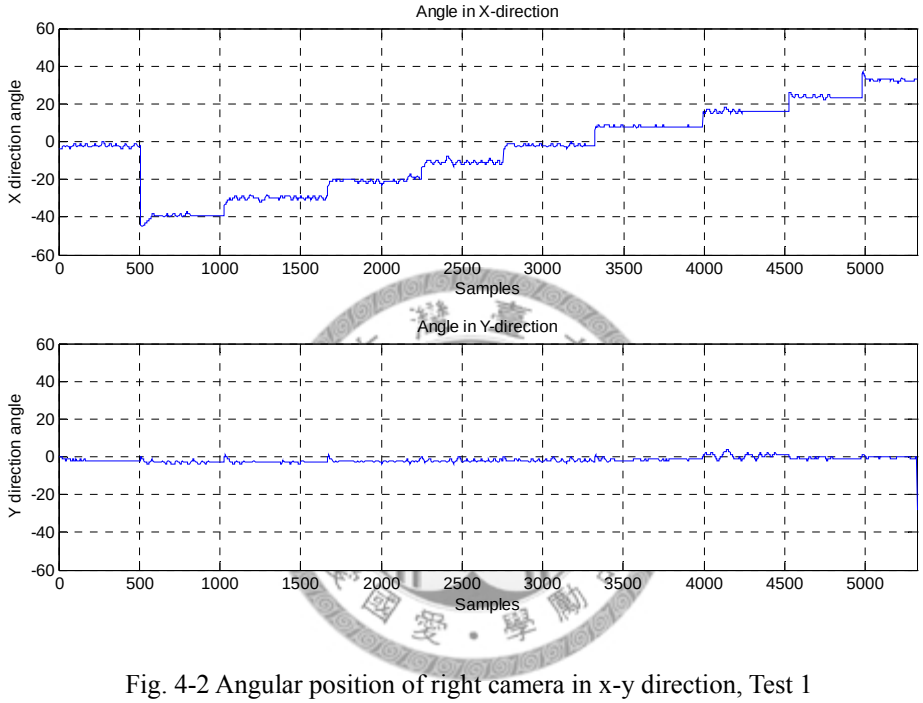


Fig. 4-2 Angular position of right camera in x-y direction, Test 1

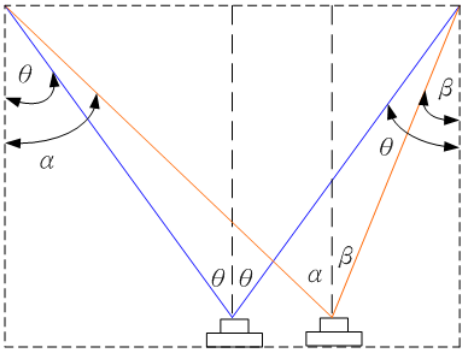


Fig. 4-3 Geometry Concept in Test 1

- Test 2

Unlike the result of Test 1, the angle change in the y-direction of the left camera would also be the angle of the right camera, instead of having a clear difference.

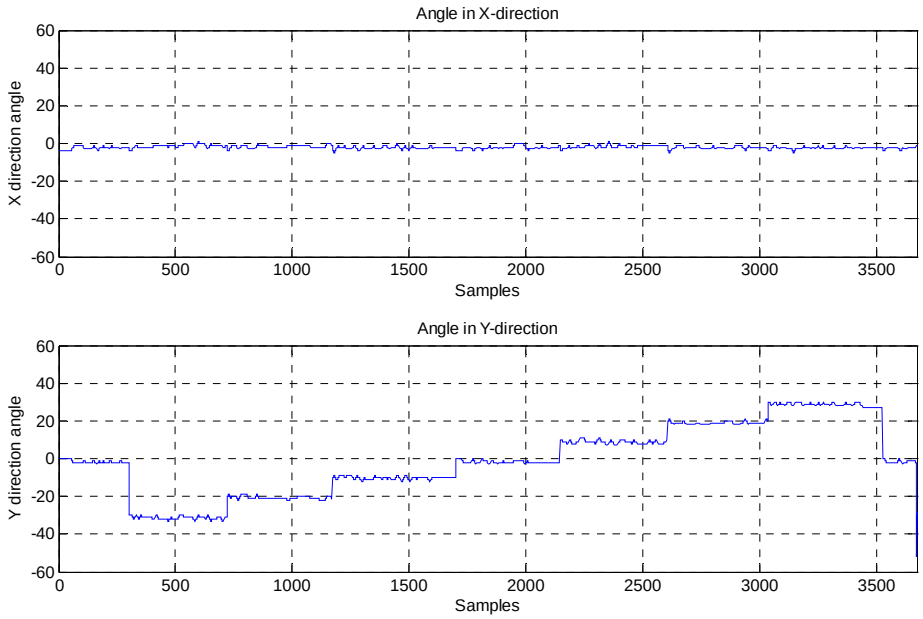


Fig. 4-4 Angular position of right camera in x-y direction, Test 2

4.3 2-D Experiments

4.3.1 Gait Control

For the following three experiments, each of them has different arrangements but for the same purpose of calculating the object's distance. As previous experiments, they still have the same goal to calculate the object's distance by feedback data. However, unlike the 1-D experiments, the followings have motion in x and y directions.

In Test 3, the relation between object's distance and the angle in x direction is of interest. Thus, the object is put at a distance of 50cm at first, then moved backward to 300cm and again back to 50cm in steps of 50cm. From the feedback data of the angle in x direction of the second camera, the object's distance is calculated.

In Test 4, the distance to the perpendicular plane where the object is located is already known to be 113 cm. However, in this experiment, the plane's distance is calculated by the feedback data instead. After the plane's distance has been calculated, the value can be further used with the angle in y direction to figure out the height of

the target from the horizontal position in order to calculate the object's distance. The plane's distance can be calculated using the angles obtained when positioned at the origin.

Test 5 is the last experiment, where random angular positions in real environment were taken. The real object's distance can be compared to the value calculated from the feedback data. The geometry concepts of these experiments are shown below with their results.

4.3.2Results and Discussion

- Test 3

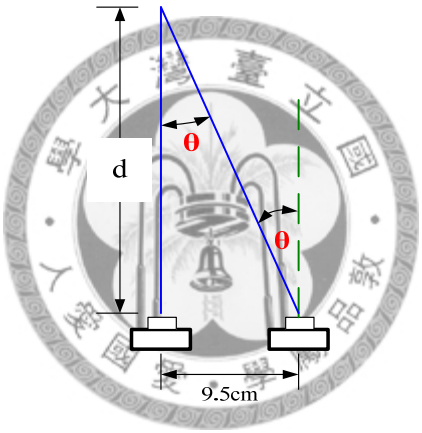


Fig. 4-5 Geometric Concept in Test 3

$$\frac{9.5}{\sin\theta} = \frac{d}{\cos\theta} \tag{3.12}$$

$$d = 9.5\cot\theta \tag{3.13}$$

From Fig. 4-5, it can be understood that when d becomes larger, the value of θ becomes smaller. To a certain value of d, the changing of θ would be covered by the error. The distance between the two cameras and the angle θ have the relation showed in the sketch. From the last equation and the measured value of θ, the object's distance marked as d in the sketch can be calculated. From Table 4-1, the angle in the x-direction is the value used as θ to operate. Besides paying attention to the angle in the x-direction, the angle in the y-direction seldom changes during the experiment

processing. It is similar to the results in Test 1, where the target moved along the x-direction.

To verify the idea is workable, the calculated distance in the table is compared to the given object's distance. As shown by the error, the farther the object is, the larger the error. The low resolution of the camera is a reason. By increasing the resolution, the same image can be divided in finer grid lines and the coordinates can be more accurate, and consequently, the angle in x-direction.

Table 4-1 Test 3 result

Object's distance (cm)	50	100	150	200	300
Angle in x direction (rad)	-12	-6.1037	-4.3080	-3.1151	-2.2222
Angle in y direction (rad)	-2	-1.9170	-1.8152	-1.8849	-1.8148
Calculated Distance (cm)	44.6940	88.8389	126.1113	174.5611	244.8166
Error %	10.612	11.1611	15.9258	12.7194	18.3945
Object's distance (cm)	200	150	100	50	
Angle in x direction (rad)	-3.0984	-3.8750	-5.8736	-11.0227	
Angle in y direction (rad)	-1.9606	-2	-2	-2.0227	
Calculated Distance (cm)	175.5018	140.2528	92.3450	48.7700	
Error %	12.2491	6.4981	7.6550	2.4601	

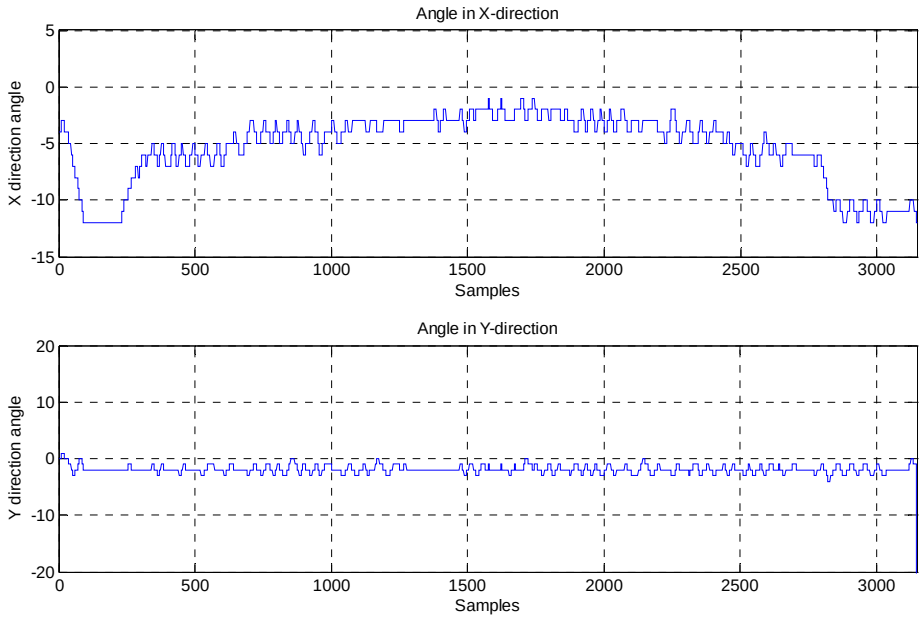


Fig. 4-6 Angular position of right camera in x-y direction, Test 3

- Test 4

The geometric concept of Test 4 is showed in Fig. 4-7. The two squares represent the two cameras. This figure took the angular position of (0°, 20°) as example to illustrate the way how the depth and distance can be figured out. The left camera angular position (β_x, β_y) is the input and α_x , α_y are what were received from USART. With the four angles, depth can be calculated from equation(3.15). As mentioned before, the depth is calculated using data where $(\beta_x, \beta_y) = (0, 0)$ and corresponding α_x is 0.0826. Therefore, the calculated depth is 114.7990cm, which is similar to the actual depth 113cm. The value of α , used to get the results shown in Table 4-2 was an average of each period as shown in Fig. 4-8.

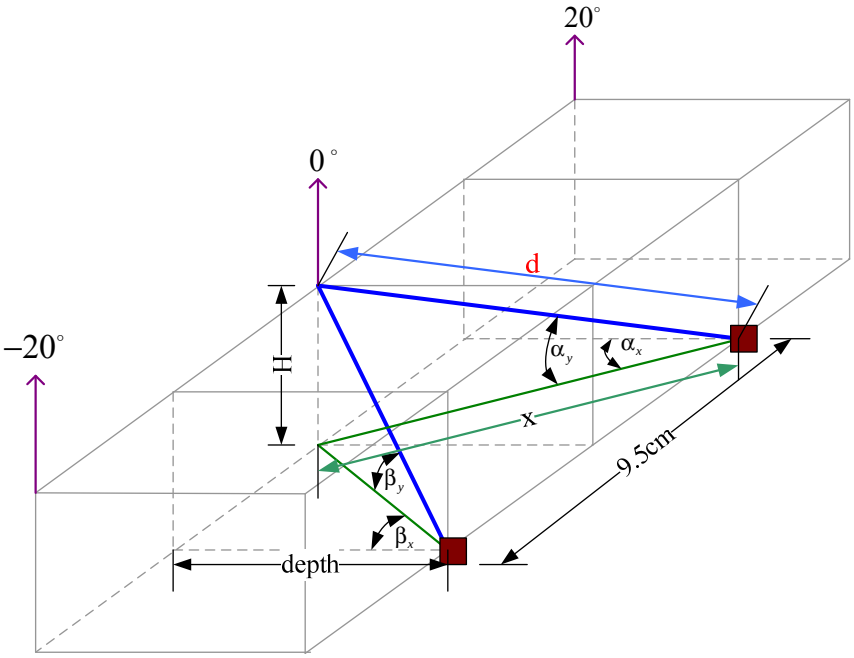


Fig. 4-7 Geometric Concept in Test 4

$$depth \times \tan\alpha_x + depth \times \tan\beta_x = 9.5 \tag{3.14}$$

$$depth = \frac{9.5}{\tan\alpha_x + \tan\beta_x} \tag{3.15}$$

$$d = \frac{depth}{\cos\alpha_x \times \cos\alpha_y} = \frac{depth}{\cos\beta_x \times \cos\beta_y} \tag{3.16}$$

$$H = d\sin\alpha_y = d\sin\beta_y \tag{3.17}$$

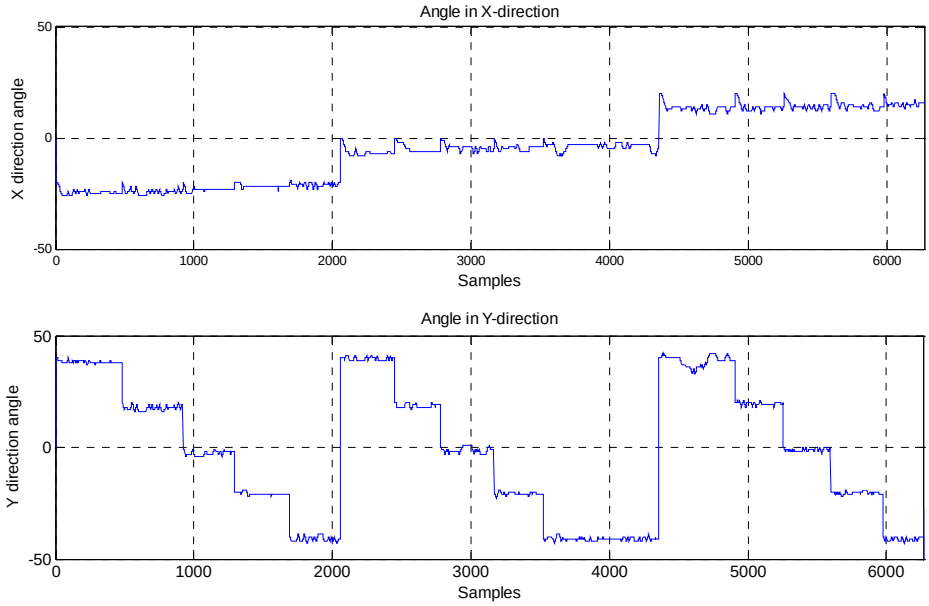


Fig. 4-8 Angular position of right camera in x-y direction, Test 4

Table 4-2 Test 4 result of distance

(β_x, β_y)	(-20°, 40°)	(-20°, 20°)	(-20°, 0°)	(-20°, -20°)	(-20°, -40°)
theoretical d	156.9780	127.9696	120.2521	127.9696	156.9780
calculated d	160.8557	132.2262	124.8802	132.4017	162.8708
Error d %	2.4703	3.3263	3.8487	3.4634	3.7540
(β_x, β_y)	(0°, 40°)	(0°, 20°)	(0°, 0°)	(0°, -20°)	(0°, -40°)
theoretical d	147.5110	120.2521	113	120.2521	147.5110
calculated d	150.0531	121.6217	115.2234	123.1134	152.2785
Error %	1.7233	1.1389	1.9676	2.3795	3.2319
(β_x, β_y)	(20°, 40°)	(20°, 20°)	(20°, 0°)	(20°, -20°)	(20°, -40°)
theoretical d	156.9780	127.9696	120.2521	127.9696	156.9780
calculated d	150.3610	124.9606	118.1981	126.4189	157.8839
Error %	4.2152	2.3514	7.7081	1.2118	0.5771

The value of original d is calculated from the equation $d = \frac{113}{\cos\beta_x \times \cos\beta_y}$.The depth is replaced with the actual depth 113cm, and only β_x and β_y are used. On the other hand, the value of "estimated d" is calculated from the equation $d = \frac{depth}{\cos\alpha_x \times \cos\alpha_y}$ and only α_x and α_y are used. The errors are acceptable. After calculating the object's distance, the vertical distance H is calculated from the horizontal plane using the calculated d and the measured angles. As the results show in Table 4-3, the calculated H and the theoretical H are very similar. The fields denoted as "N/A" correspond to a theoretical value of H=0, which means that the error is always infinity, having no meaning to our purposes.

Table 4-3 Test 4 result of H

(β_x, β_y)	(-20°, 40°)	(-20°, 20°)	(-20°, 0°)	(-20°, -20°)	(-20°, -40°)
theoretical H	100.9035	43.7682	0	43.7682	100.9035
calculated H	99.8034	39.6980	6.0934	47.0115	106.6169
Error H %	1.0903	9.2994	N/A	7.4102	5.6622
(β_x, β_y)	(0°, 40°)	(0°, 20°)	(0°, 0°)	(0°, -20°)	(0°, -40°)
theoretical H	94.8183	41.1286	0	41.1286	94.8183
calculated H	95.5703	38.6387	2.7150	43.4531	99.7703
Error H %	0.7932	6.0541	N/A	5.6517	5.2226
(β_x, β_y)	(20°, 40°)	(20°, 20°)	(20°, 0°)	(20°, -20°)	(20°, -40°)
theoretical H	100.9035	43.7682	0	43.7682	100.9035
calculated H	93.3706	41.3513	2.6309	43.9622	103.8762
Error H %	7.4654	5.5219	N/A	0.4434	2.9461

- Test 5

The experiment is no longer carried out in a prearrangement environment; instead, it was executed in a random environment. The concept of geometry is shown in Fig. 4-9. The value of actual d in Table 4-4 is manual measured.

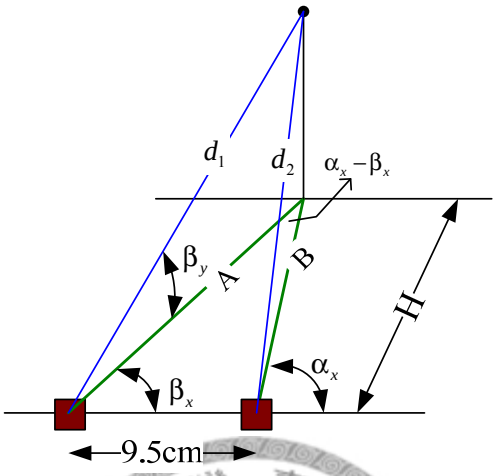


Fig. 4-9 Geometric concept in Test 5

$$\frac{9.5}{\sin(\alpha_x - \beta_x)} = \frac{A}{\sin(180 - \alpha_x)} = \frac{B}{\sin\beta_x} \tag{3.18}$$

$$d_1 = \frac{A}{\cos\beta_y}; d_2 = \frac{B}{\cos\alpha_y} \tag{3.19}$$

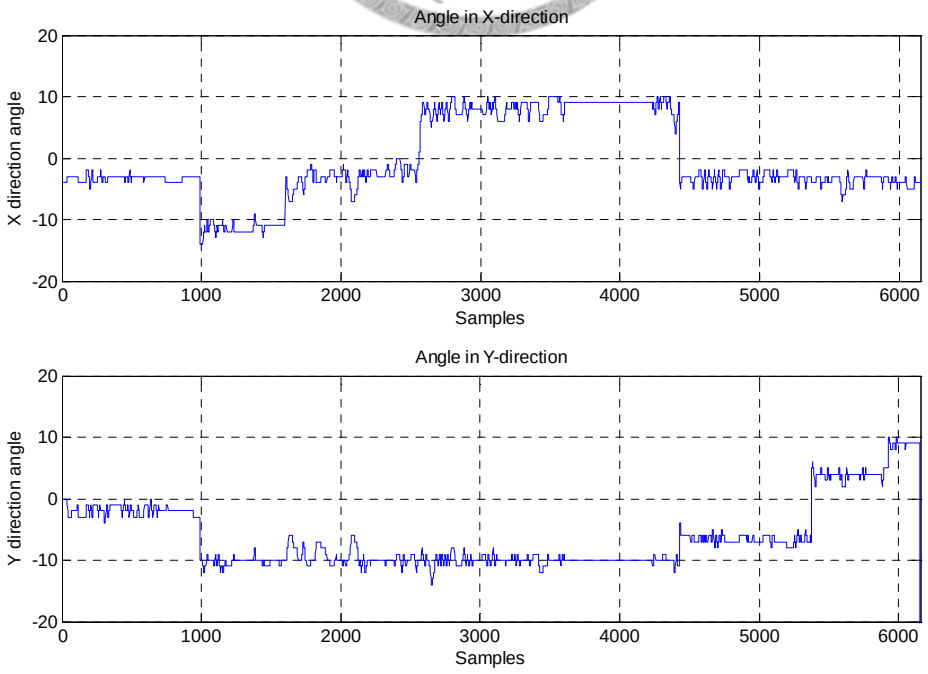


Fig. 4-10 Angular position of right camera in x-y direction, Test 5

Since 9.5cm is relative smaller than d_1 and d_2, it can assumed to be the same in value as shown in Fig. 4-9

The errors 1 and 2 at $(\beta_x, \beta_y) = (-10°, -10°)$ are unreasonable large. The reason may be the resolution of the camera and the slight angle. Due to the very small angle, it is easily covered by the chattering and error.

Table 4-4 Test 5 result

(β_x, β_y)	(0°, 0°)	(-10°, -10°)	(0°, -10°)	(10°, -10°)
Actual d (cm)	186	186	184	192
d_1 (cm)	164.7326	400.2679	193.7214	331.7216
Error1 %	11.4341	115.1978	5.2834	72.7717
d_2 (cm)	165.0926	402.3491	193.7260	330.2294
Error2 %	11.2405	116.3167	5.2859	71.9945
(β_x, β_y)	(0°, -5°)	(0°, 5°)	(0°, 10°)	
Actual d(cm)	186	187	188	
d_1 (cm)	178.1542	152.3997	127.7909	
Error1 %	4.2182	18.5028	32.0261	
d_2 (cm)	178.9517	152.4579	127.7807	
Error2 %	3.7894	18.4717	32.0316	

- Test 6

Test 6 is also executed in a random environment like Test 5. The difference is that Test 6 is focused on some gazing points with peculiar conditions, for example: much closer to the human eye or with large deflection angle. The geometric concept is

shown in Fig. 4-9. The actual d in Table 4-5 is manual measured and the calculated d is calculated from the equation (3.18) and equation (3.19). Gazing point 1 and 4 are trying to confirm whether the device is suitable or not with such close distance. The others are trying to confirm if the camera can also get focused when the target is at the position with relatively bigger α and β.

Table 4-5 Test 6 result

gazing point	1	2	3	4	5
(α_x , α_y)	(112°, 1°)	(135°, 0°)	(40°, 0°)	(102°,-32°)	(99°, -29°)
(β_x , β_y)	(90°, 1°)	(132°, 0°)	(46°, 0°)	(86°, -30°)	(94°, -20°)
actual d	20	108	67	41	128
calculated d	23.513	128.354	58.419	38.928	114.57
Error d %	17.56	18.85	12.81	5.05	10.49

As the error percentage showed above, the ability of getting focused on the gazing point with these particular conditions is also great.

Chapter 5 Conclusions and Future Work

5.1 Conclusions

This thesis presents the design of an eye tracking system mounted on a helmet. A mechanism is developed and using a microcontroller, the PIC18F4580 by Microchip, the actuators can be controlled to move the camera to the desired direction. Two kinds of cameras are used as inputs and outputs of the system.

Two cameras have two degrees of freedom to rotate and a third camera, which is fixed, is used to capture the eye's image. The two rotating cameras are capable of focusing on the same target and the system, using the data of the angles after focused, is capable of calculating the focus distance.

A communication method through CAN BUS that allows the two cameras to turn towards the same direction prior to the focus process is established. This arrangement can avoid losing the direction during the tracking process.

The desired direction of the camera is controlled by the image processing function in Simulink. The main purpose of image processing is to find the location where the template appears in the image. According to the position of the template, the PIC implements a controller responsible of turning the camera to the position where the two center images of the cameras match up.

The data used to calculate the object's distance is provided by the servo motor. From these sets of angular positions and using geometric relationship, the distance from human eye to the object is calculated. Since the direction is closely related to the angle change in the horizontal direction, the error becomes larger as the target moves away from the system.

5.2 Future Work

The mechanism design restricts the range of the eyesight. To remove this problem and reduce its weight, the fixed structure and the helmet can be replaced with a headband. Supposedly, it has a better ability to fit the shape of the head. This way, it becomes capable of better maintaining the relative position between the mechanism and the human eye. In order to remove the obstacle away from the eyesight, the camera responsible for capturing the eye's image should be changed to an alternative position instead of being fixed directly in front of the eye. Using a mirror to reflect the image of the eye may make this idea realizable. In order to get a better tracking efficiency and accuracy, the resolution of the camera needs to be elevated, making the slightest movement readable for the camera. After modifying the tracking system, some applications and experiments with the eye tracking system could be executed.

Reference

[1] Kaufman, A. Bandopadhay, B. Shaviv, "An eye tracking computer user interface", *Proc. of the Research Frontier in Virtual Reality Workshop, IEEE Computer Society Press*, pp. 78-84, 1993

[2] H. Crane, C. Steele, "Accurate three-dimensional eyetracker", *Journal of the Optical Society of America*, vol.17, pp.691-705, 1978

[3] Morimoto C.H., Mimica M.R.M., 'Eye gaze tracking techniques for interactive applications", Computer Vision and *Image Understanding*, vol.98, No.1, pp.4-24, 2005

[4] T. Hutchinson, K. White Jr., K. Reichert, L. Frey, "Human-computer interaction using eye-gaze input", IEEE Trans. Syst. Man Cybernetics, vol.19, pp. 1527-1533, 1989

[5] C. Morimoto, D. Koons, A. Amir, M. Flickner, "Pupil detection and tracking using multiple light sources", Image Vis. Compt., vol.18, No.4, pp.331-336, 2000

[6] LCTech, The eyegaze development system: http://www.eyegaze.com

[7] Y. Ebisawa, M. Ohtani, "A. Sugioka, Proposal of a zoom and focus control method using an altrasonic distance-meter for video-based eye-gaze detection under free-hand condition", Conf. of the IEEE Engineering in Medicine and Biology Society, pp.523-525, 1996

[8] ASL Model H6: http://www.inition.co.uk/

[9] SMI iView X HED system: http://www.smivision.com/

[10] S. shih, J. Liu, "A novel approach to 3d gaze tracking using stereo cameras", IEEE Trans. *Syst. Man Cybernetics*, Part B 3, pp.1-12, 2003

[11] Datasheets: PIC18F2480/2580/4480/4580 Datasheet, Microchip,

http://ww1.microchip.com/downloads/en/DeviceDoc/39637c.pdf

[12] Micro controller units: http://www.microchip.com

[13] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, "Digital Image Processing Using MATLAB, 1st Edition", Prentice Hall, 2005.

[14] Rafael C. Gonzalez Richard E. Woods, "Digital Image Processing, 3rd Edition." Prentice Hall, 2008.

[15] Align DS410 Servo Motor actuation: http://www.align.com.tw/shop/product_info.php?products_id=2722

[16] Microsoft LifeCam HD-5000: http://www.microsoft.com/hardware/digitalcommunication/productdetails.aspx?pid=019

[17] ACV-566F: http://www.jinwei.tw/MENU-04.files/04-AVC-566F.pdf

[18] Controller Area Network: http://en.wikipedia.org/wiki/Controller_Area_Network

[19] Matlab, Image Processing Toolbox, The MathWorks, Inc., 2008.