สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

กล้องวีดีโอที่เก็บข้อมูลด้วยฮาร์ดดิสก์

HARD DISK VIDEO RECORDER

นางสาวพัชรินทร์ อุดมชัยศรี

นายสุธรรม โรจนวิภาต

วพ.
พ523ก
2549

เลขหมู่.....
เลขทะเบียน 72094
วัน,เดือน,ปี - 8 มิ.ย. 2550

b 11763358
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล้องวีดีโอที่เก็บข้อมูลด้วยฮาร์ดดิสก์

HARD DISK VIDEO RECORDER

โดย

นางสาวพัชรินทร์ อุดมชัยศรี

นายสุธรรม โรจนวิภาต

อาจารย์ที่ปรึกษา

อ.เจริญ วงษ์ชุ่มเย็น

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง กล้องวีดีโอที่เก็บข้อมูลด้วยฮาร์ดดิสก์

HARD DISK VIDEO RECORDER

ผู้จัดทำ

1. นางสาวพัชรินทร์ อุดมชัยศรี รหัสนักศึกษา 46010507

2. นายสุธรรม โรจนวิภาต รหัสนักศึกษา 47015341

_____ อาจารย์ที่ปรึกษา

(อ. เจริญ วงษ์ชุ่มเย็น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล้องวีดีโอที่เก็บข้อมูลด้วยฮาร์ดดิสก์

นางสาวพัชรินทร์ อุดมชัยศรี 46010507
นายสุธรรม โรจนวิภาต 47015341
อ.เจริญ วงษ์ชุ่มเย็น อาจารย์ที่ปรึกษา
ปีการศึกษา 2549

บทคัดย่อ

เนื่องจากในปัจจุบันมีการใช้กล้องถ่ายภาพเคลื่อนไหว กันอย่างแพร่หลายแต่มีข้อจำจัดในเรื่องของหน่วยความจำที่มีขนาดจำกัด และต้องทำการบันทึกบนเครื่องคอมพิวเตอร์ส่วนบุคคล ซึ่งจะทำให้ไม่สามารถบันทึกภาพได้อย่างอิสระ

ดังนั้นหากมีแนวคิดที่จะพัฒนาอุปกรณ์ที่สามารถเชื่อมต่อระหว่างกล้อง เข้ากับ ฮาร์ดดิสก์ ได้โดยตรงและไม่จำเป็นต้องทำการเชื่อมต่อกับคอมพิวเตอร์ โครงงานนี้จึงได้นำเสนอกล้องวีดีโอที่เก็บข้อมูลด้วยฮาร์ดดิสก์มาช่วยในการอำนวยความสะดวก โดยโครงงานนี้จะแบ่งออกเป็น 2 ส่วน คือ ส่วนของฮาร์ดแวร์ ที่ทำการเก็บข้อมูลภาพจากกล้องมาบันทึกลงฮาร์ดดิสก์ และส่งข้อมูลภาพผ่านระบบเครือข่าย และส่วนของซอฟแวร์ ที่ใช้ในการดูภาพผ่านทางระบบเครือข่ายตามเวลาจริง และดูภาพที่ได้ทำการบันทึกไว้แล้วในฮาร์ดดิสก์

ซึ่งโครงงานนี้สามารถนำมาประยุกต์ใช้งานในลักษณะของกล้องวงจรปิดได้ นอกจากนั้นยังสามารถส่งข้อมูลภาพจากกล้องที่ตั้งอยู่ในระยะไกลส่งผ่านระบบเครือข่าย ไปยังเครื่องปลายทางที่ต้องการเข้ามาดู เพื่อให้ทราบว่ามีเกิดอะไรขึ้น ณ จุดที่เราตั้งกล้องเอาไว้ ผ่านทางระบบเครือข่ายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hard disk Video Recorder

Patcharin Udomchaisri 46010507
Suthum Rojanavipark 47015341
Charoen Vongchumyen Advisor
Academic Year 2006

ABSTRACT

Nowadays, people use camera to take motional pictures. But they are limited by memory size and inability to recording the sequences images on personal computer.

We purposed the way to develop some devices that can directly connect camera to hard disk and not necessary to connect camera with computer. This project present Hard disk Video Recorder that solves our problem to some extended. We have developed this project in two parts. Hardware part's that receives picture from camera to be recorded on hard disk and send picture data through network, and software part's that is used for watch motional pictures from network in real-time or watch recorded motional pictures from hard disk.

We can apply this project to use as closed circuit television camera. Also, it can send picture from remote camera to end users through network. End users can see real time events that are happened in remote camera area.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึง เพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์เสร็จลงได้ก็คือ อ.เจริญ วงษ์ชุ่มเย็น ซึ่งเป็นที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำความรู้และความช่วยเหลือในทุกๆ ด้านเสมอมา ซึ่งต้องขอกราบขอบพระคุณเป็นอย่างยิ่ง นอกจากนี้ผู้ที่ต้องขอขอบพระคุณเป็นอย่างสูงอีกท่านหนึ่ง คือ คุณสิริโรจน์ จึงถาวรรณ และบริษัทดีไซน์เกทเวย์ จำกัดที่เอื้อเฟื้อ อุปกรณ์และที่ช่วยแก้ปัญหาเกี่ยวกับ Embeddded Board และเป็นผู้ที่คอยให้คำแนะนำดีๆ ในหลายๆ อย่างอีกท่านหนึ่ง

และต้องขอขอบพระคุณบุคคลสำคัญที่สุด ที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมาในทุกๆ ด้าน อันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

พัชรินทร์ อุดมชัยศรี
สุธรรม โรจนวิภาต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หัวเรื่อง **หน้าที่**

บทคัดย่อ ภาษาไทย.....I
บทคัดย่อ ภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....VII
สารบัญรูปภาพ.....VIII

บทที่ 1 บทนำ.....1
1.1 ความเป็นมาของปัญหา.....1
1.2 วัตถุประสงค์ของโครงงาน.....1
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....1
1.4 ขอบเขตของโครงงาน.....2
1.5 ส่วนประกอบของรายงาน.....2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....4
2.1 ฮาร์ดดิสก์ (Hard Disk).....4
2.1.1 มาตรฐานการเชื่อมต่อของฮาร์ดดิสก์.....4
2.1.2 อินเตอร์เฟสในกลุ่ม IDE/ATA (Integrated Drive Electronics).....4
2.1.3 อินเตอร์เฟสในกลุ่ม EIDE (Enhanced IDE).....4
2.1.4 โหมดการทำงานใน Programmed I/O.....7
2.1.5 อินเตอร์เฟสในกลุ่ม Serial ATA (Enhanced IDE).....8
2.2 เจเพ็ก JPEG (Joint Photographic Experts Group).....10
2.3 M-JPEG (Motion Joint Photographic Experts Group).....10
2.4 USB (Universal Serial Bus).....11
2.4.1 ลักษณะโดยรวมของ USB.....11
2.4.2 โมเดลของการสื่อสารใน USB.....11
2.4.3 ชนิดของข้อมูล.....13
2.4.4 USB ไดรเวอร์, ไออาร์พี และ เฟรม (USB Driver, IRPs and Frames).....14
2.4.5 โฮสต์คอนโทรลเลอร์ไดรเวอร์ (Host Controller Driver).....14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หัวเรื่อง **หน้าที่**

2.5 ARM (Advanced RISC Machine) 15

2.5.1 ชุดคำสั่งของหน่วยประมวลผล ARM9 16

2.5.2 รุ่นและความแตกต่างของหน่วยประมวลผล ARM9 17

2.6 ระบบปฏิบัติการ ลินุกซ์ (OS Linux) 18

2.6.1 ประวัติและที่มาของระบบปฏิบัติการ ลินุกซ์ 19

2.6.2 การนำลีนุกซ์มาใช้งานบน Embedded Board 19

2.6.3 โครงสร้างของ Unix 19

บทที่ 3 การออกแบบและพัฒนา 20

3.1 แนวคิดและการออกแบบโดยรวม 20

3.2 ขั้นตอนการจัดเตรียมอุปกรณ์ 22

3.3 แนวคิดในการออกแบบ ซอฟแวร์ 27

3.4 ขั้นตอนการเตรียม Operating System เพื่อใช้ในการพัฒนา Program 31

3.5 ขั้นตอนการเตรียม การใช้งาน Embedded Board ARM AT91RM9200 35

3.6 การพัฒนาโปรแกรมที่ใช้ในการเชื่อมต่อ 41

บทที่ 4 การทดลองและผลการทดลอง 43

4.1 ตัวอย่างการทำงานบน PC ด้วย Fedora Core 4 43

4.2 ตัวอย่างการทำงานบน Embedded Board 45

4.3 สรุปผลการทดลอง 49

บทที่ 5 บทวิจารณ์และบทสรุป 50

5.1 สรุปผลการทำงาน 50

5.2 ปัญหาและอุปสรรคในการทำงาน 50

5.3 บทวิจารณ์โครงงาน 51

5.4 แนวทางแก้ไข 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม..... 52
ภาคผนวก ก ซอร์สโค้ดเมนโปรแกรม..... 53
ภาคผนวก ข คู่มือการใช้งานโปรแกรมบน PC.....66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้าที่

ตารางที่ 2.1 แสดงอัตราการรับส่งข้อมูลในมาตรฐาน SCSI..... 7

ตารางที่ 2.2 แสดงอัตราการรับส่งข้อมูล PIO..... 7

ตารางที่ 2.3 แสดงอัตรา Throughput ของมาตรฐาน DMA..... 8

ตารางที่ 2.4 แสดงอัตราการรับส่งข้อมูลมาตรฐาน DMA..... 8

ตารางที่ 2.5 ตารางแสดงจุดเด่นในละรุ่นของหน่วยประมวลผล ARM9..... 17

ตารางที่ 3.1 แสดงและเปรียบเทียบคุณสมบัติของบอร์ดเพื่อใช้ทำการตัดสินใจ..... 23

ตารางที่ 3.2 แสดงคุณสมบัติของกล้อง Creative Live! Webcam..... 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้าที่

รูปที่ 2.1 ลักษณะพอร์ต มาตรฐาน IDE 5
รูปที่ 2.2 ลักษณะพอร์ท มาตรฐาน SCSI 6
รูปที่ 2.3 ลักษณะพอร์ท มาตรฐาน Serial ATA 9
รูปที่ 2.4 แสดงการทำงานพื้นฐานของการสื่อสารข้อมูลของอุปกรณ์ผ่านพอร์ท USB 12
รูปที่ 2.5 ตัวอย่างเฟรมของ USB 14
รูปที่ 2.6 Device Framework 15
รูปที่ 2.7 โครงสร้างโดยรวมของ CPU ARM920T 16
รูปที่ 2.8 ความแตกต่างระหว่าง CPU ARM922T และ ARM920T 17
รูปที่ 2.9 โครงสร้างของ Unix 18

รูปที่ 3.1 แสดง บล็อกไดอะแกรมโดยรวม 20
รูปที่ 3.2 แสดง บล็อกไดอะแกรม ระบบการทำงานหลัก 22
รูปที่ 3.3 แสดง ลักษณะของ ชิพ ARM 24
รูปที่ 3.4 แสดงลักษณะของกล้อง Creative Live! Webcam 24
รูปที่ 3.5 บล็อกไดอะแกรม การทำงานภายในกล้อง 25
รูปที่ 3.6 แสดงลักษณะและอุปกรณ์ของ IDE/SATA/USB2.0 Adapter 26
รูปที่ 3.7 แสดงลักษณะ Harddisk IDE 26
รูปที่ 3.8 แสดง Sequence Diagram 29
รูปที่ 3.9 แสดง Flowchart ของโปรแกรม 30
รูปที่ 3.10 Menu ที่ต้องเลือกของ Linux Fedora core 4 31
รูปที่ 3.11 Compile ที่ต้องแสดงในการลง Linux Fedora core 4 31
รูปที่ 3.12 Module SDL ที่ต้องทำการเลือกลงบน Linux Fedora core 4 32
รูปที่ 3.13 Menu การ Configuration Buildroot 33
รูปที่ 3.14 Menu การ Configuration Toolchain 34
รูปที่ 3.15 Menu ที่ใช้ในการ สร้าง root filesystem 35
รูปที่ 3.16 การปรับแต่งให้ใช้งานบน Embedded Board 37
รูปที่ 3.17 การปรับแต่งให้ใช้งานกล้อง Video บน Embedded Board 37
รูปที่ 3.18 เลือก USB SPCA5XX ซึ่งเป็นDriver ของกล้องที่นำมาใช้งาน 38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้าที่

รูปที่ 3.19 สิ่งที่ต้องเพิ่มในการ Configuration....39
รูปที่ 3.20 การใช้งาน HyperTerminal เข้าไปใช้งาน Embedded Board.... 40
รูปที่ 3.21 การใช้งาน Embedded Board.... 40
รูปที่ 3.22 สิ่งที่ต้องสร้างก่อนการใช้งาน กล้อง.... 41
รูปที่ 3.23 การใช้งาน โปรแกรมใน Mode ที่ 1.... 42
รูปที่ 3.24 การใช้งาน โปรแกรมใน Mode ที่ 2.... 43

รูปที่ 4.1 แสดงผลการบันทึกภาพทางฝั่ง Server.... 44
รูปที่ 4.2 แสดงผลการแสดงภาพทางฝั่ง Client.... 44
รูปที่ 4.3 แสดงให้ทราบว่ามีการเชื่อมเข้ามาที่ Server จะทำการเปิดให้บริการทันที.... 45
รูปที่ 4.4 แสดงผล ของภาพที่ถูกบันทึกลงบน Hard disk....45
รูปที่ 4.5 แสดงผลการบันทึกภาพทางฝั่ง Server.... 47
รูปที่ 4.6 แสดงผลการแสดงภาพทางฝั่ง Client.... 47
รูปที่ 4.7 แสดงให้ทราบว่ามีการเชื่อมเข้ามาที่ server จะทำการเปิดให้บริการทันที....48
รูปที่ 4.8 แสดงผล ของภาพที่กำลังถูกบันทึกลงบน Hard disk.... 48
รูปที่ 4.9 แสดงผล ของภาพที่ถูกบันทึกลงบน Hard disk....49
รูปที่ 4.10 แสดงผล การใช้งาน Application ที่ใช้งานจากบน windows....49

รูปที่ ข-1 ภาพรวมของโปรแกรม....67
รูปที่ ข-2 mode การเปิดภาพจาก Storage.... 68
รูปที่ ข-3 แสดงการทำงานของ Mode การเปิดภาพจาก Storage....68
รูปที่ ข-4 mode การเชื่อมต่อผ่าน network....69
รูปที่ ข-5 แสดงการทำงานของ Mode การเชื่อมต่อผ่าน network.... 69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

เนื่องจากในปัจจุบันมีการใช้กล้องถ่ายภาพเคลื่อนไหว กันอย่างแผ่หลายแต่มีข้อจำจัดในเรื่องของหน่วยความจำที่มีขนาดจำกัดและต้องทำการบันทึกบนเครื่องคอมพิวเตอร์ส่วนบุคคล ซึ่งจะทำให้ไม่สามารถบันทึกภาพได้อย่างอิสระ ดังนั้นโครงงานนี้จึงได้คิดและพัฒนาอุปกรณ์ที่สามารถเชื่อมต่อระหว่างกล้อง เข้ากับ ฮาร์ดดิสก์ ได้โดยตรงและไม่จำเป็นต้องทำการเชื่อมต่อกับคอมพิวเตอร์

ถึงแม้ว่าในปัจจุบันอาจมีกล้องหลายๆยี่ห้อ ที่มีการพัฒนาสินค้าของตนให้สามารถเชื่อมต่อการทำงานของกล้องเข้ากับ ฮาร์ดดิสก์ได้แล้ว แต่ก็ยังมีราคาค่อนข้างแพง และมุ่งเน้นประโยชน์ไปทางด้านการใช้งานในเรื่องของความบันเทิง

โครงงานที่เรานำเสนอนั้นคือการนำเอา ฮาร์ดดิสก์ที่เรามีใช้กันอยู่ในปัจจุบันมาประยุกต์ใช้งานกับกล้อง ซึ่งเราสามารถนำมาประยุกต์ใช้งานในลักษณะของกล้องวงจรปิดได้

โดยขั้นตอนต่อไปถ้าหากทำการเชื่อมต่อกันได้แล้ว เราจะนำไปใช้ในการส่งข้อมูลภาพจากกล้องที่ตั้งอยู่ในระยะไกลส่งผ่านระบบ Network ไปยังเครื่องที่ต้องการเข้ามาดู เพื่อให้ทราบว่ามีเกิดอะไรขึ้น ณ จุดที่เราตั้งกล้องเอาไว้ ผ่านทางระบบ ได้

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อแก้ปัญหาในเรื่องของหน่วยความจำในการบันทึกข้อมูลของกล้องที่มีอยู่อย่างจำกัด และ ไม่พอเพียงต่อการบันทึกข้อมูลที่มีระยะเวลานานๆ

1.2.2 สามารถนำเอาข้อมูลที่ได้มาจากการทำงานไป วิจัยในเรื่องของสมองกลฝังตัว ในอนาคตได้ เพื่อเป็นประโยชน์กับนักศึกษาในรุ่นถัดต่อไปในอนาคต

ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 สามารถบันทึกภาพจากกล้อง VDO ลง Harddisk ได้โดยตรงโดยไม่ต้องผ่าน เครื่องคอมพิวเตอร์ จากลักษณะการทำงานนี้เราไม่จำเป็นต้องเข้าไปยุ่งเกี่ยวกับการทำงานของอุปกรณ์

1.3.2 มีระบบการจัดการกับ ไฟล์ ข้อมูลโดยอัตโนมัติ ซึ่งถือเป็นข้อดีข้อหนึ่งที่เราไม่จำเป็นต้องเข้าไปจัดการกับ ไฟล์ เราไม่ต้องทำการเปลี่ยน Harddisk เมื่อข้อมูลเต็มระบบจะจัดการได้โดยอัตโนมัติ

1.3.3 ในการทำงานของระบบ จะไม่มีการยุ่งเกี่ยวในเรื่องของคุณภาพของภาพ แต่จะมุ่งเน้นไปที่การจัดการกับไฟล์ ในแบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของโครงงาน

ศึกษา ออกแบบและพัฒนาอุปกรณ์ที่สามารถ บันทึกภาพจากกล้องวีดีโอลงอุปกรณ์ Harddisk ซึ่งสามารถแบ่งหัวข้อได้ดังนี้

1.4.1 สามารถบันทึกภาพจากกล้องลง Harddisk ได้อย่างต่อเนื่อง โดยการบันทึกภาพนั้นจะทำการบันทึกภาพต่อเนื่องไปเรื่อยๆ และมีสามารถบันทึกภาพทับลงที่ไฟล์เดิมที่เคยบันทึกเอาไว้ได้

1.4.2 เชื่อมต่อเพื่อดูภาพในเวลา ปัจจุบันได้ โดยอาศัยการทำงานของระบบ Network ซึ่งจะใช้การเขียนโปรแกรมเพื่อเชื่อมต่อกับ Socket และการรับส่งข้อมูลแบบ TCP/IP โปรโตคอล ซึ่งเป็นที่ใช้งานกันอย่างแผ่หลาย

1.4.3 สามารถทำงานในลักษณะที่เป็นกล่าวถึงได้พร้อมกันทั้งสองรูปแบบ คือการ บันทึกข้อมูลลง Harddisk ไปพร้อมกับการเชื่อมต่อเครือข่าย Network ได้พร้อมกัน

1.4.4 สามารถพัฒนาโปรแกรม ที่ใช้งานกับโครงงานที่ออกแบบไว้ โดยสามารถทำการเชื่อมต่อเข้าไปทาง Socket เพื่อเข้าไปเปิดดูภาพ และ สามารถทำการบันทึกภาพทางโปรแกรม รวมไปถึงการเปิดไฟล์ภาพที่บันทึกลง Harddisk ที่เคยบันทึกเอาไว้ได้

1.5 ส่วนประกอบของรายงาน

อธิบายเนื้อหาของรายงานฉบับนี้ในแต่ละบทอย่างคร่าวๆ ดังนี้

เนื้อหาในบทที่ 1 กล่าวถึงความเป็นมาของปัญหา วัตถุประสงค์ของโครงงาน ประโยชน์ที่คาดว่าจะได้รับ ขอบเขตของโครงงาน และส่วนประกอบของรายงานฉบับนี้

เนื้อหาในบทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงงาน ซึ่งประกอบด้วย

- ทฤษฎี ฮาร์ดดิสก์(Hard Disk) ซึ่งจะเป็นลักษณะของโครงสร้างและระบบการโอนถ่ายข้อมูลของอุปกรณ์ฮาร์ดดิสก์โดยประมาณ
- ทฤษฎี เจเพ็ก (JPEG) โดยจะเป็นการอธิบายที่มาและหลักการของไฟล์ข้อมูลภาพ
- ทฤษฎี เอ็มเจเพ็ก (MJPEG) จะเป็นการอธิบายที่มาและหลักการของไฟล์ข้อมูลภาพเคลื่อนไหว
- ทฤษฎี ยูเอสบี (USB) เป็นการอธิบายหลักโดยรวมของการทำงานในรูปแบบการส่งข้อมูล ยูเอสบี
- ทฤษฎี อาร์ม (ARM) เป็นการอธิบายโครงสร้างและข้อมูลพื้นฐานของ CPU ARM
- ทฤษฎี ระบบปฏิบัติการ ลินุกซ์ (OS Linux) เป็นการอธิบายประวัติและที่มาของระบบปฏิบัติการ ลินุกซ์ รวมไปถึงขั้นตอนการใช้งาน ลินุกซ์บน อาร์ม

เนื้อหาในบทที่ 3 กล่าวถึงแนวคิดและวิธีการการทำงานของระบบ โดยแยกออกเป็นส่วน Hard ware และ Soft ware

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื้อหาในบทที่ 4 กล่าวถึงการทดลองและผลการทดลอง โดยจะจำลองการใช้งานทั้งหมดบนเครื่อง PC

เนื้อหาในบทที่ 5 กล่าวถึงการสรุปผลของโครงงาน โดยจะบอกถึงปัญหาที่เกิดขึ้นกับการทำงาน และแนวทางการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 ฮาร์ดดิสก์ (Hard Disk)

2.1.1 มาตรฐานการเชื่อมต่อของฮาร์ดดิสก์

อินเตอร์เฟส (Interface) ของฮาร์ดดิสก์ หมายถึง ช่องสัญญาณที่ฮาร์ดดิสก์ใช้รับส่งคำสั่ง / ข้อมูล/ สถานะกับระบบ อินเตอร์เฟสเป็นส่วนที่มีบทบาทต่อประสิทธิภาพของฮาร์ดดิสก์ ไม่น้อยไปกว่าคุณลักษณะภายในตัวฮาร์ดดิสก์เองเพราะแม้ว่าหัวอ่าน/เขียนจะสามารถอ่านข้อมูลจากพื้นผิวของฮาร์ดดิสก์ได้รวดเร็วเพียงใดก็ตามถ้าอินเตอร์เฟสไม่เอื้ออำนวยต่อการส่งผ่านข้อมูลที่อัตราเร็วนั้นก็ถือว่าเปล่าประโยชน์ อินเตอร์เฟสเป็นมาตรฐานการรับ – ส่งข้อมูลระหว่างฮาร์ดดิสก์กับระบบบัส จะเป็นตัวกำหนดวิธีการส่งข้อมูล ฮาร์ดแวร์ที่สนับสนุนและชุดคำสั่งในการโอนย้ายข้อมูลฮาร์ดดิสก์ในยุคแรกๆ อินเตอร์เฟสของมันมักจะต้องอาศัยอุปกรณ์ที่เรียกว่า " ฮาร์ดดิสก์ คอนโทรลเลอร์" (Harddisk Controller) คอยควบคุมการรับ – ส่งข้อมูลอีกที ต่อมาเมื่อความต้องการด้านการจัดเก็บข้อมูลมีมากขึ้น ต้องการความเร็วในการรับ – ส่งข้อมูลมากขึ้น อินเตอร์เฟสใหม่ๆ จึงได้ถูกพัฒนาขึ้นมาเพื่อรองรับความต้องการเหล่านั้น

2.1.2 อินเตอร์เฟสในกลุ่ม IDE/ATA (Integrated Drive Electronics)

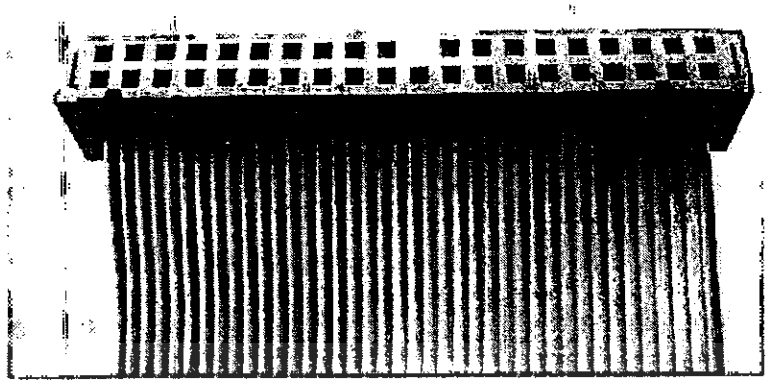
สิ่งสำคัญที่ทำให้อินเตอร์เฟสตัวนี้แตกต่างจาก ST506 และ ESDI ก็คือ การรวมเอาตัวคอนโทรลเลอร์หรือส่วนควบคุมในการรับส่งข้อมูลเข้าไว้ในฮาร์ดดิสก์ด้วยเลยทำให้บริษัทผู้ผลิตฮาร์ดดิสก์สามารถที่จะเพิ่มเทคโนโลยีใหม่ๆให้กับฮาร์ดดิสก์ของตนได้อย่างอิสระ และยังช่วยลดต้นทุนโดยรวมของระบบคอมพิวเตอร์เครื่องหนึ่งๆได้อีกทางหนึ่งด้วย

คุณสมบัติเด่นที่มีอยู่ในอินเตอร์เฟสIDE ได้แก่ ความสามารถในการเชื่อมต่อฮาร์ดดิสก์ได้พร้อมกัน 2 ตัว ซึ่งฮาร์ดดิสก์แต่ละตัวจะมีขนาดความจุมากที่สุด 528 MB นอกจากนั้น อินเตอร์เฟสแบบ IDE ยังมีราคาที่นับว่าไม่แพงเมื่อเทียบกับประสิทธิภาพของมันในเวลาต่อมาเนื่องจากเทคโนโลยีของซีพียูที่มีความเร็วเพิ่มขึ้นอย่างมาก ในขณะที่ระบบบัสแบบ PCI ที่มีความเร็วสูงได้ถูกนำมาใช้ ทำให้ประสิทธิภาพความเร็วของอินเตอร์เฟส IDE เริ่มเป็นอุปสรรคในด้านความเร็วของการรับ – ส่งข้อมูล

2.1.2.1 อินเตอร์เฟสในกลุ่ม EIDE (Enhanced IDE)

ด้วยข้อจำกัดของอินเตอร์เฟสแบบ IDE ในปี 1993 เทคโนโลยี EIDE (Enhanced IDE) ตัวใหม่ที่พัฒนามาจาก IDE และคิดค้นโดย Western Digital จึงได้ถือกำเนิดขึ้น ในขณะที่มาตรฐานในการรับ – ส่งข้อมูลก็ได้เปลี่ยนจาก ATA ไปเป็น ATA – 2 ที่มีความเร็วมากกว่า 16.6 MB/s และสามารถจุข้อมูลได้มากถึงระดับกิกะไบต์ รองรับขนาดความจุของฮาร์ดดิสก์ได้สูงถึง 137 GB และล่าสุดสามารถพัฒนาความจุได้มากถึง 200 กิกะไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ลักษณะพอร์ต มาตรฐาน IDE

อินเตอร์เฟสแบบ EIDE รองรับการเชื่อมต่อฮาร์ดดิสก์ใน 2 Channel ซึ่งในแต่ละ Channelจะรองรับการเชื่อมต่อฮาร์ดดิสก์ได้ 2 ตัว รวมเป็นมากที่สุด 4 ตัว และยังสามารถเชื่อมต่อกับอุปกรณ์แบบ IDE และระบบปฏิบัติการตัวเก่าที่รองรับการใช้งานกับอินเตอร์เฟส IDE ได้อย่างไม่มีปัญหา นอกจากนี้ อินเตอร์เฟสแบบ EIDE ยังรองรับการเชื่อมต่อกับอุปกรณ์ต่อพ่วงชนิดอื่นๆ นอกเหนือจากฮาร์ดดิสก์ เช่น CD-ROM และอุปกรณ์เทปแบ็คอัพ ซึ่งเรียกเทคโนโลยีนี้ว่า "ATAPI" (At Attachment Packet Interface) ที่จะมีชุดคำสั่งและรีจิสเตอร์เฉพาะเอาไว้เพื่อที่จะให้อุปกรณ์ดังกล่าวเรียกใช้งานได้ ในส่วนของช่องที่ใช้เชื่อมต่อแบบ EIDE นี้จะมีจำนวน 40 ขา โดยจะเชื่อมต่อเข้ากับสายสัญญาณหรือที่อาจจะเรียกว่า สายแพ ก็ได้ เครื่องคอมพิวเตอร์แต่ละเครื่องจะสามารถต่อฮาร์ดดิสก์ได้ 4 ตัว ฮาร์ดดิสก์แบบ EIDE เป็นมาตรฐานที่นิยมใช้กันโดยทั่วไป เนื่องจากราคาถูก หาซื้อง่ายและมีการติดตั้งไม่ยุ่งยาก

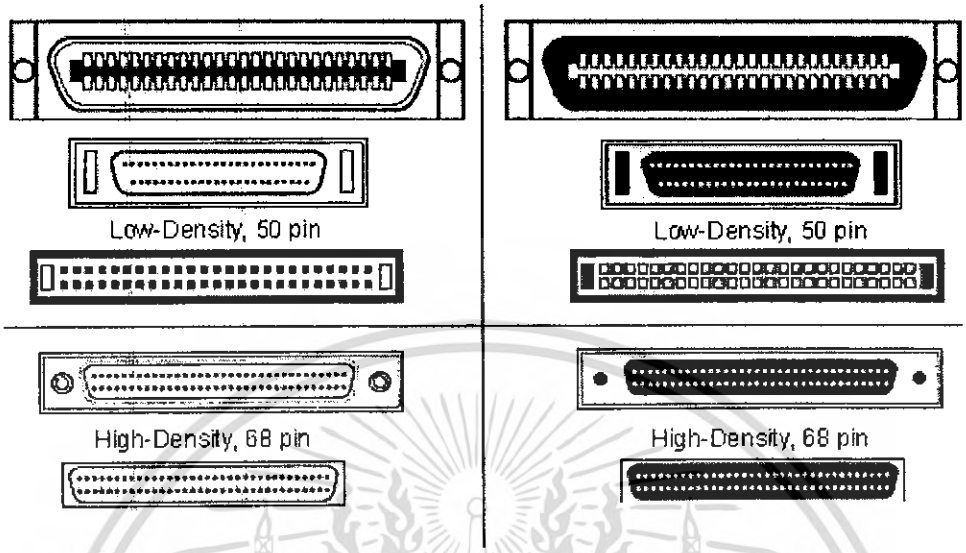
2.1.3 อินเตอร์เฟสในกลุ่ม SCSI (Small Computer Systems Interface)

SCSI เป็นอินเตอร์เฟสที่ใช้ต่อพ่วงอุปกรณ์ที่ใช้งานได้ทั้งบนเครื่องคอมพิวเตอร์พีซีและเครื่องแมคอินทอช ซึ่งไม่ได้หมายความเพียงแค่อุปกรณ์อย่างฮาร์ดดิสก์ แต่ยังรวมไปถึง CD-ROM, Scanner หรือ Printer ฯลฯ หลักการทำงานของ SCSI ค่อนข้างมีความซับซ้อนจนเปรียบได้กับระบบแลนเล็กๆ เป็นฮาร์ดดิสก์ที่มีจุดเด่นในเรื่องของความจุที่มีมากกว่า EIDE และมีความเร็วในการส่งถ่ายข้อมูลสูงกว่าแบบ EIDE

เหตุที่ SCSI ได้รับความนิยมโดยเฉพาะการใช้งานกับเครื่องระดับ Server เนื่องจากมีความเร็วในการรับ – ส่งข้อมูลสูงที่สุดถึง 160 MB/s และมีความสามารถในการเชื่อมต่ออุปกรณ์ได้มากถึง 8 อุปกรณ์สำหรับ Narrow SCSI และมากที่สุดถึง 16 อุปกรณ์สำหรับ Wide SCSI ซึ่งอุปกรณ์แต่ละตัวจะมีคอนโทรลเลอร์ในตัวมันเอง จุดเด่นของ SCSI อีกอย่างก็คือ อุปกรณ์ทุกๆตัวมีความฉลาดทำการรับ – ส่งข้อมูลระหว่างอุปกรณ์ด้วยกันโดยไม่ต้องพึ่งซีพียู และสามารถทำงานไปได้พร้อมๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นเดียวกับการทำงานของระบบ Multitasking ใน Windows ขณะที่ในอุปกรณ์ประเภท IDE ไม่สามารถรองรับการทำงานเช่นนี้ได้



รูปที่ 2.2 ลักษณะพอร์ท มาตรฐาน SCSI

ฮาร์ดดิสก์แบบ SCSI เหมาะที่จะใช้สำหรับเครื่องประเภทเครื่องแม่ข่าย (Server) มากกว่าเครื่องที่ใช้ตามบ้าน เนื่องจากมีราคาแพง และมีการติดตั้งที่ยุ่งยากกว่า เพราะจะต้องมีการติดตั้งการ์ดควบคุมเพิ่มเติม ในการติดตั้งจะต้องใช้ 50 ขา จะเห็นว่าสายสัญญาณของฮาร์ดดิสก์แบบ SCSI จะใหญ่กว่า แบบ EIDE ที่มีเพียง 40 ขา

	SCSI	Fast	Wide	Fast	Ultra	UltraWide	Ultra 2	Ultra-3
บัสข้อมูล (บิต)	8	8	19	16	16	32	16	32
ความถี่(MHz)	5	10	5	10	20	20	40	40
รับส่งข้อมูล(MB/s)	5	10	10	20	40	80	80	160
คอนเน็คเตอร์	SCSI-1	SCSI-2	SCSI -2	SCSI-2	SCSI-3	SCSI-3	SCSI-3	SCSI-3

ตารางที่ 2.1 แสดงอัตราการรับส่งข้อมูลในมาตรฐาน SCSI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 โหมดการทำงานใน Programmed I/O

เนื่องจากระบบบัสแบบ IDE เป็นระบบบัสที่ใช้เป็นมาตรฐานในการรับ-ส่งข้อมูลของฮาร์ดดิสก์ ในปัจจุบัน โดยที่ประสิทธิภาพความเร็วในการรับ – ส่งข้อมูลของมันจะขึ้นอยู่กับรูปแบบในการรับส่งข้อมูลหรือเรียกว่า "โหมดการทำงานของฮาร์ดดิสก์" (Transfer Mode) ซึ่งในแต่ละโหมดการทำงานนั้นจะมีความเร็วไม่เท่ากัน โดยสามารถจะแบ่งโหมดการทำงานของฮาร์ดดิสก์ออกเป็น 2 กลุ่มใหญ่ๆด้วยกัน คือ โหมด PIO และ DMA

2.1.4.1 PIO (Programmed Input Output)

เป็นโหมดการทำงานที่ใช้กันมาตั้งแต่ฮาร์ดดิสก์ในยุคแรกๆซึ่งการรับส่งข้อมูลของฮาร์ดดิสก์นั้นจะต้องทำงานร่วมกับซีพียูโดยตรง ไม่ว่าจะเป็นการเขียนหรืออ่านข้อมูล โดยซีพียูจะทำหน้าที่เป็นตัวกลางในการติดต่อกับหน่วยความจำอีกทีหนึ่ง ซึ่งประสิทธิภาพความเร็วในโหมด PIO ค่อนข้างช้าและยังใช้ซีพียูแบกรับภาระค่อนข้างมาก ในปัจจุบันจึงไม่ค่อยได้รับความนิยมมากนัก

โหมด PIO นั้นยังแบ่งเป็นโหมดการทำงานย่อยๆ ได้อีกเป็น 5 ระดับ คือ โหมด PIO 0, 1, 2 ซึ่ง ใช้ได้กับฮาร์ดดิสก์แบบ IDE ทุกๆประเภท ส่วนในโหมด 3 และ 4 นั้นจะรองรับก็แต่เฉพาะฮาร์ดดิสก์แบบ IDE ที่มีมาตรฐาน ATA – 2 ขึ้นไป โดยในแต่ละโหมดการทำงานจะมีความเร็วในการรับ – ส่งข้อมูลได้ไม่เท่ากัน ดังแสดงตารางที่ 2.1

PIO mode	อัตราการรับส่งข้อมูล (MB./sec)	อินเตอร์เฟส
0	3.3	ATA
1	5.2	ATA
2	8.3	ATA
3	I1.1	ATA-2
4	16.6	ATA-2

ตารางที่ 2.2 แสดงอัตราการรับส่งข้อมูล PIO

2.1.4.2 DMA (Direct Memory Access)

เป็นวิธีการส่งข้อมูลแบบใหม่ที่ทำงานได้เร็วกว่าในโหมด PIO เนื่องจากฮาร์ดดิสก์จะสามารถรับส่งข้อมูลกับแรมได้โดยตรง โดยอาศัย DMA Controller จึงไม่ต้องพึ่งพาซีพียูในการโอนย้ายข้อมูล ทำให้อัตราเร็วในการรับส่งข้อมูลเพิ่มสูงขึ้น ซึ่งวิธีการส่งข้อมูลแบบ DMA นี้ก็ยังสามารถแบ่งออกเป็นโหมดการทำงานย่อยได้ 3 ระดับ คือ DMA 0, 1, 2 ต่อมาเมื่อ DMA ได้กลายเป็นมาตรฐานที่ใช้กันอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แพร่หลายในอุปกรณ์ต่อพ่วงประเภทต่างๆ จนต้องผนวกความสามารถของ DMA เอาไว้ในฮาร์ดดิสก์เลย ทำให้ไม่ต้องใช้ DMA Controller ในการจัดการโอนย้ายข้อมูลเช่นเดิม อัตราความเร็วของการส่งข้อมูลจะยิ่งสูงขึ้นไปอีก มาตรฐานใหม่นี้เรียกว่า Ultra DMA ซึ่งประกอบด้วยภาวะโหมดการทำงาน Ultra DMA33 , Ultra DMA 66 และ Ultra DMA 100 โดยแต่ละโหมดของ DMA จะมีความเร็วในการรับ-ส่งข้อมูล ดังในตารางที่ 2.2

DMA Transfer Mode	Maximum Throughput (MB/s)	Standard
DMA Mode 0	4.16	ATA
DMA Mode 1	13.3	ATA-2
DMA Mode 2	16.6	ATA-2
Ultra DMA 33	33.3	Ultra ATA 33
Ultra DMA 66	66.7	Ultra ATA 66
Ultra DMA 100	100	Ultra ATA 100

ตารางที่ 2.3 แสดงอัตรา Throughput ของมาตรฐาน DMA

	DMA mode	อัตราการรับส่งข้อมูล (MB./sec)	อินเตอร์เฟส
Single Word	0	2.1	ATA
	1	4.2	ATA
	2	8.3	ATA
Multi Word	0	4.2	ATA
	1	13.3	ATA-2
	2	16.6	ATA-2

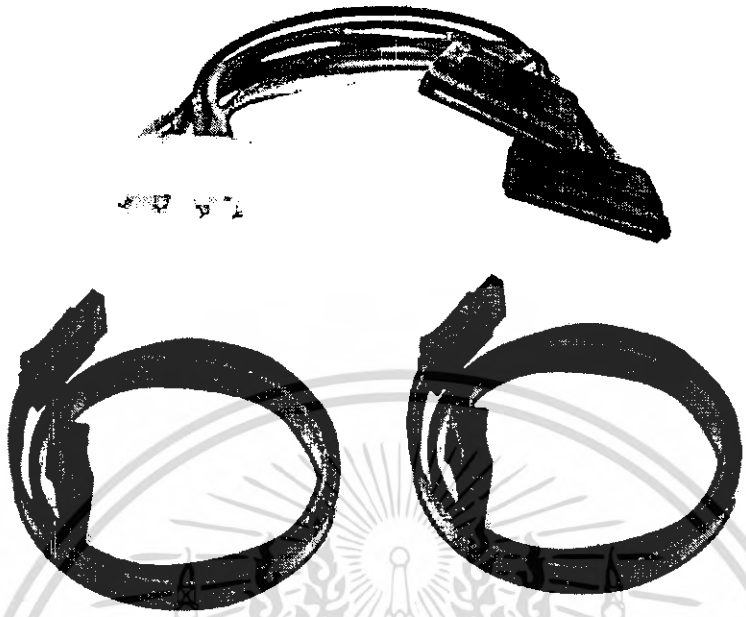
ตารางที่ 2.4 แสดงอัตราการรับส่งข้อมูลมาตรฐาน DMA

2.1.5 Serial ATA

Serial ATA เป็นเทคโนโลยีใหม่ที่มีการส่งข้อมูลแบบอนุกรมความเร็วสูง โดยคาดว่าจะเข้ามาแทนที่อินเตอร์เฟส EIDE (EIDE เป็น Parallel ATA เนื่องจากมีการส่งข้อมูลแบบขนาน คือ ส่งข้อมูลไปทีละหลายๆบิตพร้อมกัน) ในอนาคตอันใกล้ ทั้งนี้เทคโนโลยีการส่งข้อมูลแบบอนุกรมเคยถูกนำมาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนหน้านี้แล้ว ได้แก่ อุปกรณ์ประเภท USB (Universal Serial Bus) และ IEEE 1394 (Fire Wire) ในกลุ่มอุปกรณ์ต่อพ่วง (Peripherals)



รูปที่ 2.3 ลักษณะพอร์ท มาตรฐาน Serial ATA

ในปี 1999 ได้ก่อตั้งเป็นองค์กร Serial ATA Working Group ขึ้นมาเพื่อกำหนดมาตรฐานและทิศทางในอนาคตของ Serial ATA และเมื่อเปรียบเทียบ Serial ATA การเชื่อมต่อ ATA แบบเก่าอย่าง EIDE ข้อดีของ Serial ATA ที่มีเหนือกว่าก็คือ

- มีการสิ้นเปลืองพลังงานที่น้อยกว่า
- จำนวนขาสัญญาณที่ลดลง
- มีอัตราการรับ –ส่งข้อมูลสูงถึง 150 MB/s (ในอนาคตจะถูกพัฒนาให้มีความเร็วสูงถึง 300 MB/s)
- มีการตรวจสอบข้อผิดพลาดของข้อมูลที่มีประสิทธิภาพสูงกว่า
- ใช้สายเคเบิลที่มีขนาดเล็กกะทัดรัด
- สามารถต่อพ่วงอุปกรณ์ได้มากถึง 127 อุปกรณ์

โดยรูปแบบในการเชื่อมต่อเครื่องคอมพิวเตอร์กับอินเตอร์เฟส Serial ATA จะมีอยู่ใน 2 ลักษณะ ดังนี้

1. เชื่อมต่ออุปกรณ์ประเภท Serial ATA เข้ากับชิปเซตที่รองรับการทำงานของ Serial ATA
2. เชื่อมต่อกับอุปกรณ์ Adapter ที่ใช้สำหรับแปลงกลับไปกลับมาระหว่าง Serial ATA กับ Parallel ATA หรือ Parallel ATA เป็น Serial ATA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial ATA มีจุดเด่นอยู่ที่สามารถพัฒนาความเร็วในการโอนข้อมูลระหว่างฮาร์ดดิสก์กับระบบได้สูงกว่ามาตรฐานเดิม Ultra ATA (EIDE) มีความเร็วสูงสุดที่ 133 เมกกะไบต์ และเป็นที่คาดว่าจะสามารถพัฒนาความเร็วได้สูงสุดถึง 600 เมกกะไบต์ต่อวินาที

2.2 เจเพ็ก JPEG (Joint Photographic Experts Group)

เป็นรูปแบบของกราฟิกที่เป็นแนวคิดที่ใช้สำหรับภาพของธรรมชาติในเชิงซับซ้อน ซึ่งรวมถึงภาพถ่าย ภาพศิลปะ และภาพวาด (รูปแบบนี้ไม่เหมาะสำหรับภาพวาดลายเส้น ข้อความ หรือภาพการ์ตูนง่ายๆ) รูปแบบกราฟิกนี้ใช้ในการบีบอัดภาพแบบมีบางส่วนสูญหาย (lossy compression) เข้าช่วยโดยอาศัยหลักการมองเห็นของมนุษย์ที่ว่า การเปลี่ยนแปลงของสี เพียงเล็กน้อยจะสังเกตเห็นได้ไม่ชัดเจนเท่าการเปลี่ยนแปลงในความสว่าง การบีบอัดของเจเพ็กจะไม่สามารถสังเกตเห็นได้ ตราบใด ที่ไม่ใช้สัดส่วนของการบีบอัดสูง โดยปกติแล้วเจเพ็กจะใช้สัดส่วนของการบีบอัด 10:1 หรือ 20:1 โดยเราจะไม่ สามารถสังเกตเห็นคุณภาพของภาพนั้นด้อยลง ซึ่งเป็นสัดส่วนที่ดีกว่าที่ใช้ในรูปแบบการเปลี่ยนระหว่างกราฟิก (Graphics Interchange Format : GIF) รูปแบบของกราฟิกแบบเจเพ็กนี้ พัฒนาโดยคณะกรรมการของผู้เชี่ยวชาญด้านคอมพิวเตอร์กราฟิก (JPEG) ดู Joint Photographic Experts Group (JPEG) ประกอบ

มาตรฐานการบีบขนาดแบบ JPEG ไม่ได้ถูกออกแบบมาเพื่อฮาร์ดแวร์หรือซอฟต์แวร์ใดๆ โดยเฉพาะ แต่ได้นำเสนอวิธีการบีบขนาดที่สามารถใช้ทั่วๆ ไปหลายวิธี ดังนั้นจึงมีการบีบขนาดหลายวิธีที่เกิดขึ้นมาโดยใช้มาตรฐานการบีบขนาดแบบ JPEG การบีบขนาดด้วยวิธีนี้ช่วยลดขนาดของภาพกราฟิกและประหยัดเวลาในการโหลดได้มาก เหลือเพียงหนึ่งในสิบของภาพเดิม และบางครั้งสามารถลดขนาดลงได้มากถึง 100 ต่อ 1

JPEG เป็นไฟล์ที่เหมาะสำหรับใช้ในภาพประเภทภาพถ่าย (โทนสีต่อเนื่อง) เนื่องจากใช้สีทั้งสเปกตรัมสีที่มีในมอนิเตอร์ และเป็นไฟล์ประเภทที่ถูกบีบอัดให้เล็กลงเพื่อให้โหลดเร็วขึ้นเช่นเดียวกับ GIF โดยการตัดค่าสี ในช่วงที่ตามองไม่เห็นทิ้งไป แต่เมื่อบันทึกไฟล์เป็น JPEG แล้ว ข้อมูลสีที่ถูกตัดทิ้งไปจะไม่สามารถเรียกกลับมาได้อีก ถ้าต้องการใช้ค่าสีเหล่านั้นในอนาคต ควรจะบันทึกเป็นไฟล์ชนิดอื่นแล้วเปลี่ยนเป็นไฟล์ JPEG ด้วยการบันทึกเป็นไฟล์ก็อปปี้

2.3 M-JPEG (Motion Joint Photographic Experts Group)

คือ รูปแบบของไฟล์ภาพเครื่องไหวซึ่งเกิดจากการนำภาพที่เกิดจากการบีบอัดในมาตรฐาน เจเพ็กมาเรียงกันแล้วทำการเล่นภาพ จะทำให้เกิดภาพเครื่องไหวออกมา โดยมากจะใช้สัดส่วนการบีบอัดข้อมูลของภาพเท่ากับ 12 : 1 หรือ 5: 1 ซึ่งจะเหมาะกับงานที่ไม่ต้องการลายละเอียดมากและงานที่ไม่จำเป็นต้องทำการเป็นแสดงข้อมูลเสียง เช่น ลักษณะการบันทึกภาพวงจรปิด หรือ การบันทึกสภาพการจราจรบนท้องถนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 USB (Universal Serial Bus)

2.4.1 ลักษณะโดยรวมของ USB

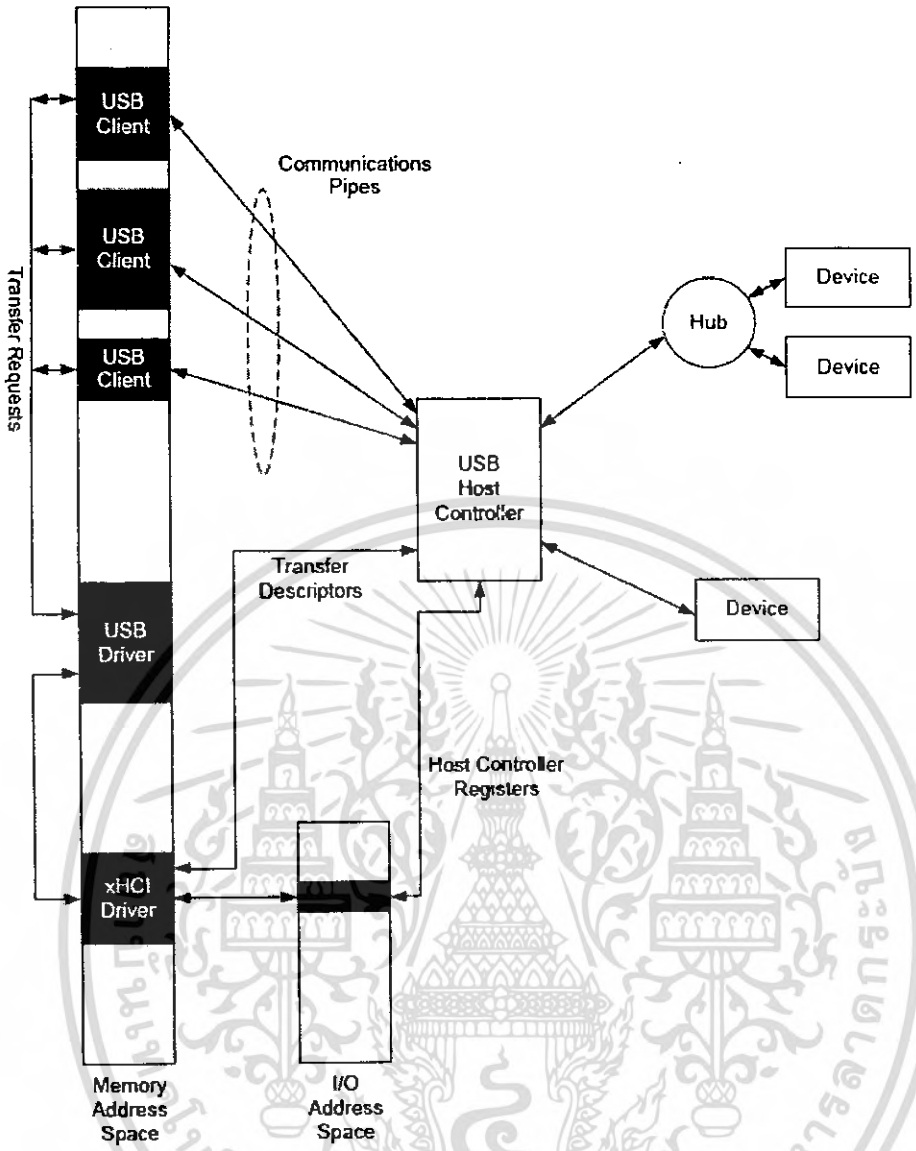
USB ได้ถูกนำมาใช้เพื่อกำจัดข้อเสียบางอย่างของ PC IO ที่มีใช้อยู่เดิมในการติดต่อ อุปกรณ์รอบข้าง ของคอมพิวเตอร์ส่วนบุคคล ที่มีใช้อยู่ทั่วไป โดยจะช่วยให้การติดต่อ อุปกรณ์รอบข้างสะดวกมากขึ้นเมื่อเทียบกับแบบเก่า

โดย USB จะมีบัสที่ใช้ในการติดต่อโดยจะเรียกระบบบัสนี้ว่า การติดต่อแบบ "Host/Slave" หมายถึงตัว PC จะเป็นผู้จัดการ การจัดส่งข้อมูลทั้งหมด และอุปกรณ์ที่นำมาต่อพ่วงเพียงแต่พร้อมที่จะรับส่งข้อมูลเท่านั้น โดยตัว USB Host Controller หรือตัวควบคุมของ USB นั้น จะรวมอยู่ในตัว Chipset บนเมนบอร์ด แต่ถ้าหากเป็น Chipset รุ่นเก่าๆก็จะยังไม่มีตัวควบคุม USB ในการจัดส่งข้อมูลนั้น ภายในสาย USB จะมีสายภายในทั้งหมด 4 เส้น ในจำนวนนี้ 2 เส้นใช้เพื่อเป็นสายในการส่งข้อมูล อีก 2 เส้นใช้สำหรับจ่ายไฟเลี้ยง แรงดัน +5 V และ กราวด์ ข้อมูลที่ถูกส่งผ่าน USB นั้นจะต่างไปจาก Parallel และ Serial port ซึ่งทั้ง 2 ชนิดนี้นั้น จะส่งข้อมูลเป็น bits เดี่ยวๆ แต่ USB จะส่งเป็นชุดข้อมูล สิ่งที่ทำให้ USB ส่งข้อมูลได้เร็วถึง 12Mbits/s นั้นอยู่ที่ความแม่นยำของการส่ง เพราะ สายที่นำมาใช้สำหรับ USB นั้น มีระดับสัญญาณรบกวน และ ความเพี้ยนของรูปสัญญาณนั้นน้อยกว่า ทั้ง Parallel และ Serial port และการส่งสัญญาณแบบ isochronous data delivery หมายถึง อุปกรณ์แต่ละชิ้นบนระบบบัสนั้น จะถูกจำกัดขนาดของ bandwidth ให้แน่นอน อย่างไรก็ตาม bandwidth โดยรวมแล้วก็ไม่เกิน 12 Mbits/s นี่เป็นเหตุผลที่ทำให้ USB สามารถต่ออุปกรณ์ได้มากที่สุด 127 อุปกรณ์

2.4.2 โมเดลของการสื่อสารใน USB (USB Communication model)

การติดต่อสื่อสารในระบบ USB มีข้อดีคือลดความซับซ้อนของการติดต่อสื่อสาร ระบบUSBไม่ต้องกำหนดหน่วยความจำและไอโอแอดเดรสให้กับอุปกรณ์ทุกตัว ระบบUSBนั้นต้องการเพียงแค่กำหนดพื้นที่หน่วยความจำให้กับUSBซิสเต็มซอฟท์แวร์ (USB System Software) และกำหนดพื้นที่หน่วยความจำ ไอแอดเดรส และอินเทอร์รัปต์แอดเดรส ให้กับUSBโฮสต์คอนโทรลเลอร์ (USB Host Controller)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงถึงการทำงานพื้นฐานของการสื่อสารข้อมูลของอุปกรณ์ผ่านพอร์ท USB

เริ่มแรกUSBไคลเอนต์(USB Client) ส่งคำขอการส่งข้อมูลไปยัง USB ซิสเต็มซอฟท์แวร์ USB ไคลเอนต์ไดรเวอร์จะสร้างบัฟเฟอร์ของหน่วยความจำที่ใช้สำหรับการเก็บข้อมูลเมื่อทำการส่งหรือรับข้อมูลจากอุปกรณ์USB การติดต่อแต่ละครั้งระหว่างUSBไคลเอนต์และอุปกรณ์USBจะติดต่อผ่านคอมมิวนิเคชั่นไปป์ (communication pipe) ซึ่งถูกกำหนดในการตั้งค่าบนUSBซิสเต็มซอฟท์แวร์ USBซิสเต็มซอฟท์แวร์แบ่งคำขอของไคลเอนต์แต่ละตัวตามความต้องการใช้แบนด์วิชคำขอจะถูกส่งผ่านไปยังUSBโฮสต์คอนโทรลเลอร์ไดรเวอร์ โดยที่USBโฮสต์คอนโทรลเลอร์ไดรเวอร์จัดการแบ่งเวลาการติดต่อของไคลเอนต์แต่ละตัว โฮสต์คอนโทรลเลอร์จะจัดการแบ่งการติดต่อตามตารางเดสคริปเตอร์ที่สร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยUSBโอสต์คอนโทรลเลอร์ไดรเวอร์ USBโฮสต์คอนโทรลเลอร์ไดรเวอร์จะต้องรู้ข้อมูลต่างๆที่สำคัญของการติดต่อแต่ละครั้ง โดยที่ข้อมูลที่สำคัญบนเดสคริปเตอร์นั้นประกอบไปด้วย

- แอดเดรสของอุปกรณ์USBที่ต้องการติดต่อ
- ชนิดของการติดต่อ
- ขนาดของแพคเกจ
- ตำแหน่งของบัฟเฟอร์บนไคลเอนต์

โฮสต์คอนโทรลเลอร์สร้างการติดต่อสื่อสารที่ถูกกำหนดด้วยเดสคริปเตอร์ การติดต่อแต่ละครั้ง ข้อมูลที่ถูกรับจะถูกเก็บไว้ในบัฟเฟอร์ขึ้นอยู่กับทิศทางของการส่ง ว่าส่งจากไคลเอนต์หรือส่งจากอุปกรณ์ เมื่อการข้อมูลส่งผ่านเรียบร้อยแล้ว USBซิสเต็มซอฟท์แวร์จะส่งข้อมูลบอกกับไคลเอนต์ไดรเวอร์

2.4.3 ชนิดของข้อมูล (Transfers)

การส่งข้อมูลของUSBถูกออกแบบมากับอุปกรณ์หลายๆชนิด อุปกรณ์แต่ละชนิดมีลักษณะของข้อมูลที่ต้องการต่างกัน เช่นเมื่อส่งข้อมูลให้กับลำโพง การส่งข้อมูลจะต้องติดต่อกันและมีอัตราการส่งคงที่เพื่อไม่ให้เกิดการขาดหายเสียง อุปกรณ์แต่ละชนิดต้องการลักษณะของการรับส่งแตกต่างกันไป ชนิดของข้อมูลของการสื่อสาร ที่USBรองรับประกอบด้วย

- ข้อมูลแบบตามเวลา (Isochronous Transfers)
- ข้อมูลแพคเกจใหญ่ (Bulk Transfers)
- ข้อมูลอินเตอร์รัปต์ (Interrupt Transfer)
- ข้อมูลควบคุม (Control Transfers)

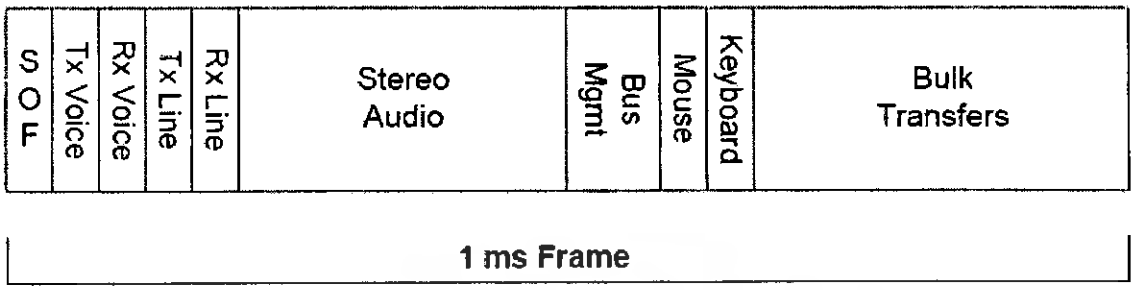
ชนิดของข้อมูลเหล่านี้จะอยู่บนไคลเอนต์ไดรเวอร์ โดยที่จะพิจารณาจากเดสคริปเตอร์ของอุปกรณ์

2.4.4 USB ไดรเวอร์, ไออาร์พี และ เฟรม (USB Driver, IRPs and Frames)

เมื่อไคลเอนต์ไดรเวอร์ต้องการที่จะส่งข้อมูล จะส่งคำขอไปยัง USB ไดรเวอร์ คำขอที่ส่งไปนี้เรียกว่า ไออาร์พี (IO Request packet : IRP) บางครั้งเมื่อต้องการส่งข้อมูลขนาดใหญ่ หรือขณะนั้นUSB ถูกแบ่งไปใช้กับอุปกรณ์อื่นด้วย การส่งข้อมูลครั้งเดียวไม่สามารถที่จะทำได้ การส่งข้อมูลจะแบ่งข้อมูลขนาดใหญ่ออกเป็นแพคเกจขนาดเล็กโดยใช้เวลาในการส่งที่นานขึ้น ทำให้มั่นใจได้ว่าแบนด์วิชของ USB สามารถรองรับอุปกรณ์ อื่นๆได้ด้วยการส่งข้อมูลของUSBมีเวลามาตรฐาน 1 มิลลิวินาที เรียกว่า เฟรม (Frame) อุปกรณ์แต่ละชนิดจะ จองพื้นที่ใน 1 มิลลิวินาทีเพื่อส่งข้อมูลของตัวเอง การจองแบนด์วิชจะขึ้นอยู่กับทรูพุทที่ต้องการของอุปกรณ์ (ถูกกำหนดในเดสคริปเตอร์ของอุปกรณ์) และขึ้นอยู่กับแบนด์วิชที่เหลืออยู่ของUSBที่ไม่มีอุปกรณ์อื่นใช้งาน USB บีซิสเต็มซอฟท์แวร์อ่านค่าเดสคริปเตอร์ของอุปกรณ์แต่ละชนิดและคำนวณแบนด์วิชทั้งหมดที่ต้องการใช้ ถ้าซิสเต็มซอฟท์แวร์ตรวจสอบและพบว่าแบนด์วิชสามารถส่งได้ก็จะทำการส่งตามคำขอ แต่ถ้าแบนด์วิชไม่เพียงพอตามคำขอเช่น มีอุปกรณ์อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้งานอยู่แล้ว อุปกรณ์ชนิดที่ส่งคำขอนั้นจะไม่สามารถทำงานได้ โดยจะมีการแจ้งเตือนให้ผู้ใช้ทราบด้วย



รูปที่ 2.5 ตัวอย่างเฟรมของยูเอสบี

2.4.5 โฮสต์คอนโทรลเลอร์ไดรเวอร์ (Host Controller Driver)

โฮสคอนโทรลเลอร์ไดรเวอร์รับแพคเกจร้องขอจากUSBไดรเวอร์ จากนั้นทำการจัดเรียงข้อมูล โดยการจัดเรียงข้อมูลจะขึ้นอยู่กับอัลกอริธึมของโฮสคอนโทรลเลอร์ โฮสคอนโทรลเลอร์ไดรเวอร์จะทำการสร้างโครงสร้างข้อมูลแบบอนุกรม จากนั้นส่งไปให้โฮสคอนโทรลเลอร์ โฮสคอนโทรลเลอร์อ่านข้อมูลที่รับมาแล้วการเอ็กซิคิว

2.4.5.1 โฮสต์คอนโทรลเลอร์ และแพคเกจ (Host Controller and Packets)

โฮสคอนโทรลเลอร์ และรูทฮับ สร้างการติดต่อที่เรียกว่าทรานเซคชั่น (Transaction) ทรานเซคชั่นประกอบไปด้วยการเรียงต่อกันของแพคเกจ โดยทั่วๆไปจะประกอบไปด้วย โทเคนแพคเกจ ดาต้าแพคเกจ แฮนด์เช็คแพคเกจ

2.4.5.2 เฟรมเวิร์กของอุปกรณ์ (Device Framework)

เฟรมเวิร์กของUSBประกอบไปด้วย 3 ลำดับชั้น

1. USB บัสอินเตอร์เฟสเลเยอร์ (USB Bus Interface Layer)

USB อินเตอร์เฟสเลเยอร์ ประกอบไปด้วย

- การติดต่อทางกายภาพ (physical connection)
- สัญญาณทางไฟฟ้า (electrical signaling environment)
- กระบวนการส่งแพคเกจ (packet transfer mechanisms)

ในชั้นนี้แทนการส่งข้อมูลจริง ผ่านสายUSB ระหว่างโฮสซิสเต็มกับอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

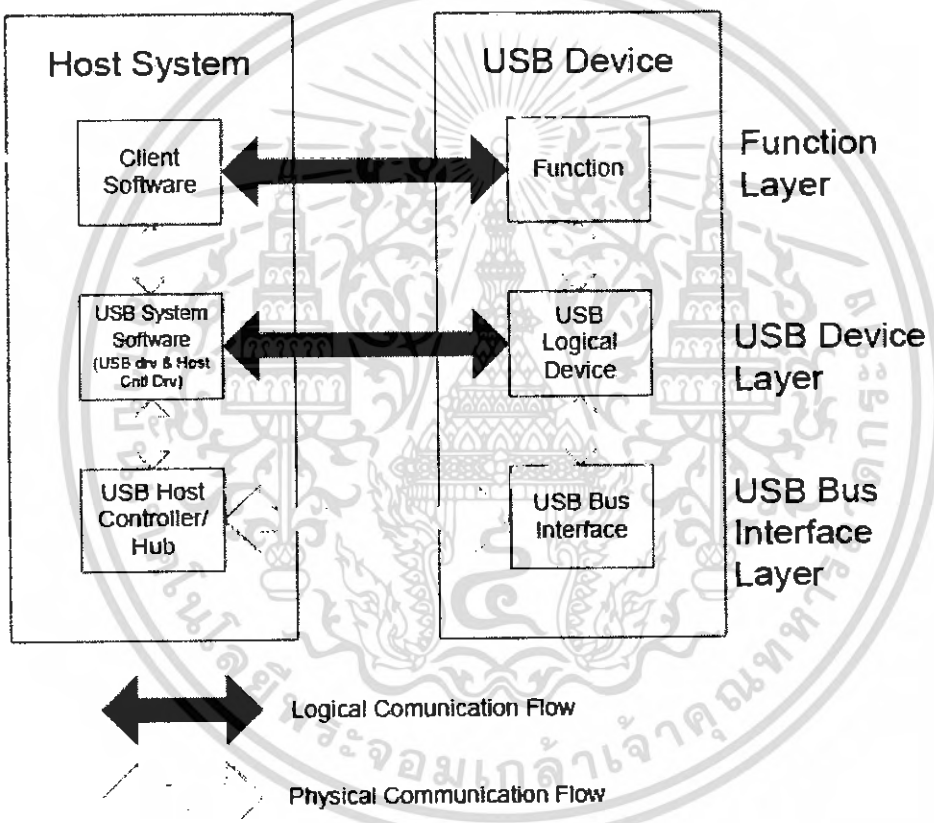
2. USB ดีไวซ์เลเยอร์ (USB Device Layer)

ในชั้นนี้ประกอบไปด้วย USBซิสเต็มซอฟท์แวร์ บนโฮส และโลจิคอลวิวบนอุปกรณ์ USBซิสเต็มซอฟท์แวร์สามารถมองเห็นโลจิคอลดีไวซ์ โดยเห็นเป็นกลุ่มของอุปกรณ์ USBซิสเต็มซอฟท์แวร์

ประกอบไปด้วย USBไดรเวอร์ และUSBโฮสต์คอนโทรลเลอร์ไดรเวอร์

3. ฟังก์ชั่นเลเยอร์ (Function Layer)

ในชั้นนี้ครอบคลุมถึงไคลเอนต์ซอฟท์แวร์ และฟังชั่นการทำงานบนอุปกรณ์



รูปที่ 2.6 Device Framework

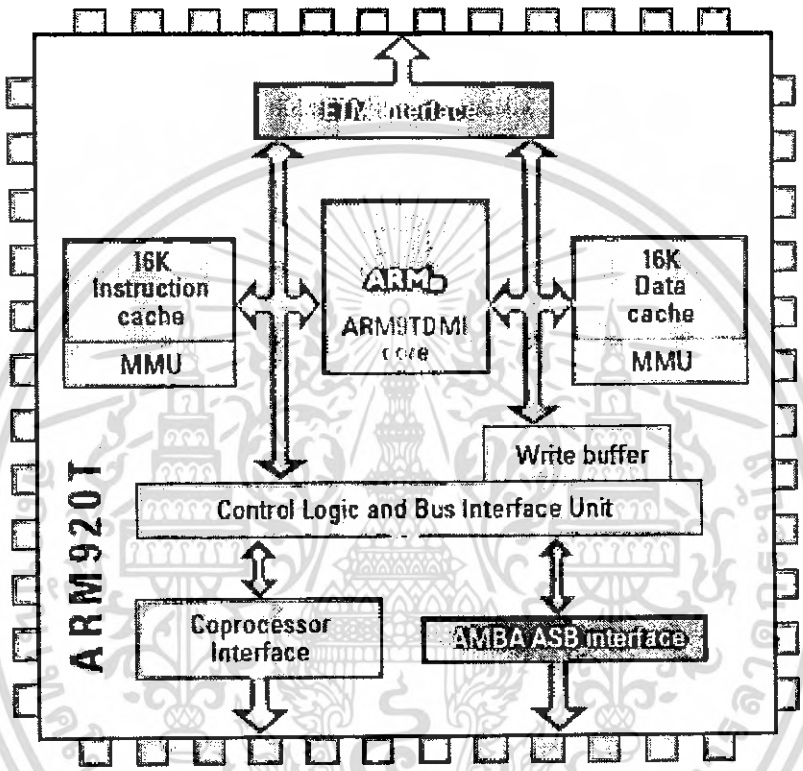
2.5 ARM (Advanced RISC Machine)

2.5.1 หน่วยประมวลผล ARM

ARM Acorn RISC Machineหรือ Advanced RISC Machineเป็นบริษัท ซึ่งออกแบบแกนของหน่วยประมวลผล (CPU Core) และขายให้กับบริษัทผู้ผลิตแผงวงจรรวม (Integrated Circuit หรือ IC) ต่าง ๆ รวมทั้งกำหนดมาตรฐานและชุดคำสั่ง รวมทั้งจัดทำชุดพัฒนาออกขาย โดยบริษัท ARM จะไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลิตหน่วยประมวลผลเอง บริษัท ARM ได้ออกแบบแกนของหน่วยประมวลผลมาหลาย รุ่นโดยรุ่นที่เป็นที่รู้จักและได้รับความนิยมมากทั้งในในอดีตและปัจจุบัน ได้แก่ ARM6, ARM7, ARM9 ARM10 และ ARM11 ซึ่งโดยส่วนใหญ่จะมีขนาดของชุดคำสั่งเป็น 32บิด และเป็นสถาปัตยกรรมแบบ RISC (Reduce Instruction Set Computer) และถูกนำไปผลิตโดยผู้ผลิตวงจรรวมชั้นนำ เช่น Samsung, Sharp, Motorola (Free scale), Intel ฯลฯ



รูปที่ 2.7 โครงสร้างโดยรวมของ CPU ARM920T

Chip ของหน่วยประมวลผล ARM ที่ผลิตออกขายโดยบริษัทต่าง ๆ จะมีการนำวงจรของอุปกรณ์เชื่อมต่อเพิ่มเติม มารวมกับแกนของหน่วยประมวลผล ARM ในแผงวงจรรวม เดียวกัน กลายเป็นระบบที่เรียกว่า Microcontroller หรือว่า System on Chip (SoC) ทำให้ลดต้นทุนในการทำแผ่นวงจร (PCB) ที่ใช้ในการทำผลิตภัณฑ์ต่าง ๆ เช่น UART, USB, Ethernet MAC ฯลฯ

2.5.2 ชุดคำสั่งของหน่วยประมวลผล ARM9

ชุดคำสั่งของ ARM9 เป็นสถาปัตยกรรมแบบ RISC ซึ่งค่อนข้างมี ชุดคำสั่งข้อมูล (Instruction Set) ขนาดคงที่ โดยขนาดของชุดคำสั่งมาตรฐาน จะมีขนาด 32 บิด สำหรับ ARM9 จะใช้ชุดคำสั่งแบบ ARMv4 แต่ในหน่วยประมวลผล ARM9 ได้มีการปรับปรุงขึ้นมาอีกขั้นโดยมีการเพิ่มความสามารถในการใช้ชุดคำสั่งขนาด 16 บิตได้ด้วย ทำให้สามารถทำงานได้เทียบเท่าชุดคำสั่ง 32 บิตด้วยขนาดของ

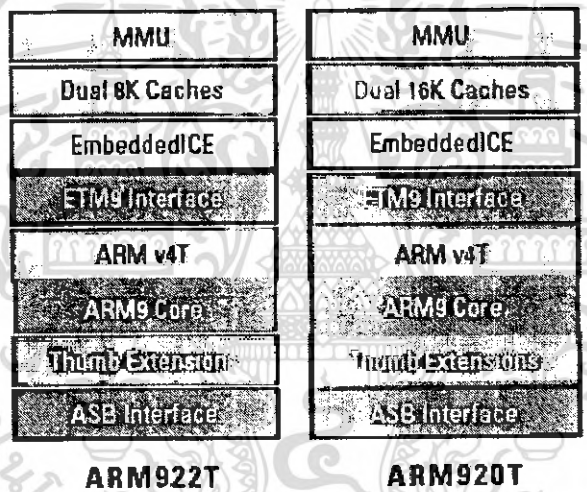
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ลดลง โดยหน่วยประมวลผล ARM9 ที่สามารถใช้ชุดคำสั่งขนาด 16 บิตได้ และเรียกชุดคำสั่งนั้นว่า Thumb และเรียกชุดคำสั่งที่ใช้งานได้ทั้ง 32/16 บิตว่า ARMv4t

2.5.3 รุ่นและความแตกต่างของหน่วยประมวลผล ARM9

ARM9 จะมีรูปแบบหน่วยประมวลแบ่งย่อยในแต่ละรุ่นลงไปอีก (เรียกว่า Core Version) เช่น ARM9 จะมี ARM920T ARM922T เป็นต้น ซึ่ง แต่ละรุ่นจะมีรายละเอียดแตกต่างกันออกไป มีความซับซ้อนและจุดประสงค์ต่างกันด้วย เช่น

- แคชของชุดคำสั่ง (Instruction Cache หรือ I-Cache)
- แคชของข้อมูล (Data Cache หรือ D-Cache)
- ตัวจัดการหน่วยความจำ (Memory Management Unit หรือ MMU)
- หน่วยประมวล DSP และ FPU (Digital Signal Processing, Floating Point Unit)



รูปที่ 2.8 ข้อมูลความแตกต่างระหว่าง CPU ARM922T และ ARM920T

แกนหน่วยประมวลผล	ชุดคำสั่ง	I-Cache D-Cache	MMU	Frequency MHz(worst case)
ARM922T	Armv4	8k	MMU	200
ARM920T	Armv4t	16k	MMU	200

ตารางที่ 2.5 ตารางแสดงจุดเด่นในละรุ่นของหน่วยประมวลผล ARM9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

72094

2.6 ระบบปฏิบัติการ อินุกซ์ (OS Linux)

2.6.1 ประวัติและที่มาของระบบปฏิบัติการ ลินุกซ์

ลินุกซ์ (Linux) เป็นระบบปฏิบัติการและชื่อเคอร์เนล (Kernel) ที่นิยมตัวหนึ่งในฐานะของซอฟต์แวร์โอเพนซอร์ส ระบบปฏิบัติการลินุกซ์หรือที่เรียกเต็มๆว่า "กนู/ลินุกซ์" (GNU/Linux) มีลักษณะคล้ายกับระบบปฏิบัติการยูนิกซ์ ตามมาตรฐาน POSIX โดยมีลินุกซ์ เคอร์เนล เป็นศูนย์กลางทำงานร่วมกับไลบรารีและเครื่องมือต่างๆ ลินุกซ์นิยมจำหน่ายหรือแจกฟรีในลักษณะเป็นแพคเกจ โดยผู้จัดทำจะรวมซอฟต์แวร์สำหรับใช้งานในด้านต่างๆ เป็นชุดเข้าด้วยกัน

ลินุกซ์ ถูกพัฒนาขึ้นโดยนาย ไลนัส โทวอลด์ (Linux Tovalds) โดยในช่วงแรก ๆ ลินุกซ์ ถูกพัฒนาและใช้งานในเฉพาะกลุ่มผู้ที่สนใจ ซึ่งในปัจจุบันลินุกซ์ได้รับความนิยมเนื่องมาจากระบบการทำงานที่เป็นอิสระ ปลอดภัย เชื่อถือได้ และราคาต่ำ จึงได้มีการพัฒนาจากองค์กรต่างๆ เช่น ไอบีเอ็ม ฮิวเลตต์-แพกการ์ด และ โนเวลล์ใช้สำหรับในระบบเซิร์ฟเวอร์และพีซี เริ่มแรกลินุกซ์พัฒนาสำหรับใช้กับเครื่อง อินเทล 386 (Intel 386)

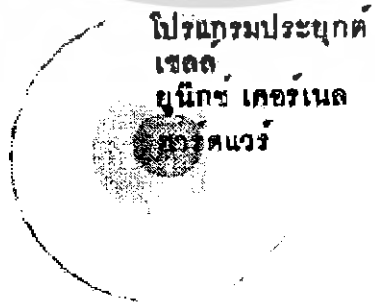
ลิขสิทธิ์ของหลายๆส่วนของ ลินุกซ์จดภายใต้ ลิขสิทธิ์GPL ซึ่งเป็นลิขสิทธิ์ที่กำหนดให้ผู้ที่นำโค้ดไปใช้ต้องใช้ลิขสิทธิ์แบบเดิมต่อคือใช้ลิขสิทธิ์ GPL เช่นเดียวกัน (ในบางครั้งจะเรียกว่า copy left) ซึ่งเรียกซอฟต์แวร์ประเภทนี้ว่า รหัสเปิด (Open Source)

2.6.2 การนำลีนุกซ์มาใช้งานบน Embedded Board

เนื่องจากพื้นที่จำกัดบน Embedded Board ทำให้เราจำเป็นต้องสร้าง Core Linux และชุดคำสั่งที่มีขนาดเล็กที่สามารถทำงานบน Embedded Board ได้ทำให้เราต้องสร้าง zImage เพื่อเป็น Core Linux สำหรับ Board และ rootfs ที่เก็บชุดคำสั่งที่เราจะใช้งานบน Board ก่อนและ Load File เหล่านี้ลงไปใน Embedded Board เพื่อให้ Board สามารถใช้คำสั่งต่างๆ ได้

2.6.3 โครงสร้างของ Unix

ระบบปฏิบัติการ Unix สามารถแบ่งโครงสร้างหลัก ๆ ได้ 4 ระดับ แต่ละระดับก็จะทำหน้าที่ต่างกัน



รูปที่ 2.9 โครงสร้างของ Unix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3.1 ฮาร์ดแวร์

หมายถึงอุปกรณ์หรือทุกชิ้นส่วนของคอมพิวเตอร์ ที่เราสามารถจับต้องได้ เช่น จอภาพ คีย์บอร์ด เม้าส์ ดิกส์ไดรซ์ ซีดีรอม เป็นต้น

2.6.3.2 ยูนิกซ์ เคอร์เนล

เคอร์เนล จะทำหน้าที่ควบคุมการทำงานทั้งหมดของระบบ ได้แก่ การจัดสรรทรัพยากร การจัดการข้อมูลบริการหน่วยความจำ ซึ่งเคอร์เนลนี้จะขึ้นกับฮาร์ดแวร์ เช่น ถ้ามีการเปลี่ยนแปลงฮาร์ดแวร์ เคอร์เนลนี้ก็จะถูกเปลี่ยนไปด้วย เป็นต้น

2.6.3.3 เชลล์

คือ ตัวกลางระหว่างผู้ใช้กับตัวเคอร์เนล ทำหน้าที่รับคำสั่งจากผู้ใช้ แล้วนำไปแปลเป็นภาษาที่เครื่องคอมพิวเตอร์เข้าใจ เราเรียกอีกอย่างหนึ่งได้ว่า command interpreter แต่ถ้ามีการนำ เชลล์หลาย ๆ ตัวมาเขียนรวมกัน (คล้าย ๆ กับ batch file ในระบบปฏิบัติการ DOS) เราจะเรียกว่า เชลล์สคริปต์ นอกจากนี้ เชลล์ (Shell) ยังมีอีกหลาย ๆ ประเภท แต่ที่นิยมกันได้แก่

Bourne shell (sh) เป็นเชลล์ดั้นแบบของทุก ๆ เชลล์ มีความสามารถในการเขียน เชลล์สคริปต์ได้ด้วย

C shell (csh) เป็นเชลล์ที่สร้างหลัง Bourne shell ความสามารถพิเศษของ C shell คือเก็บข้อมูลเกี่ยวกับคำสั่งที่เคยใช้

Korn shell (ksh) ซึ่งพัฒนาโดย AT&T โดยได้นำคุณสมบัติเด่น ๆ ของ Bourne shell และ C shell มารวมกัน

Bourne again shell (bash) มีคุณสมบัติและความสามารถคล้ายกับ Korn shell แต่ shell นี้ถูกสร้างขึ้นมาใช้สำหรับแจกฟรี ซึ่งเป็นเหตุผลสำคัญที่ทาง Linux นำมาใช้

คำสั่งที่ทำให้ทราบว่าเราใช้งาน เชลล์อะไรอยู่คือ echo $SHELL

2.6.3.4 โปรแกรมประยุกต์

หมายถึง โปรแกรมการใช้งานเพิ่มเติมต่าง ๆ ที่ใช้บนระบบปฏิบัติการยูนิกซ์ ได้แก่ pine เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ระบบปฏิบัติการ อินุกซ์ (OS Linux)

2.6.1 ประวัติและที่มาของระบบปฏิบัติการ ลินุกซ์

ลินุกซ์ (Linux) เป็นระบบปฏิบัติการและชื่อเคอร์เนล (Kernel) ที่นิยมตัวหนึ่งในฐานะของซอฟต์แวร์โอเพนซอร์ส ระบบปฏิบัติการลินุกซ์หรือที่เรียกเต็มๆว่า "กนู/ลินุกซ์" (GNU/Linux) มีลักษณะคล้ายกับระบบปฏิบัติการยูนิกซ์ ตามมาตรฐาน POSIX โดยมีลินุกซ์ เคอร์เนล เป็นศูนย์กลางทำงานร่วมกับไลบรารีและเครื่องมือต่างๆ ลินุกซ์นิยมจำหน่ายหรือแจกฟรีในลักษณะเป็นแพคเกจ โดยผู้จัดทำจะรวมซอฟต์แวร์สำหรับใช้งานในด้านต่างๆ เป็นชุดเข้าด้วยกัน

ลินุกซ์ ถูกพัฒนาขึ้นโดยนาย ไลนัส โทวอลด์ (Linux Tovalds) โดยในช่วงแรก ๆ ลินุกซ์ ถูกพัฒนาและใช้งานในเฉพาะกลุ่มผู้ที่สนใจ ซึ่งในปัจจุบันลินุกซ์ได้รับความนิยมเนื่องมาจากระบบการทำงานที่เป็นอิสระ ปลอดภัย เชื่อถือได้ และราคาต่ำ จึงได้มีการพัฒนาจากองค์กรต่างๆ เช่น ไอบีเอ็ม ฮิวเลตต์-แพกการ์ด และ โนเวลล์ใช้สำหรับในระบบเซิร์ฟเวอร์และพีซี เริ่มแรกลินุกซ์พัฒนาสำหรับใช้กับเครื่อง อินเทล 386 (Intel 386)

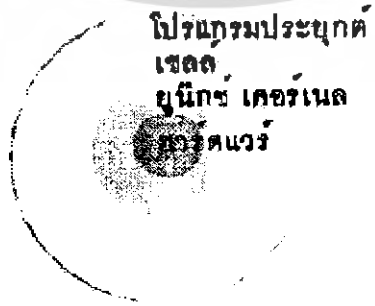
ลิขสิทธิ์ของหลายๆส่วนของ ลินุกซ์จดภายใต้ ลิขสิทธิ์GPL ซึ่งเป็นลิขสิทธิ์ที่กำหนดให้ผู้ที่นำโค้ดไปใช้ต้องใช้ลิขสิทธิ์แบบเดิมต่อคือใช้ลิขสิทธิ์ GPL เช่นเดียวกัน (ในบางครั้งจะเรียกว่า copy left) ซึ่งเรียกซอฟต์แวร์ประเภทนี้ว่า รหัสเปิด (Open Source)

2.6.2 การนำลีนุกซ์มาใช้งานบน Embedded Board

เนื่องจากพื้นที่จำกัดบน Embedded Board ทำให้เราจำเป็นต้องสร้าง Core Linux และชุดคำสั่งที่มีขนาดเล็กที่สามารถทำงานบน Embedded Board ได้ทำให้เราต้องสร้าง zImage เพื่อเป็น Core Linux สำหรับ Board และ rootfs ที่เก็บชุดคำสั่งที่เราจะใช้งานบน Board ก่อนและ Load File เหล่านี้ลงไปใน Embedded Board เพื่อให้ Board สามารถใช้คำสั่งต่างๆ ได้

2.6.3 โครงสร้างของ Unix

ระบบปฏิบัติการ Unix สามารถแบ่งโครงสร้างหลัก ๆ ได้ 4 ระดับ แต่ละระดับก็จะทำหน้าที่ต่างกัน



รูปที่ 2.9 โครงสร้างของ Unix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3.1 ฮาร์ดแวร์

หมายถึงอุปกรณ์หรือทุกชิ้นส่วนของคอมพิวเตอร์ ที่เราสามารถจับต้องได้ เช่น จอภาพ คีย์บอร์ด เม้าส์ ดิกส์ไดรซ์ ซีดีรอม เป็นต้น

2.6.3.2 ยูนิกซ์ เคอร์เนล

เคอร์เนล จะทำหน้าที่ควบคุมการทำงานทั้งหมดของระบบ ได้แก่ การจัดสรรทรัพยากร การจัดการข้อมูลบริการหน่วยความจำ ซึ่งเคอร์เนลนี้จะขึ้นกับฮาร์ดแวร์ เช่น ถ้ามีการเปลี่ยนแปลงฮาร์ดแวร์ เคอร์เนลนี้ก็จะถูกเปลี่ยนไปด้วย เป็นต้น

2.6.3.3 เชลล์

คือ ตัวกลางระหว่างผู้ใช้กับตัวเคอร์เนล ทำหน้าที่รับคำสั่งจากผู้ใช้ แล้วนำไปแปลเป็นภาษาที่เครื่องคอมพิวเตอร์เข้าใจ เราเรียกอีกอย่างหนึ่งได้ว่า command interpreter แต่ถ้ามีการนำ เชลล์หลาย ๆ ตัวมาเขียนรวมกัน (คล้าย ๆ กับ batch file ในระบบปฏิบัติการ DOS) เราจะเรียกว่า เชลล์สคริปต์ นอกจากนี้ เชลล์ (Shell) ยังมีอีกหลาย ๆ ประเภท แต่ที่นิยมกันได้แก่

Bourne shell (sh) เป็นเชลล์ดั้นแบบของทุก ๆ เชลล์ มีความสามารถในการเขียน เชลล์สคริปต์ได้ด้วย

C shell (csh) เป็นเชลล์ที่สร้างหลัง Bourne shell ความสามารถพิเศษของ C shell คือเก็บข้อมูลเกี่ยวกับคำสั่งที่เคยใช้

Korn shell (ksh) ซึ่งพัฒนาโดย AT&T โดยได้นำคุณสมบัติเด่น ๆ ของ Bourne shell และ C shell มารวมกัน

Bourne again shell (bash) มีคุณสมบัติและความสามารถคล้ายกับ Korn shell แต่ shell นี้ถูกสร้างขึ้นมาใช้สำหรับแจกฟรี ซึ่งเป็นเหตุผลสำคัญที่ทาง Linux นำมาใช้

คำสั่งที่ทำให้ทราบว่าเราใช้งาน เชลล์อะไรอยู่คือ echo $SHELL

2.6.3.4 โปรแกรมประยุกต์

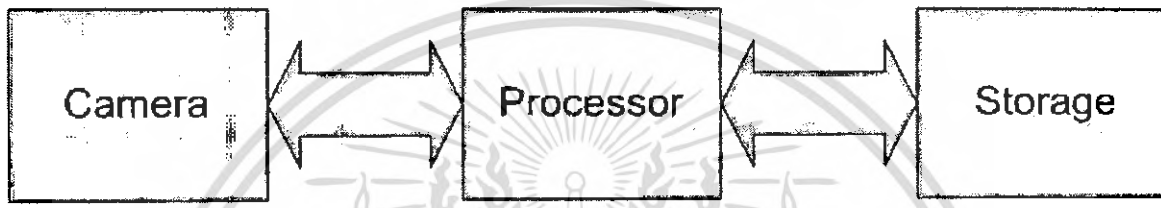
หมายถึง โปรแกรมการใช้งานเพิ่มเติมต่าง ๆ ที่ใช้บนระบบปฏิบัติการยูนิกซ์ ได้แก่ pine เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและพัฒนา

3.1 แนวคิดในการออกแบบโดยรวม

ในส่วนของ Hard Ware สามารถมองภาพการทำงานของระบบโดยรวมออกเป็นส่วนหลักๆอยู่ 3 ส่วนด้วยกัน คือ ส่วนของการรับภาพ ส่วนการประมวลผลข้อมูล ส่วนการเก็บข้อมูลโดยจะแสดงดังรูปที่ 3.1



รูปที่ 3.1 แสดง บล็อกไดอะแกรมโดยรวม

จากรูปสามารถวิเคราะห์การทำงานในแต่ละส่วนได้ดังนี้

3.1.1 ในบล็อกแรกเป็นการกล่าวถึงระบบของการรับสัญญาณภาพ โดยระบบจะทำการรับสัญญาณภาพที่เป็นสัญญาณอนาล็อกทำการแปลงข้อมูลให้กลายเป็นสัญญาณดิจิตอลเพื่อใช้ในการเก็บข้อมูลและเป็นรูปแบบที่คอมพิวเตอร์สามารถเข้าใจได้ ซึ่งในส่วนนี้ไม่จำเป็นต้องสร้างวงจรหรือประกอบขึ้นเองทั้งหมด เพราะในส่วนนี้มีระบบกล้องถ่ายภาพซึ่งสามารถทำงานได้ตามที่ระบบต้องการอยู่แล้ว ดังนั้นจึงได้วิเคราะห์ และสรุปออกมาว่าควรจะใช้งานกล้อง Web Camera เพราะมีการแปลงภาพให้เป็นระบบดิจิตอลอยู่ในระบบแล้ว และมีการส่งข้อมูลภาพผ่านทาง พอร์ต USB ซึ่งในปัจจุบันมีการใช้งานกันอย่างแพร่หลาย และยังง่ายต่อการศึกษาหาข้อมูลอีกด้วย

3.1.2 ในบล็อกที่สอง เป็นส่วนของการประมวลผลข้อมูล โดยในระบบส่วนนี้จะเป็นหัวใจหลักของการทำงานทั้งหมด เพราะเป็นส่วนที่ใช้ติดต่อควบคุมการรับภาพ และยังเป็นส่วนที่ทำการเลือกว่าจะทำการเก็บข้อมูลในลักษณะใดและที่ใดดังนั้นจึงจำเป็นที่จะต้องใช้ตัวประมวลผลที่มีประสิทธิภาพสูง

จากการศึกษาและเปรียบเทียบหน่วยประมวลผลซึ่งมีอยู่หลายตระกูล ดังที่จะกล่าวต่อไปนี้ เพื่อการตัดสินใจเลือกใช้หน่วยประมวลผลที่มีประสิทธิภาพและราคาเหมาะสมกับงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.1 Microcontroller ตระกูล MCS 51 เป็นหน่วยประมวลผลที่มีความเร็วสูงสุดที่ 40 MHz ซึ่งไม่พอที่จะนำมาใช้ในการประมวลผลข้อมูลที่ต้องการความเร็วสูงอย่างข้อมูลภาพที่ถูกส่งมาจากกล้อง

3.1.2.2 Field Programmable Gate Array หรือ FPGA เป็นหน่วยประมวลผลที่มีความเร็วสูงสุดอยู่ที่ 200 MHz ซึ่งพอที่จะนำมาใช้ในการประมวลผลข้อมูลที่ต้องการความเร็วสูง แต่การที่จะใช้ FPGA นั้นจำเป็นที่จะต้องออกแบบวงจรที่ยุ่งยากมาก เพื่อให้สามารถติดต่อกับกล้องและฮาร์ดดิสก์ได้ รวมไปถึงเราไม่สามารถสร้าง Hardware เองได้ อีกทั้งไม่มีบริษัทใดในประเทศไทยที่สามารถสร้าง Hardware ที่เป็นแผ่น PCB สำหรับ FPGA ได้ ดังนั้นการใช้ FPGA เป็นหน่วยประมวลผลในงานเฉพาะทางเช่นนี้จึงไม่เหมาะสม และถ้าจะต้องการทำจริงนั้นต้องใช้ค่าใช้จ่ายสูงมาก โอกาสที่งานจะสำเร็จทันเวลาก็น้อยมากด้วย

3.1.2.3 Advanced RISC Machine หรือ ARM เป็นหน่วยประมวลผลที่มีความเร็วสูงสุดอยู่ที่ 400 MHz ซึ่งเหมาะที่จะนำมาใช้ในการประมวลผลข้อมูลที่ต้องการความเร็วสูง และในท้องตลาดก็มักจะมี Embedded Board สำเร็จรูปที่มี ARM เป็นหน่วยประมวลผลอยู่แล้ว เราจึงสามารถเลือก Embedded Board เหล่านั้นมาใช้งานได้เลย โดยเลือกบอร์ดที่มีคุณสมบัติครบตามลักษณะของงาน ดังนั้นการเลือกใช้ Embedded Board ที่มี ARM เป็นหน่วยประมวลผลมาใช้ในงานนี้จึงมีความเป็นไปได้สูงมาก

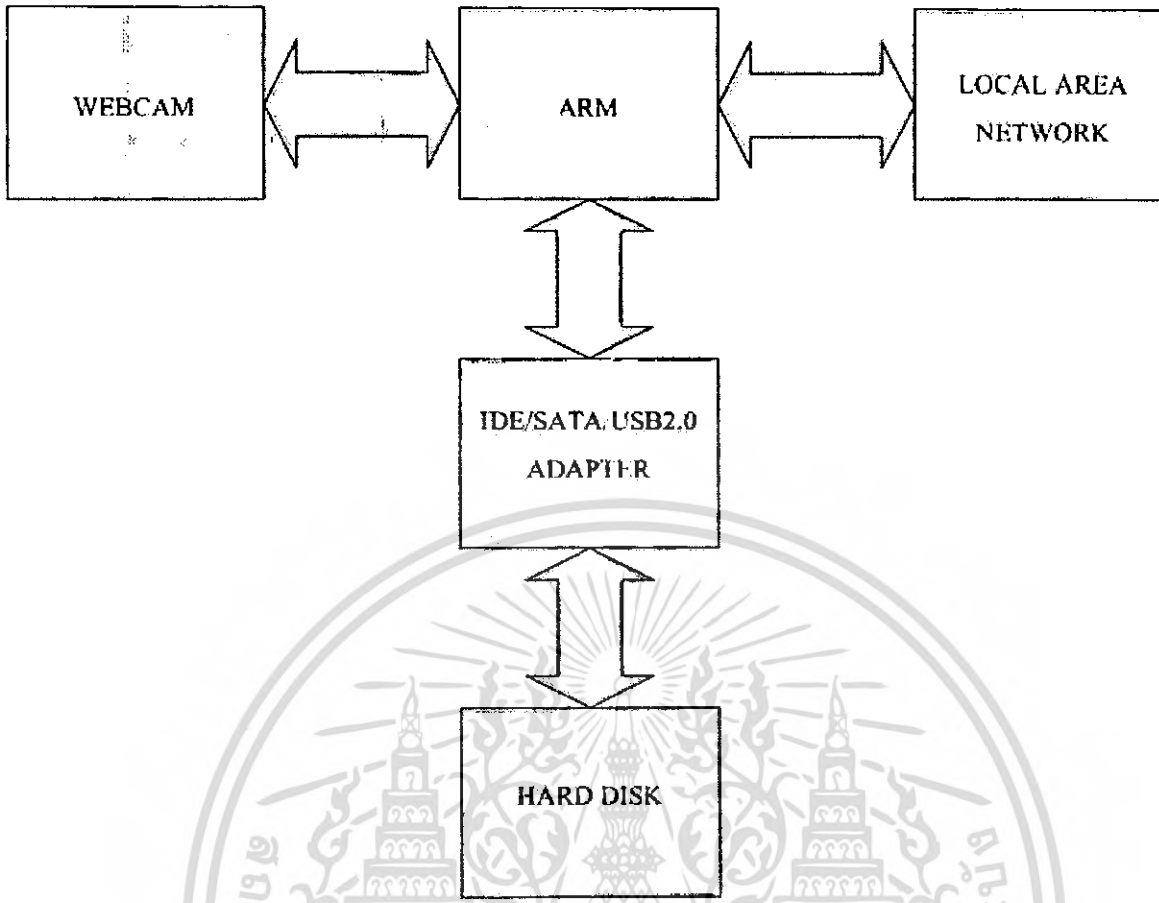
3.1.2.4 CPU เป็นหน่วยประมวลผลที่มีความเร็วสูงสุดถึง 4 GHz ซึ่งเหมาะที่จะนำมาใช้ในการประมวลผลข้อมูลที่ต้องการความเร็วสูง และในท้องตลาดก็มีมาก แต่ CPU นั้นมีความร้อนสูง และบอร์ดที่ใช้งานก็มีขนาดใหญ่ไม่สะดวกที่จะเคลื่อนย้าย

ซึ่งในที่นี้ได้เลือกใช้ตัวประมวลผล ARM ซึ่งมีคุณสมบัติตามที่ต้องการครบ และมีโครงสร้างลักษณะการเชื่อมต่อกับอุปกรณ์ภายนอกที่หลากหลาย อีกทั้งยังสามารถใช้งานในส่วนของการเชื่อมต่อกับระบบ Network และอื่นๆ โดยที่เราอาจจะนำคุณสมบัติเหล่านั้นมาทำการพัฒนาเพิ่มเติมได้ในอนาคต

3.1.3 บล็อกที่สามเป็นส่วนที่ใช้ในการบันทึกข้อมูล โดยในระบบนี้เราได้วิเคราะห์แล้วว่าเราสามารถใช้งานอุปกรณ์ช่วยในการแปลงข้อมูลในมาตรฐานการเก็บข้อมูลที่มีอยู่เดิมให้เป็นมาตรฐาน USB ได้ เพื่อลดความยุ่งยากในการติดต่อเพราะ ตัวประมวลผลที่มีอยู่นั้นมีการรองรับการใช้งานให้แล้ว

หลังจากทำการวิเคราะห์แล้วเราจะได้ลักษณะการทำงานดังแสดงในรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดง บล็อกไดอะแกรม ระบบการทำงานหลัก

3.2 ขั้นตอนการจัดเตรียมอุปกรณ์

จากการที่ได้วิเคราะห์และออกแบบทำให้เราทราบว่า ควรใช้อุปกรณ์อะไรบ้างในการทำงาน โดยอ้างอิงจากบล็อกไดอะแกรมที่ออกแบบไว้และสามารถอธิบายการทำงานได้ดังนี้

1 ARM

เนื่องจากการใช้งาน CPU ARM ไม่สามารถนำมาทดสอบและใช้งานได้เองโดยการต่อวงจรหรือ Interface เพื่อใช้งานเองได้เพราะ มีลักษณะที่ซับซ่อน ดังนั้นจึงนิยมใช้งานบอร์ดที่ใช้ในการทดลอง หรือ Development board ในการพัฒนาโครงงานนี้ ซึ่งจากการออกแบบนั้นได้กำหนดไว้ว่าบอร์ดจะต้องมีการใช้งาน usb port ซึ่งจากการค้นคว้าแล้วทำให้ทราบว่า บริษัท ภายในประเทศ กำลังพัฒนาและใช้งานอยู่โดยมีคุณสมบัติดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Embedded Board รุ่น ALTAIR 255 ของบริษัท Thaigertec
2. Embedded Board รุ่น JAMP Mini ของบริษัท Design Gateway

โดยที่ Embedded Board ทั้งสองนั้นมีคุณสมบัติที่แตกต่างกันอย่างเห็นได้ชัด ดังที่ได้แสดงในตารางด้านล่าง

Embedded Board	**ALTAIR 255**	**JAMP Mini**
CPU	Intel® XScale™ รุ่น PXA255 200 - 400 MHz	ARM920T รุ่น EP9302 200 MHz
SDRAM	64 MBytes	32 MBytes
Flash	32 MBytes (64 MByte opt.)	32 MBytes
FPGA	Xilinx® Virtex-II™	-
Ethernet	10/100Base-TX, full/half duplex	10/100Base-TX, full/half duplex
USB Interface	USB 2.0 On-The-Go/host/device (Future) , USB 2.0	Two-port USB 2.0 Full-speed Host
Serial Interface	Serial Full	Serial Debug (only tx, rx, GND) , Serial Full
LCD Interface	Maximum 640x480x16 Bit/pixel , active (TFT) passive (DSTN)	-
GPIO	yes	yes
I2C	yes	yes
ADC	4 Channels	5 Channels
Power	12 Volt 1A.	5 Volt 1A.
ราคา	50,000 – 60,000 บาท	5000 - 7000 บาท

ตารางที่ 3.1 แสดงและเปรียบเทียบคุณสมบัติของบอร์ดเพื่อใช้ทำการตัดสินใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.3 แสดง ลักษณะของ ชิพ ARM และ Embedded Board

2 WEB CAMERA

จากข้อมูลของบอร์ด ARM ที่กล่าวไปในหัวข้อข้างตน ทำให้ทราบว่าบอร์ดมีการรองรับ USB และจากที่ทำการวิเคราะห์ข้างต้นทำให้เราเลือกใช้กล้องที่เป็น WEB CAM และซึ่งเราได้เลือกใช้ กล้อง ยี่ห้อ กล้อง Creative Webcam live ดังแสดงในรายละเอียดด้านล่าง

รูปที่ 3.4 แสดงลักษณะของกล้อง Creative Live! Webcam

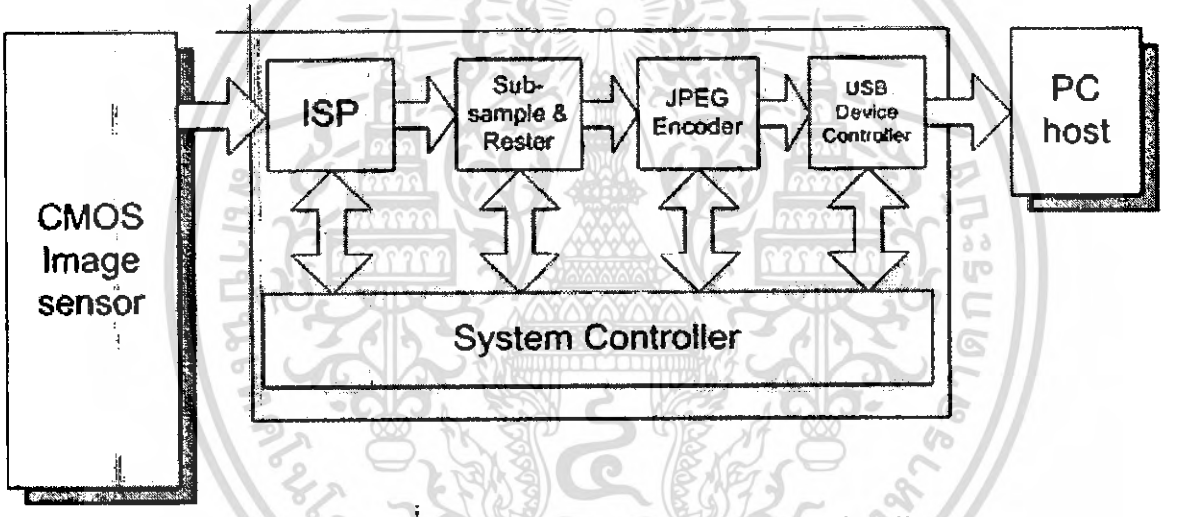
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของกล้อง

ความระเอียดสูงสุดของการถ่ายภาพนิ่ง	1024 X 768
ความระเอียดสูงสุดของการถ่ายภาพเคลื่อนไหว	640 X 480; 15 fps
ระยะโฟกัส	ปรับบริเวณหน้ากล้อง
การเชื่อมต่อ	USB 1.1 (Compatible with USB 2.0)

ตารางที่ 3.2 แสดงคุณสมบัติของกล้อง Creative Live! Webcam

ซึ่งจากข้อมูลที่ได้มาเราได้นำไปค้นคว้าหาข้อมูลต่อไปได้ว่า กล้องรุ่นนี้ใช้ชิพ รุ่น ZC301-2 ดังแสดงในรูปที่3.4 ทำให้เราสามารถใช้งานกล้องรุ่นนี้ได้เพราะมีคุณสมบัติตามที่ตั้งไว้แต่แรก



รูปที่ 3.5 บล็อกไดอะแกรม การทำงานภายในกล้อง

3 IDE/SATA/USB2.0 Adapter

เป็นอุปกรณ์ที่ช่วยในการทำงานชนิดหนึ่ง โดยได้นำเอาการเชื่อมต่อ มาตรฐาน IDE และ Serial ATA มาประยุกต์ใช้กับการเชื่อมต่อ USB ซึ่งจะทำการแปลงจาก IDE หรือ Serial ATA ให้สามารถเชื่อมต่อเข้ากับคอมพิวเตอร์ผ่าน มาตรฐาน USB โดยไม่จำเป็นต้องปิดเปิดเครื่องคอมพิวเตอร์ ขณะทำงาน ดังแสดงข้อมูลดังต่อไปนี้

- ใช้มาตรฐาน USB 2.0 ในการเชื่อมต่อ
- สามารถแปลงได้ทั้ง HDD IDE 3.5" ,2.5" / Serial ATA
- ใช้งาน HDD IDE 3.5 " หรือ Serial ATA ต้องทำการต่อไฟเลี้ยง
- เป็นระบบ Plug & Play

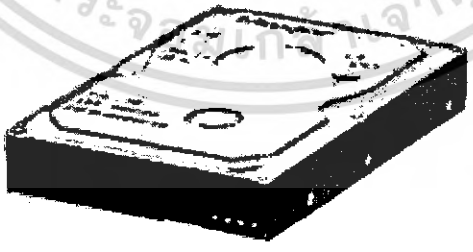
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ทั้งหมดที่มีให้ Adapter ที่ใช้แปลง

รูปที่ 3.6 แสดงลักษณะและอุปกรณ์ของ IDE/SATA/USB2.0 Adapter

4 Harddisk

เหตุผลในการเลือกใช้งาน Harddisk ในการเก็บข้อมูลนั้น ก็เพราะว่า Harddisk เป็นอุปกรณ์คอมพิวเตอร์ที่มีการใช้งานกันอย่างแผ่หลายในปัจจุบัน และมีราคาที่ถูกลงเรื่อยๆซึ่งสวนทางกับขนาดความจุของตัวอุปกรณ์ ที่มีแนวโน้มจะสูงขึ้นเรื่อยๆ ในส่วนของการเก็บข้อมูลด้วย Harddisk นั้นได้มีอุปกรณ์ช่วยในการเปลี่ยนรูปแบบการส่งข้อมูล ทำให้ไม่จำเป็นต้องในการสร้างโมดูลในการเชื่อมต่อเอง โดย Harddisk ที่เลือกใช้นั้น คือ Harddisk ในมาตรฐาน IDE ซึ่งในปัจจุบันมีราคาถูกลงมากแล้ว



รูปที่ 3.7 แสดงลักษณะ Harddisk IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 แนวคิดในการออกแบบ ซอฟแวร์

แนวคิดในการออกแบบโปรแกรมนั้นเราได้อ้างอิงการทำงานบน ระบบ Linux เป็นหลัก โดยได้แบบการทำงานของโปรแกรมออกเป็น 3 ส่วนหลักได้แก่

3.2.1 แนวคิดการออกแบบการเชื่อมต่อกับ Network

โดยในการติดต่อกับระบบ Network นั้นเราเลือกใช้การต่อติดผ่าน Sockets ในการส่งภาพผ่านนั้นจะเริ่มจากการ ตรวจสอบการเชื่อมต่อกับ Embedded Board ว่ามีการเชื่อมต่อกับอุปกรณ์ที่จำเป็นในการทำงานแล้วหรือยัง ถ้าไม่ให้มีการ แจ้ง Error ว่ายังไม่มีการเชื่อมต่ออุปกรณ์ ถ้ามีการเชื่อมต่อแล้วจะทำการตั้งค่าเริ่มต้นให้กับกล้องและ Network Interface จากนั้นโปรแกรมจะทำการ เปิด TCP/IP เมื่อเปิดให้บริการแล้วจะเข้าสู่โหมดการทำงานโดยจะทำการรับภาพจากกล้องแล้วทำการส่งผ่าน Protocol TCP/IP เพื่อนำไปแสดง จากนั้นก็ทำการตรวจสอบการทำงานว่ามีคำสั่งให้หยุดการทำงานแล้วหรือยัง ถ้ามีให้ทำการหยุด

โดยทั่วไปในการติดต่อสื่อสารที่เป็นที่นิยมใช้กันในปัจจุบันจะกระทำบนพื้นฐานของ Protocol TCP/IP และชุดของ Software ซึ่งทำงานกับ Packets ตามมาตรฐานของ TCP/IP (ซึ่งต่อไปจะเรียกว่า TCP/IP Protocol Software) นั้นจะถูกสร้างขึ้นมาเป็นส่วนหนึ่งของตัวระบบปฏิบัติการ (Operating System, OS) ดังนั้น เมื่อ Application Software ใด ๆ ที่ต้องการสื่อสารผ่านระบบเครือข่ายตามมาตรฐาน TCP/IP Protocol จะต้องติดต่อกับ ระบบปฏิบัติการเพื่อขอรับบริการ (Services) สำหรับทางผู้พัฒนามาตรฐาน TCP/IP Protocol ไม่ต้องการให้ TCP/IP Protocol ใช้งานได้กับ Platform ใดเป็นการเฉพาะหรือใช้งานได้กับเพียงระบบปฏิบัติการระบบใดระบบหนึ่ง ดังนั้นผู้พัฒนาจึงมีความระมัดระวัง ในการกำหนดมาตรฐานมิให้มีการอ้างถึงข้อมูลการเชื่อมต่อภายในซึ่งเป็นรูปแบบเฉพาะของ Platform ใด Platform หนึ่ง หรืออ้างถึงวิธีการเชื่อมต่อกับ Application Software ในลักษณะหรือรูปแบบเฉพาะของระบบปฏิบัติการจากผู้ผลิตรายใดรายหนึ่ง และด้วยแนวคิดในการออกแบบที่ไม่ยึดติดกับ Platform หรือระบบปฏิบัติการใด ๆ นี้ทำให้อาจกล่าวได้ว่า Protocol TCP/IP มีลักษณะเป็น Loosely Specified Protocol Software Interface ตาม concept ดังต่อไปนี้

- Allocate local resources for communication
- Specify local and remote communication end-points
- Initiate a connection (client side)
- Wait for an incoming connection (server side)
- Send or receive data
- Determine when data arrives
- Generate urgent data
- Handle urgent data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Terminate a connection gracefully
- Handle connection terminate from the remote site
- Abort communication
- Handle error condition or a connection abort
- Release local resources when communication finishes

นอกจาก Concept ในการทำงานดังที่ได้กล่าวมาแล้วข้างต้น ทางผู้พัฒนามาตรฐาน TCP/IP Protocol ได้สร้าง Conceptual Interface ขึ้นมาเพื่อเป็นแนวทางให้นักออกแบบระบบสามารถพัฒนาซอฟแวร์ของตนได้ง่ายและเร็วขึ้น โดย Conceptual Interface ที่ทางผู้พัฒนามาตรฐาน TCP/IP Protocol ได้สร้างขึ้นนั้นจะอยู่ในรูปของ Set of Procedures and Functions เหตุที่ Conceptual Interface อยู่ในรูปแบบของ Procedures และ Functions นั้นเนื่องมาจากในช่วงเวลาที่พัฒนา Conceptual Interface ระบบปฏิบัติการต่าง ๆ ในยุคนั้นนิยมใช้วิธีการซึ่งรียกกันว่า Procedural Mechanism ในการเคลื่อนย้ายการควบคุม (Transfer control) จาก Application Software ไปยังระบบปฏิบัติการ ดังนั้นจึงอาจกล่าวโดยสรุปเกี่ยวกับ Conceptual Interface ได้ดังนี้ *"Conceptual* Interface ซึ่งถูกกำหนดโดยผู้พัฒนามาตรฐาน TCP/IP นั้นจะไม่ระบุรูปแบบของการแทนข้อมูล (Data representation) หรือรายละเอียดในการเขียนโปรแกรม หากแต่แสดงเพียงตัวอย่างหนึ่งของของการเชื่อมต่อ ซึ่งมีความเป็นไปได้

Sockets ที่ถูกสร้างขึ้นนั้นจะมีวัตถุประสงค์ในการใช้งานอยู่เพียง 2 ลักษณะได้แก่ การรอรับการเชื่อมต่อและการเชื่อมต่อไปยังเครื่องปลายทาง (Initiate a connection) โดย Sockets ที่รอรับการเชื่อมต่อนั้นจะเป็น Sockets ที่ใช้ใน Server Application ซึ่ง Sockets ชนิดนี้มีชื่อเรียกว่า Passive Socket ในขณะที่ Sockets ที่พยายาม เชื่อมต่อไปยังเครื่องปลายทางซึ่งใช้ใน Client Application นั้นจะมีชื่อเรียกว่า Active Socketทั้ง Active Sockets และ Passive Sockets นั้นในชั้นแรกจะถูกสร้างขึ้นมาด้วยวิธีการเดียวกัน ดังนั้นข้อ แตกต่างของ Active Sockets และ Passive Sockets จึงอยู่ที่วัตถุประสงค์การใช้งานเท่านั้น

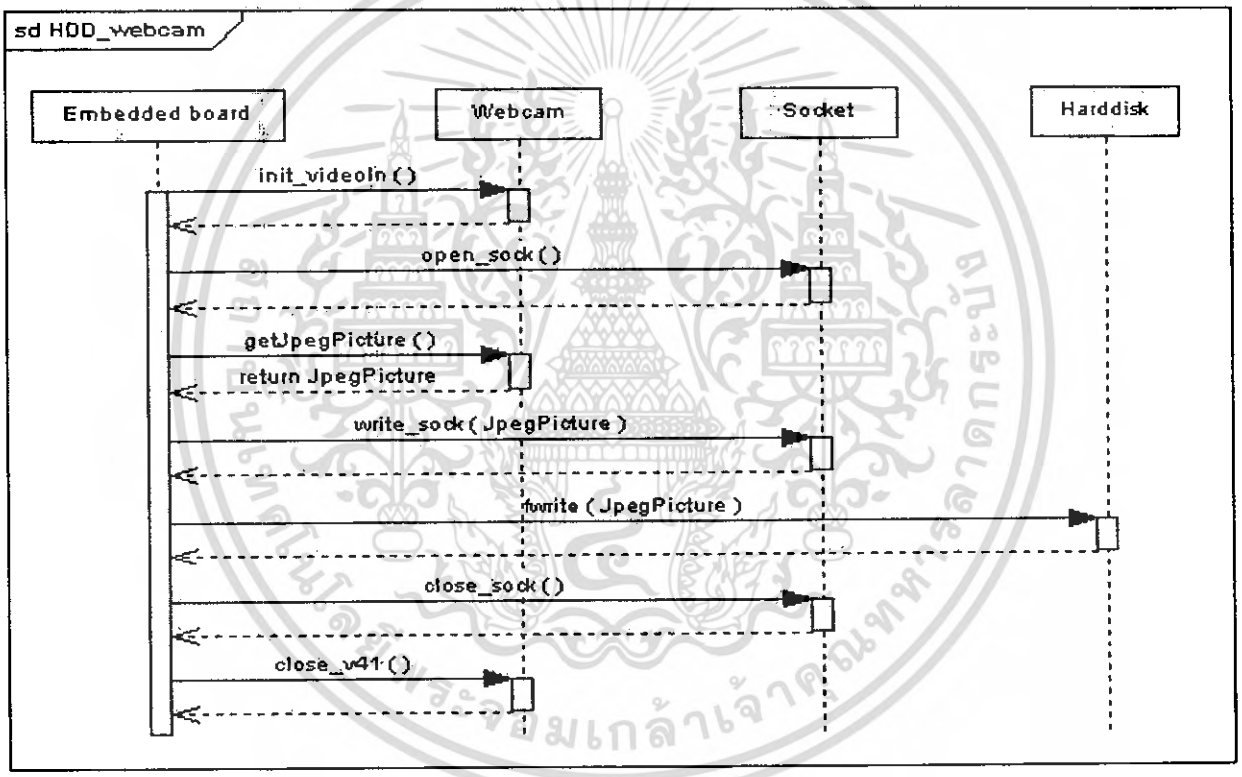
3.2.2 แนวคิดการออกแบบการแปลงรูปแบบไฟล์และการบันทึก

โปรแกรมจะเริ่มจากการ ตรวจสอบการเชื่อมต่อกับ Embedded Board ว่ามีการเชื่อมต่อกับอุปกรณ์ที่จำเป็นในการทำงานแล้วหรือยัง ถ้าไม่ให้มีการ แจ้ง Error ว่ายังไม่มีการเชื่อมต่ออุปกรณ์ ถ้ามีการเชื่อมต่อแล้วจะทำการตั้งค่าเริ่มต้นให้กับกล้องและ Harddisk จากนั้นโปรแกรมจะทำการเปิดให้บริการ Harddisk เมื่อเปิดให้บริการแล้วจะเข้าสู่โหมดการทำงานโดยจะทำการรับภาพจากกล้องแล้วทำการเปลี่ยนชื่อไฟล์ให้เป็นลำดับเพื่อใช้เป็น Index เพื่อนำใช้ทำภาพเครื่องไหว แล้วทำการนำไปเก็บบันทึกลง Harddisk จากนั้นก็ทำการตรวจสอบการทำงานว่ามีคำสั่งให้หยุดการทำงานแล้วหรือยัง ถ้ามีให้ทำการหยุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

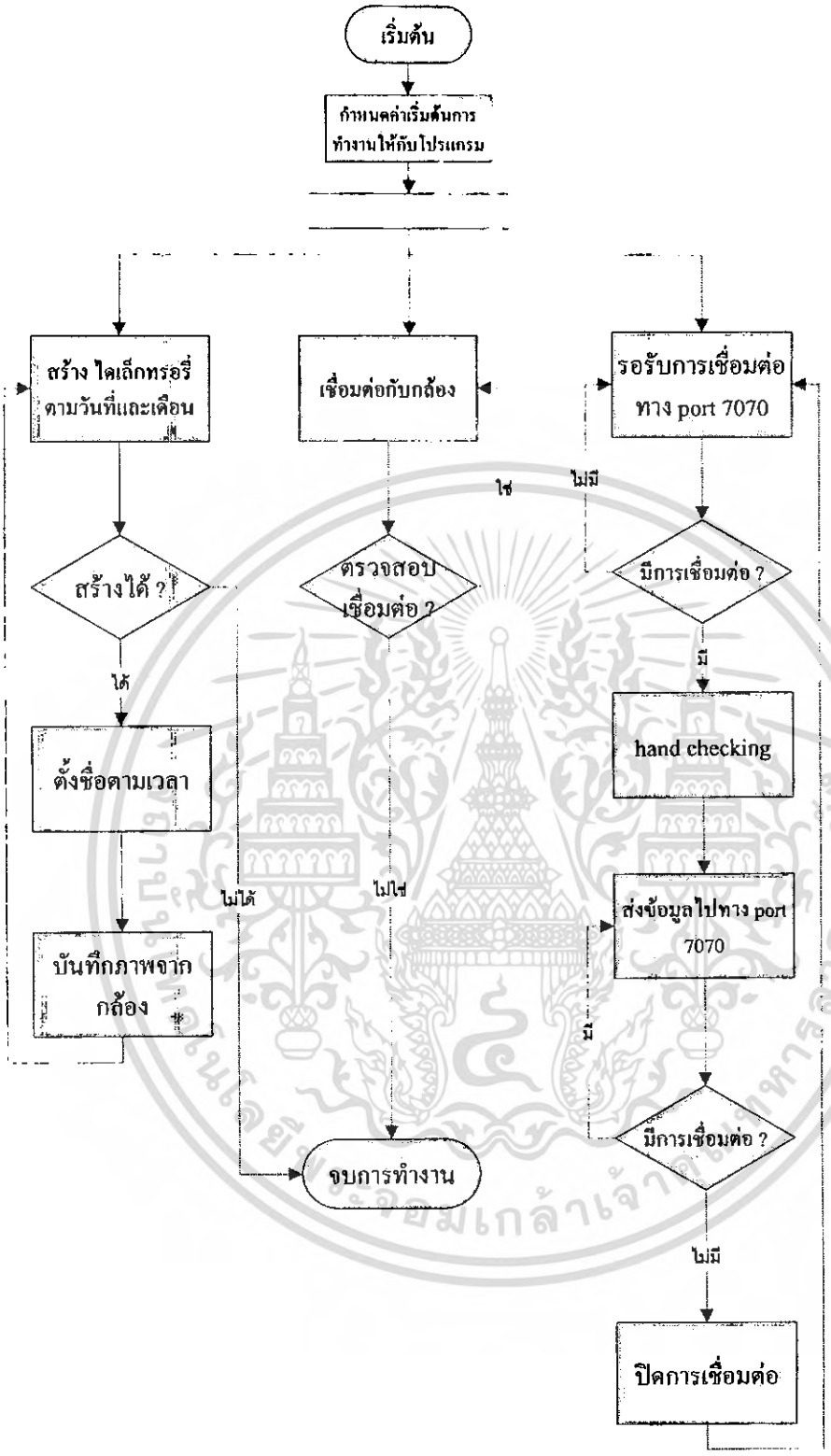
3.2.3 แนวคิดการออกแบบโปรแกรมการทำงานรวมทั้งระบบ

เป็นการนำแนวคิดการออกแบบโปรแกรมที่นำเอาแนวคิดทั้งสองลักษณะข้างต้นมารวมเข้าด้วยกัน จะเริ่มจากการ ตรวจสอบการเชื่อมต่อกับ Embedded Board ว่ามีการเชื่อมต่อกับอุปกรณ์ที่จำเป็นในการทำงานแล้วหรือยัง ถ้าไม่ให้มีการ แจ้ง Error ว่ายังไม่มีการเชื่อมต่ออุปกรณ์ ถ้ามีการเชื่อมต่อแล้วจะทำการตั้งค่าเริ่มต้นให้กับอุปกรณ์โดยรวม จากนั้นโปรแกรมจะทำการ เปิดให้บริการการทำงานต่างๆ แล้วจะเข้าสู่โหมดการทำงานโดยจะทำการรับภาพจากกล้องแล้วนำภาพที่รับมาไปแสดงผ่านทาง Protocol TCP/IP แล้วจึงทำการเปลี่ยนชื่อไฟล์ให้เป็นลำดับเพื่อใช้เป็น Index เพื่อนำใช้ทำภาพเครื่องไหว แล้วนำไปเก็บบันทึกลง Harddisk จากนั้นก็ทำการตรวจสอบการทำงานว่ามีคำสั่งให้หยุดการทำงานแล้วหรือยัง ถ้ามีให้ทำการหยุด



รูปที่ 3.8 แสดง Sequence Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



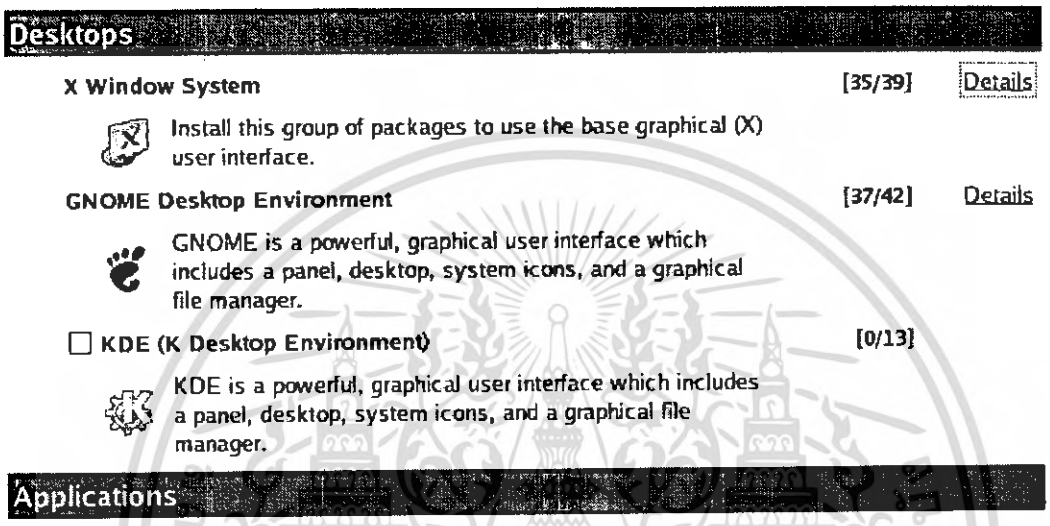
รูปที่ 3.9 แสดง Flowchart ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ขั้นตอนการเตรียม Operating System เพื่อใช้ในการพัฒนา Program

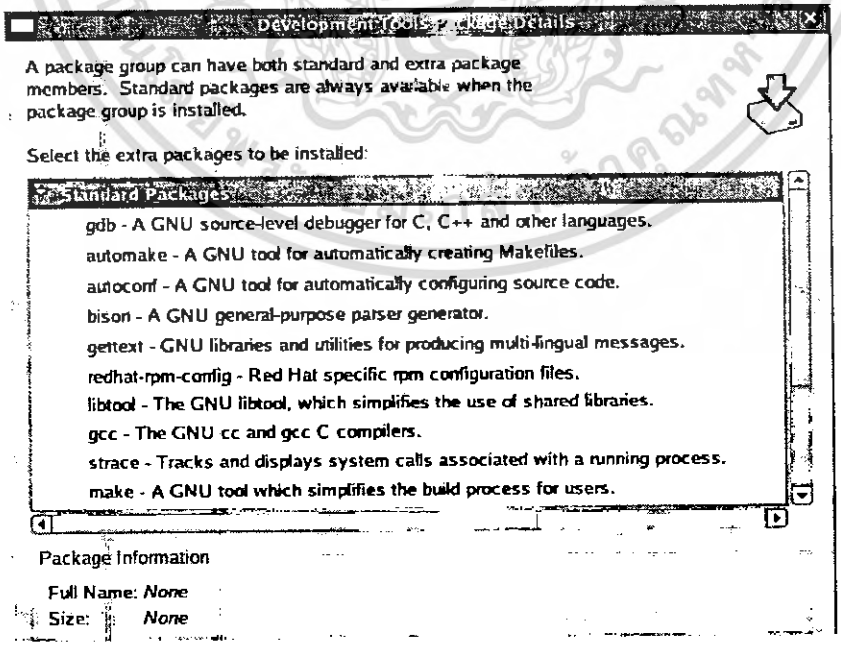
3.3.1. ในขั้นแรกต้องทำการลง Linux ซึ่งในที่นี้ใช้ Fedora Core 4 ในการทดลอง โดยPackage ที่จำเป็นต้องลง คือ

เพื่อให้สามารถใช้งานกล้อง Video ได้ในการการพัฒนาโปรแกรม ต้องทำการลง ในลักษณะที่เป็น graphical mode โดยจะต้องทำการเลือก package X Window และ เลือก Desktop Environment ตัวใดตัวหนึ่งก็ได้



รูปที่ 3.10 Menu ที่ต้องเลือกของ Linux Fedora core 4

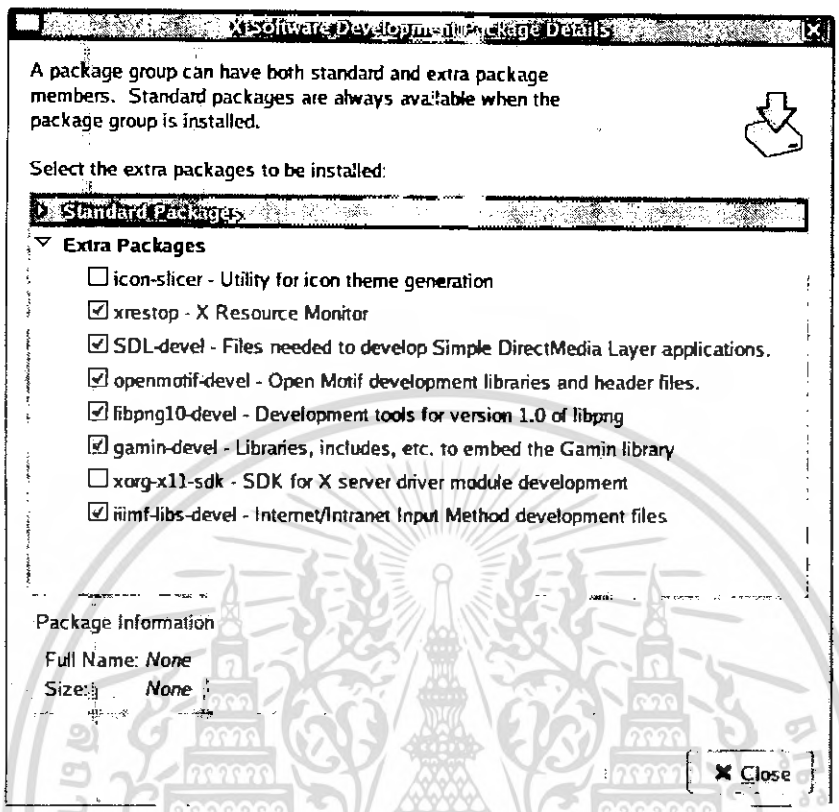
เพื่อที่จะสามารถพัฒนาโปรแกรมบน Linux ได้ต้องทำการลง Package gcc ซึ่งจะเป็นตัว Compilers ที่ใช้ใน Compile โปรแกรม ที่เป็น C หรือ C++ ได้



รูปที่ 3.11 Compile ที่ต้องแสดงในการลง Linux Fedora core 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Package ที่จำเป็นต้องติดตั้งอย่างสุดท้ายคือ SDL-Devel เพื่อรองรับการใช้งาน กล้อง video



รูปที่ 3.12 Module SDL ที่ต้องทำการเลือกลงบน Linux Fedora core 4

3.3.2. ขั้นตอนการลง Driver ของกล้อง webcam ZC301-2 บน Fedora Core 4 เพื่อให้ Operating System นี้สามารถติดต่อกับกล้อง webcam Creative Live! ตามนี้

โดยทำการสร้าง folder: /root/Project/webcam ขึ้นมาแล้ว Copy spca5xx-20060501.tar.gz เข้าไปไว้ใน /root/Project/webcam แล้วทำตามนี้

```
[root@hardware44 webcam]# tar xfz ../src/spca5xx-20060501.tar.gz -C .
[root@hardware44 webcam]# cd spca5xx-20060501/
[root@hardware44 spca5xx-20060501]# make
[root@hardware44 spca5xx-20060501]# make install
```

เหมือทำการติดตั้งแล้วให้ทดสอบการใช้คำสั่ง lsmod จะเห็นว่า มี Module spca5xx ทำงานอยู่

```
[root@hardware44 spca5xx-20060501]# lsmod
Module                  Size  Used by
spca5xx               328304  0
videodev                9537  1 spca5xx
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3. ขั้นตอนการลง Crosstool ซึ่งเป็น Tool ที่ใช้ในการ ทำ Cross Compile ซึ่งเป็นการ Compile Source Code ให้สามารถนำไปใช้ run บน Embedded Board ได้ ซึ่งในที่นี้เราจะใช้งาน uClibc เป็นตัว Cross Compiler โดยขั้นตอนการติดตั้งจะใช้ buildroot เป็นเครื่องมือที่ช่วยในการติดตั้ง ตัว Cross Compiler และ ยังช่วยในส่วนของการสร้าง root filesystem ใน หัวข้อ 4.2

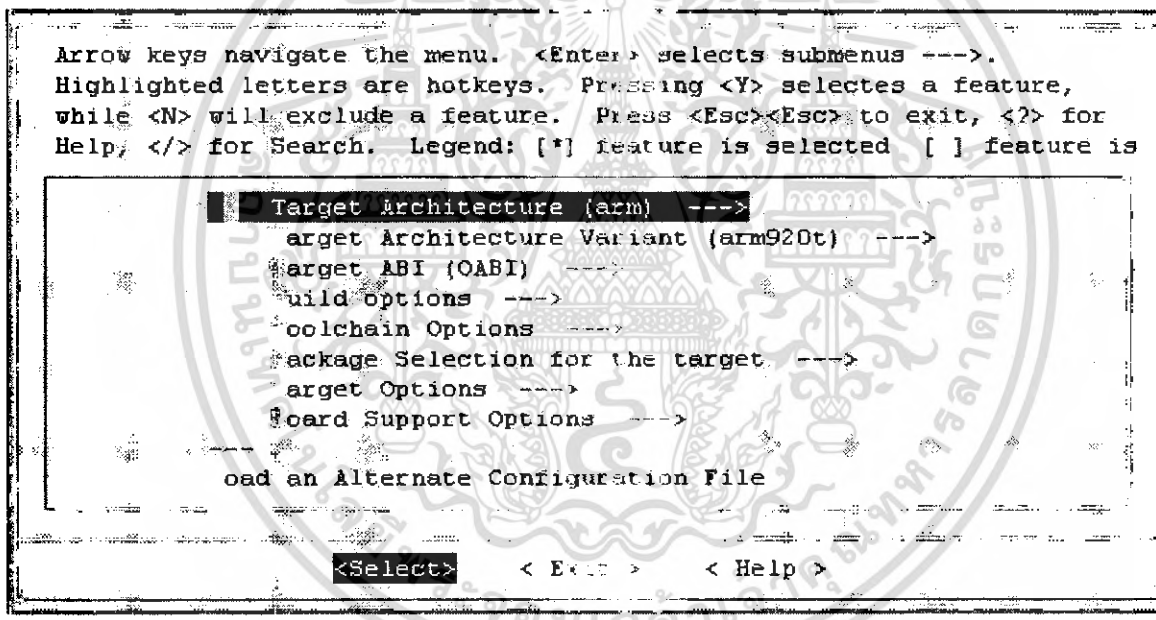
```
[root@hardware44 work]# tar xjf ../src/buildroot-20061229.tar.bz2 -C .
```

ใช้คำสั่ง make menuconfig เพื่อเข้ามา หน้า Configuration โดยเลือก

Target Architecture เป็น arm

Target Architecture เป็น arm920t

Target ABI เป็น OABI

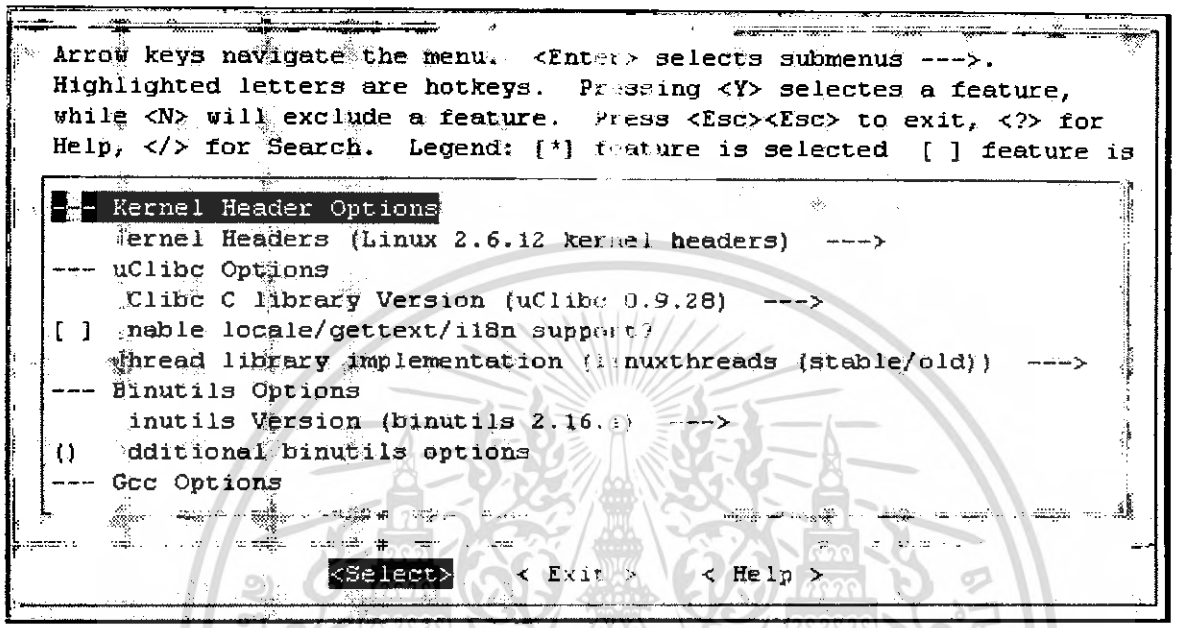


รูปที่ 3.13 Menu การ Configuration Buildroot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือก

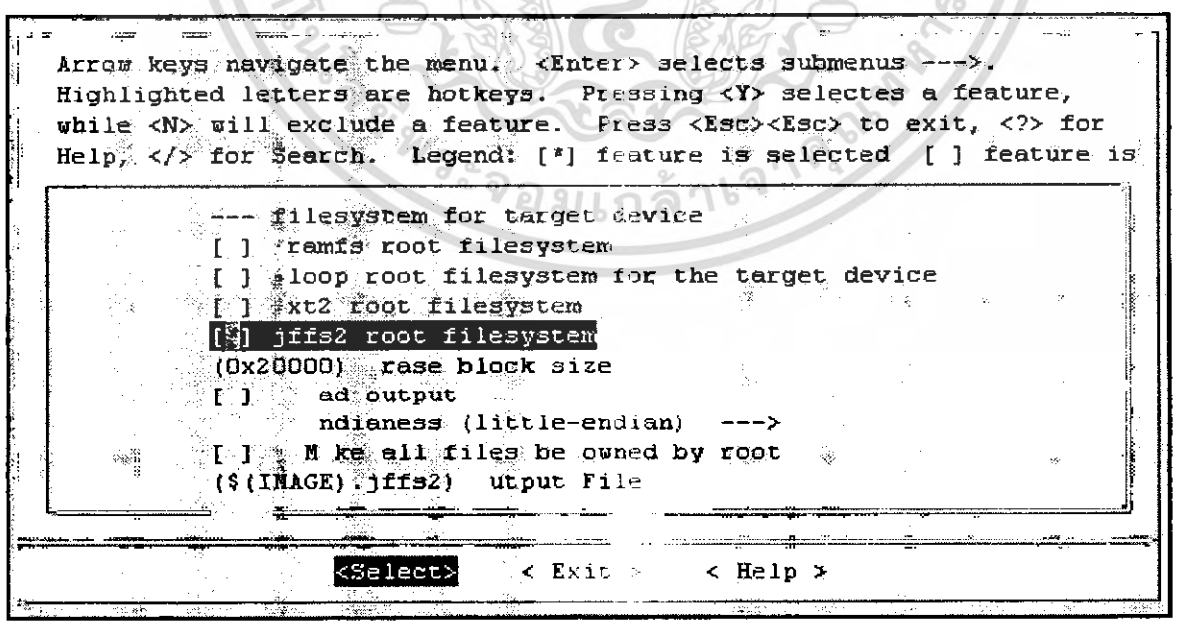
uClibc c library Version	เป็น uClibc 0.9.28
Binutils Version	เป็น Binutils 2.16.1
Gcc	เป็น gcc 3.4.4



รูปที่ 3.14 Menu การ Configuration Toolchain

ในหัวข้อ Target Options เลือก

filesystem for target device เป็น jffss2 root filesystem



รูปที่ 3.15 Menu ที่ใช้ในการ สร้าง root filesystem

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นให้ทำการ make และ make install ตามลำดับ และเมื่อ compile เรียบร้อยแล้วให้เข้าไปเพิ่ม /root/Project/work/buildroot/build_arm/staging_dir/bin ในไฟล ~/.bash_profile

สามารถทดสอบโดยใช้คำสั่ง arm-linux-gcc-3.4.4 –v จะแสดงผลออกมาดังนี้

```
Reading specs from /root/Project/work/buildroot/build_arm/staging_dir/lib/gcc/ar
m-linux-uclibc/3.4.4/specs
Configured with: /root/Project/work/buildroot/toolchain_build_arm/gcc-3.4.4/conf
igure --prefix=/root/Project/work/buildroot/build_arm/staging_dir --build=i386-p
c-linux-gnu --host=i386-pc-linux-gnu --target=arm-linux-uclibc --enable-language
s=c --disable-__cxa_atexit --enable-target-optspace --with-gnu-ld --enable-share
d --disable-nls --enable-threads --enable-multilib
Thread model: posix
gcc version 3.4.4
```

เมื่อทำจนครบทุกขั้นตอนในหัวข้อ 4.1 แล้วจะสามารถ พัฒนาโปรแกรม เพื่อ Cross Compiler และนำไปใช้งานบนบอร์ดได้

3.5 ขั้นตอนการเตรียม การใช้งาน Embedded Board ที่ใช้ CPU ประเภท ARM AT91RM9200

3.4.1. ขั้นตอนการเตรียม Linux Kernel ซึ่งเป็น Kernel ที่เราจะใช้งานบน Embedded Board โดยจะต้องเตรียมไฟล์ที่ใช้ในการสร้างมีดังนี้

linux-2.6.12.tar.bz2	จาก	www.kernel.org
2.6.12-at91.patch.gz	จาก	maxim.org.za
usb-2.6.12LE06.patch.tar.gz	จาก	mxhaard.free.fr/spca50x/embedded/ArmEP9302/

linux-2.6.12.tar.bz2 เป็น Linux Kernel ที่จะใช้งานบนบอร์ด Embedded Board

2.6.12-at91.patch.gz เป็น patch ของ Linux Kernel ที่จำเป็นต้องเพิ่มให้กับ Linux Kernel เพื่อให้สามารถใช้สร้าง Core Linux ให้กับ CPU ARM AT91RM9200 ได้

usb-2.6.12LE06.patch.tar.gz เป็น patch ในส่วน USB ของ Linux Kernel ที่จำเป็นต้องเพิ่มให้กับ Linux Kernel เพื่อให้ CPU ARM สามารถใช้ USB ในการติดต่อกับกล้อง webcam ZC301-2

ขั้นที่ 1 ทำการสร้างไดเล็กทรอรี่ /root/Project/kernel ขึ้นมาแล้ว ทำการแตกไฟล์ linux-2.6.12.tar.bz2 ไปไว้ในไดเล็กทรอรี่ ที่สร้างขึ้นตามคำสั่งนี้

```
[root@hardware44 kernel]# tar xjf ../src/linux-2.6.12.tar.bz2 -C .
[root@hardware44 kernel]# ln -s linux-2.6.12/ linux
[root@hardware44 kernel]# ls
linux  linux-2.6.12
[root@hardware44 kernel]#
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

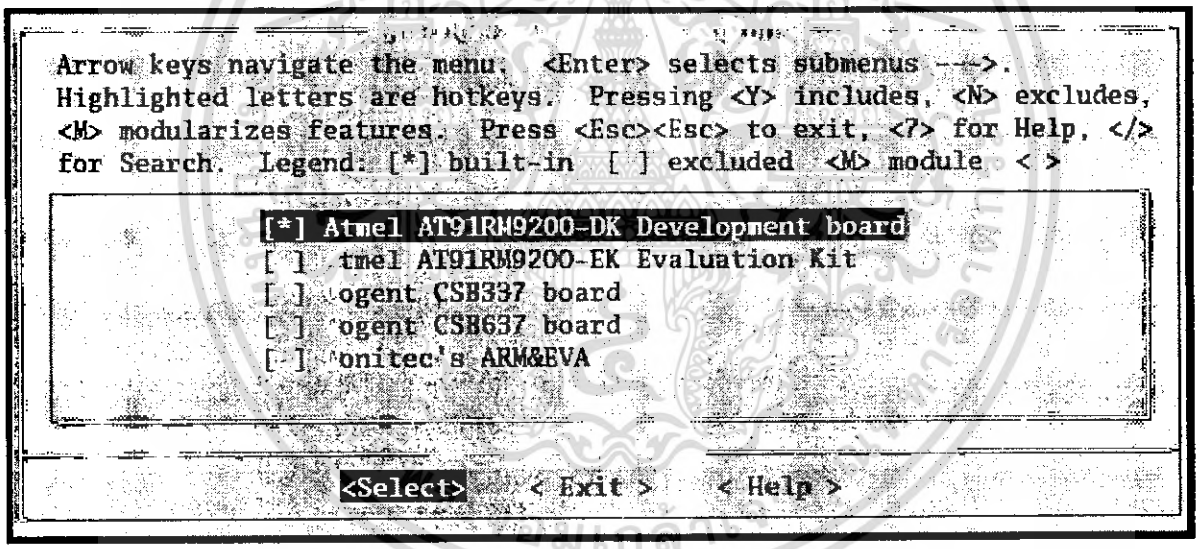
ขั้นที่ 2 ทำการ Patch 2.6.12-at91.patch และ Patch usb-2.6.12LE06.patch.tar.gz ลงไปใน Kernel ตามนี้

```
[root@hardware44 kernel]# zcat ../src/2.6.12-at91.patch.gz |patch -d ./linux -p1
[root@hardware44 kernel]# cd linux
[root@hardware44 linux]# cp arch/arm/configs/at91rm9200dk_defconfig .config
[root@hardware44 linux]# cd drivers/
[root@hardware44 drivers]# tar xfz ../../../src/usb-2.6.12LE06.patch.tar.gz -C .
[root@hardware44 drivers]# patch -p1 < usb-2.6.12.patch -d ./usb
[root@hardware44 linux]# vim Makefile
```

จากนั้นเข้าไปแก้ในไฟล์ Makefile ตามนี้

```
ARCH            ?= $(SUBARCH)       ARCH            ?= arm
CROSS_COMPILE   ?=                  CROSS_COMPILE   ?= arm-linux-
```

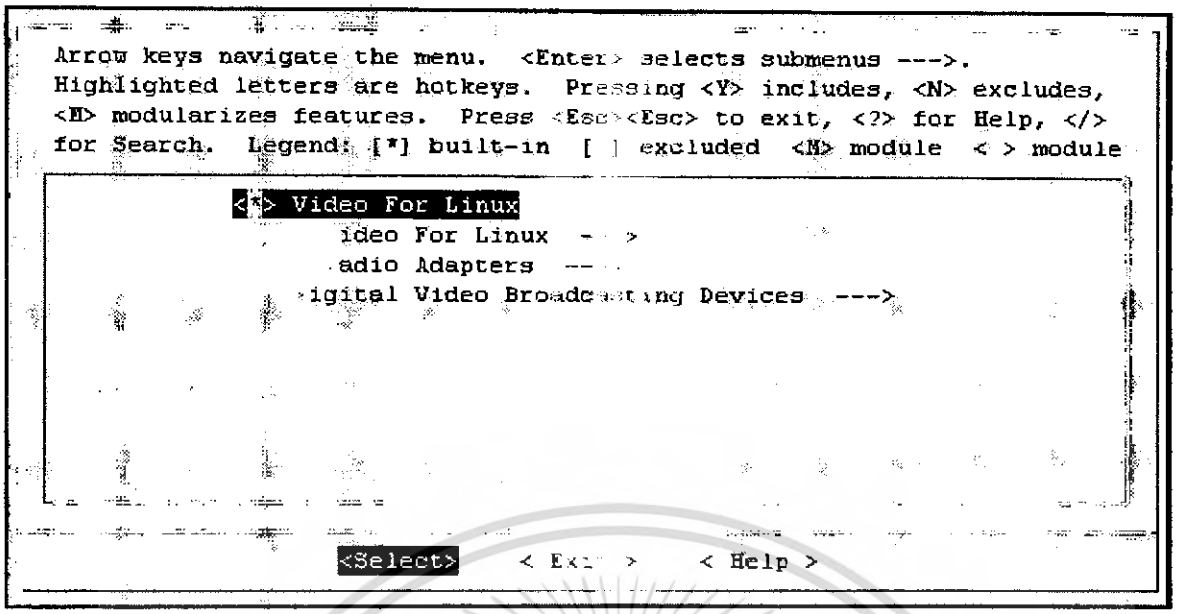
ขั้นที่ 3 เมื่อทำการลง Patch และ แก้ Makefile เรียบร้อยแล้ว ให้ ใช้คำสั่ง make menuconfig เพื่อทำการปรับแต่ง Environment บนบอร์ด โดยสิ่งที่จำเป็นต้องเลือกมีดังนี้



รูปที่ 3.16 การปรับแต่งให้ใช้งานบน Embedded Board

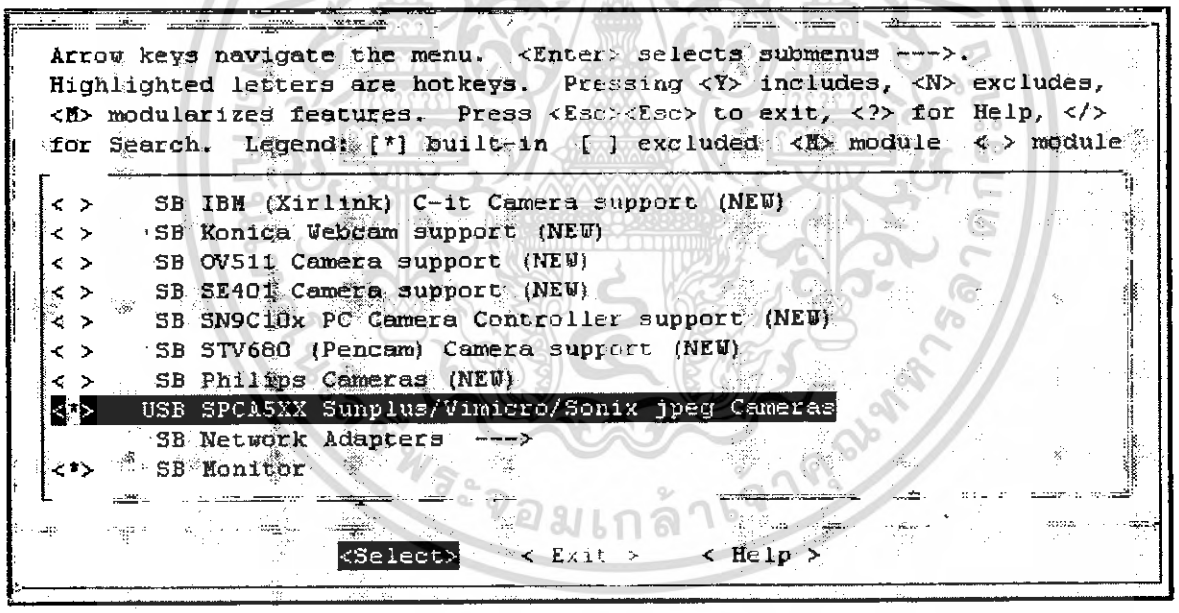
เลือก Development board เป็น Atmel AT91RM9200-DK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 การปรับแต่งให้ใช้งานกล้อง Video บน Embedded Board

เลือก Video For Linux เพื่อให้บอร์ดรองรับการใช้งาน Video



รูปที่ 3.18 เลือก USB SPCA5XX ซึ่งเป็นDriver ของกล้องที่นำมาใช้งาน

ขั้นตอนสุดท้ายให้เราทำการใช้คำสั่ง make zImage เมื่อ Compile เรียบร้อยแล้วจะได้ไฟล์ zImage ใน /root/Project/kernel/linux/arch/arm/boot ที่ต้องใช้ลงใน Embedded Board ต่อไป

3.4.2. ขั้นตอนเตรียม root filesystem ซึ่งจะใช้เป็น root filesystem บน Embedded Board

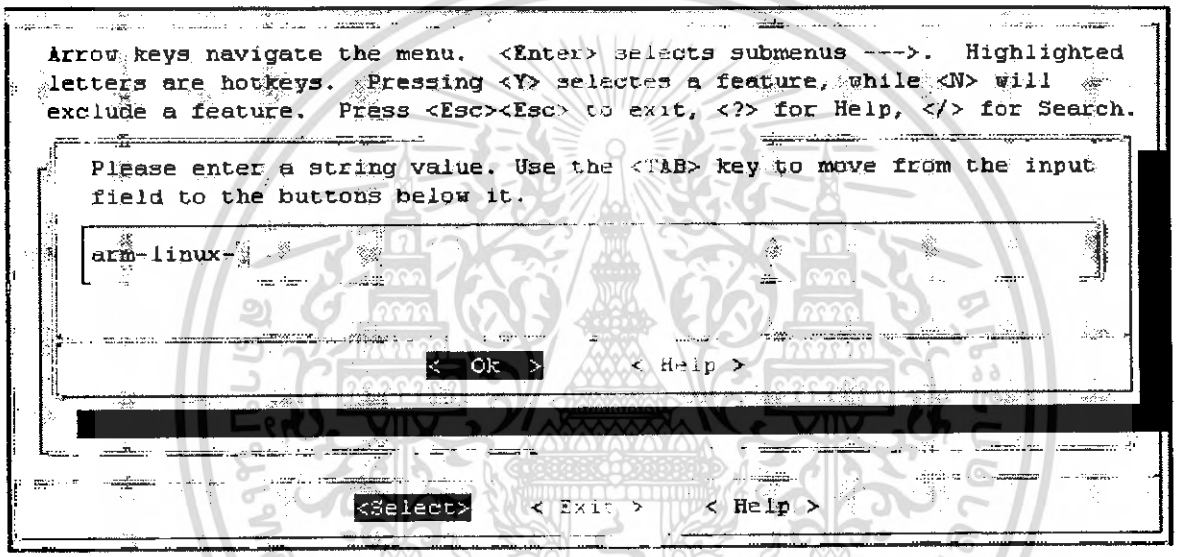
ขั้นตอนแรกเป็นการลง buildroot ซึ่งเป็น tool ซึ่งที่ใช้ช่วยสร้าง root filesystem แต่ในขั้นตอนนี้จะอธิบายในหัวข้อ 4.1.3. ไปแล้ว ซึ่งผลหลังจากการ compile buildroot จะทำให้ได้ root filesystem

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน buildroot/build_arm/root แต่มีขนาดใหญ่เกินที่จะนำลง Embedded Board จึงต้องทำการลดขนาด โดยการ สร้าง root filesystem ขึ้นมาอีกโดยใช้โปรแกรม busybox ซึ่งจะต้องทำการ ก็อปไฟล์ .config ของ busybox ใน buildroot/build_arm/busybox-1.2.2.1 ไปใช้งานใน busybox ที่ทำการสร้างใหม่

- โดยทำการสร้าง Folder: /root/Project/rootfs ขึ้นมาแล้ว Copy busybox-1.2.1.tar.bz2 เข้าไปไว้ใน /root/Project/rootfs แล้ว ทำตามคำสั่งนี้

```
[root@hardware44 work]# tar jxf ../src/busybox-1.2.2.1.tar.bz2 -C .
[root@hardware44 work]# cd busybox-1.2.2.1/
[root@hardware44 busybox-1.2.2.1]# cp ../buildroot/build_arm/busybox-1.2.2.1/.config .
[root@hardware44 busybox-1.2.2.1]# make menuconfig
```



รูปที่ 3.19 สิ่งที่ต้องเพิ่มในการ Configuration

นอกจากนั้นจำเป็นต้องเปลี่ยน Cross Compiler prefix ให้เป็น arm-linux-

เมื่อทำการเลือก package ที่จะใช้บน Embedded Board เรียบร้อยแล้ว ให้ทำตามขั้นตอนต่อไป

```
[root@hardware44 work]# make
[root@hardware44 work]# make install
```

ขั้นตอนต่อไปให้ทำการ copy ไฟล์ที่สร้างใน _install มาไว้ใน rootfs ที่ได้สร้างเอาไว้ แล้วทำการสร้าง ไดเล็กทรอรี่ ที่เหลือ ให้ครบ

```
[root@hardware44 work]# mkdir rootfs
[root@hardware44 work]# cp busybox-1.2.2.1/_install/* rootfs/ -R
[root@hardware44 work]# mkdir dev etc lib home mnt proc root tmp var
```

เมื่อได้ไดเล็กทรอรี่ ที่ต้องการครบแล้วให้ใช้คำสั่ง

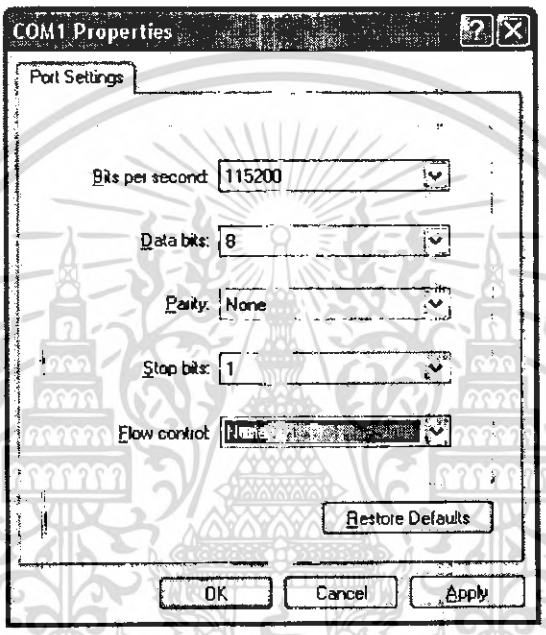
```
[root@hardware44 work]# mkfs.jffs2 -e 8448 -r rootfs -o rootfs.jffs2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อสร้างไฟล์ rootfs.jffs2 เพื่อใช้บน Embedded Board ต่อไป

4.1.2. ขั้นตอนการติดตั้ง root filesystem และ kernel บน Embedded Board โดยเมื่อเรามี ไฟล์ zImage และ rootfs .jffs2 ให้นำไฟล์ทั้งสองไปไว้ใน /tftpboot/ และใช้โปรแกรม tftpserver ทำการสร้าง server เพื่อใช้ในการส่งไฟล์ให้กับ Embedded Board

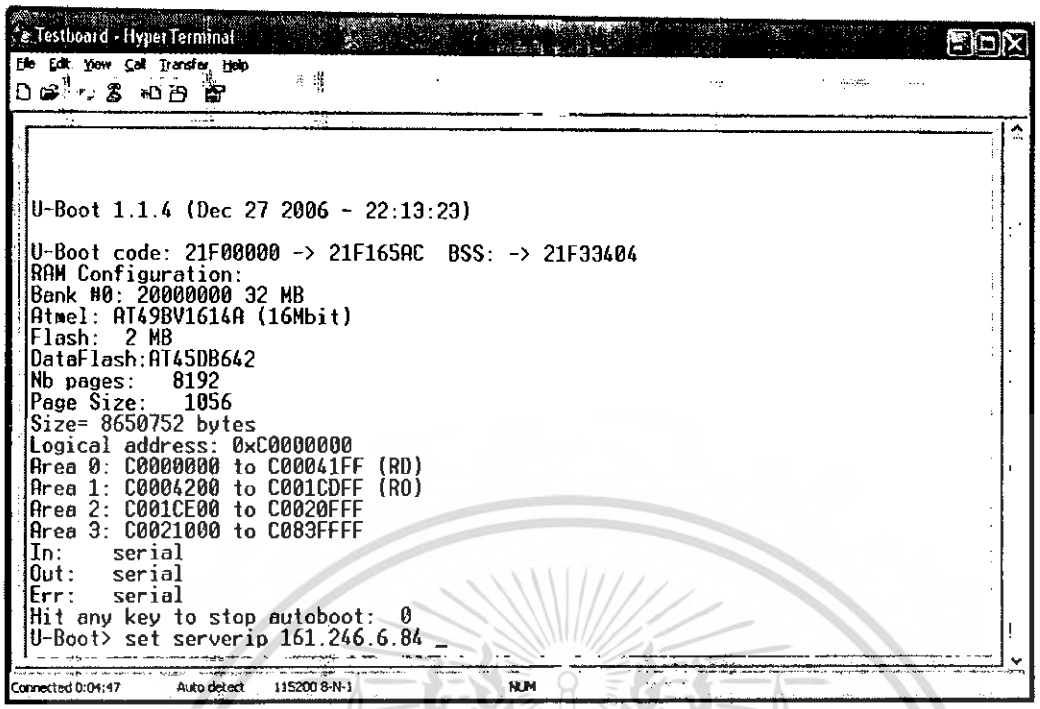
โดยเราจะใช้โปรแกรม HyperTerminal โดยเลือกตาม นี้



รูปที่ 3.20 การใช้งาน HyperTerminal เข้าไปใช้งาน Embedded Board

เริ่มแรก เมื่อ เปิดบอร์ดขึ้นมา จะทำการ boot U-boot ให้ใช้คำสั่ง set serverip เพื่อเซ็ต ip ของ server tftp ให้ต่อกับเครื่องที่เราเปิดเป็น tftpserver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 การใช้งาน Embedded Board

หลังจากนั้นให้ทำตามคำสั่งนี้เพื่อทำการ เซต address ในตำแหน่งดังกล่าวให้ เป็น บิท 1 แล้วนำไฟล์ ZImage มาวาง

```
U-Boot> mw.b 20008000 ff 14A000
U-Boot> tftp 20008000 zImage-at91-mtd02
U-Boot> cp.b 20008000 C0021000 14A000
```

หลังจากนั้นให้ทำตามคำสั่งนี้เพื่อทำการ เซต address ในตำแหน่งดังกล่าวให้ เป็น บิท 1 แล้วนำไฟล์ rootfs.jffs มาวาง

```
U-Boot> mw.b 20008000 ff 6D5000
U-Boot> tftp 20008000 rootfs.jffs2
U-Boot> cp.b 20008000 c016b000 6D5000
```

หลังจากนั้นให้ใช้คำสั่ง

```
U-Boot> set bootcmd 'cp.b C0021000 20008000 14A000;go 20008000;'
U-Boot> save
U-Boot> reset
```

เมื่อติดตั้งเรียนร้อยแล้ว จะสามารถ ใช้งาน Linux ซึ่งอยู่บน Embedded Board ได้

```
# ls
linuxrc
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำ Harddisk และกล้องมาเชื่อมต่อกับ USB แล้ว เพื่อให้สามารถใช้งานกล้องและ Harddisk ได้ ต้องทำตามขั้นตอนดังนี้ จะเป็นการตรวจสอบการเชื่อมต่อจากนั้นตามด้วนการสร้าง node เพื่อเชื่อมต่อ

```
# cat /proc/partitions
major minor  #blocks  name

  31     0        132 mtdblock0
  31     1       1320 mtdblock1
  31     2       6996 mtdblock2
   8     0   40146624 sda
   8     1          1 sda1
   8     5   40138371 sda5
# mknod /dev/sda b 8 0
# mknod /dev/sda5 b 8 5
# mknod /dev/video c 81 0
```

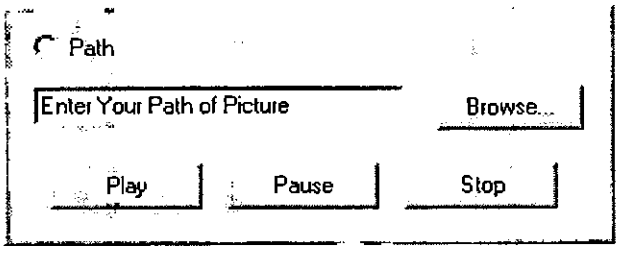
รูปที่ 3.22 สิ่งที่ต้องสร้างก่อนการใช้งาน กล้อง

ส่วน Execute File ที่เราใช้ run บน Embedded Board นั้น เราไม่ได้สร้างบน Board โดยตรง แต่เรา Compile Source Code โดยใช้ Crosstool ที่เราลงเพิ่ม คือ arm-linux-gcc แล้วก็จะได้ Execute File ที่สามารถ run บน Embedded Board นั้นๆ ได้

3.6 การพัฒนา Application ที่ใช้บน windows

ในการใช้งาน ยังใช้งานได้ยากเนื่องจาก Tool ที่ใช้บน Linux มีลักษณะเป็น Text mode ดังนั้นจึงได้มีการพัฒนา application ที่ใช้ในการเชื่อมต่อเข้าไปเพื่อดูภาพจากกล้อง ซึ่งพัฒนาด้วย ภาษา C# ในลักษณะที่เป็น GUI และใช้งานบน ระบบปฏิบัติการ Windows XP และ
มี Mode การทำงานอยู่สอง Mode การทำงานด้วยกันคือ

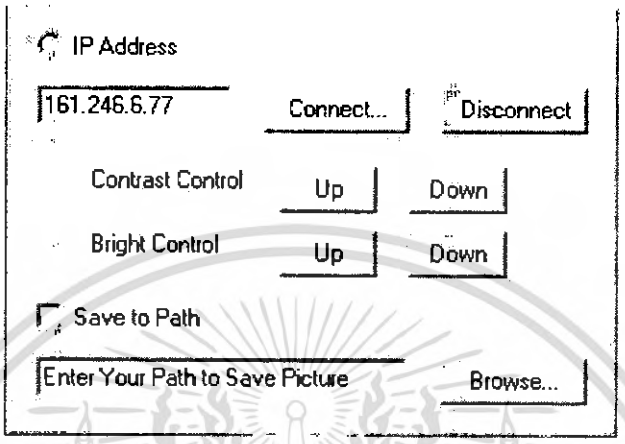
1. สามารถเปิดภาพที่ทำการบันทึกไว้จากกล้องได้เพื่อช่วยในการตรวจสอบภาพซึ่งจะมีปุ่มให้เลือก ที่เก็บภาพภายใน Harddisk โดยจะมีคำสั่งเพื่อเล่นภาพและหยุดภาพและเมนูบาร์ ที่ใช้เลือกดู ภาพในเวลาต่างๆ ได้ ดังรูป



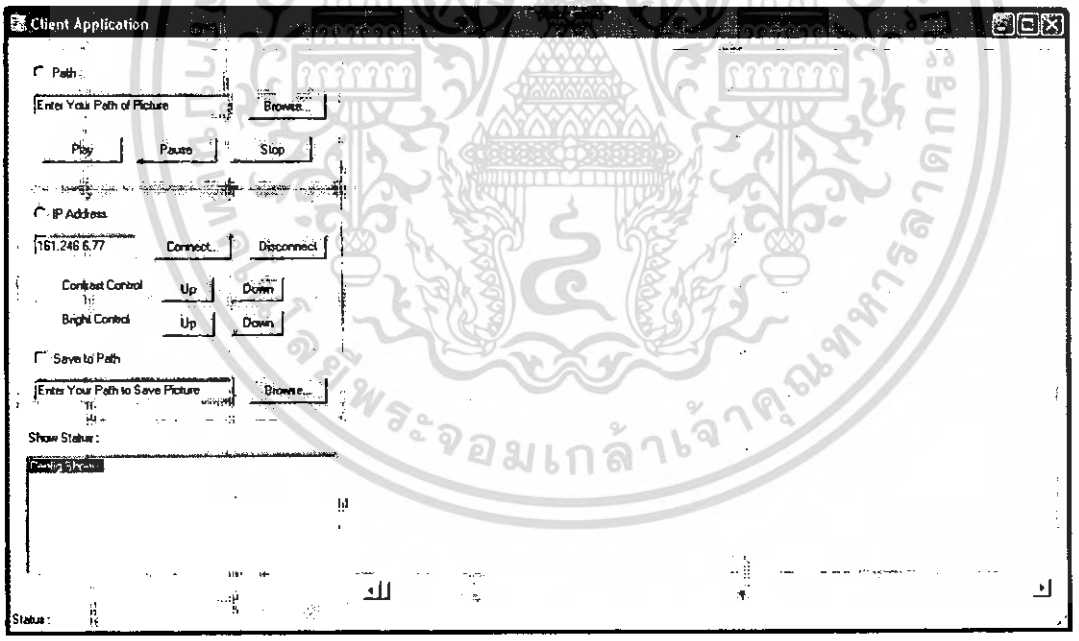
รูปที่ 3.23 การใช้งาน โปรแกรมใน Mode ที่1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เชื่อมตรงเข้าไปที่ บอร์ดได้โดยตรง โดยการพิมพ์ IP ของบอร์ดที่ใช้งาน จากนั้น โปรแกรมจะทำการแสดงภาพที่บอร์ดกำลังเก็บภาพอยู่แสดงออกมา และยังสามารถ เลือกที่จะบันทึกภาพที่ได้จากกล้องเก็บไว้ได้โดยตรง



รูปที่ 3.24 การใช้งาน โปรแกรมใน Mode ที่2



รูปที่ 3.25 ภาพรวมโปรแกรม

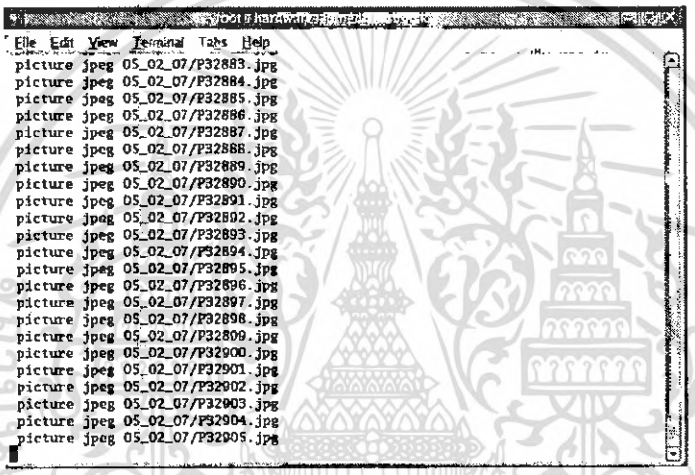
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

ในส่วนของการทดลองนั้นได้ แบ่งการทดลองออกเป็นสองส่วนใหญ่ๆ คือ การทดลองบน PC กับการทดลองบน ARM โดยจะอธิบายไว้ดังนี้

4.1 ผลการทดลองโปรแกรมบนบน PC ด้วย Fedora Core 4

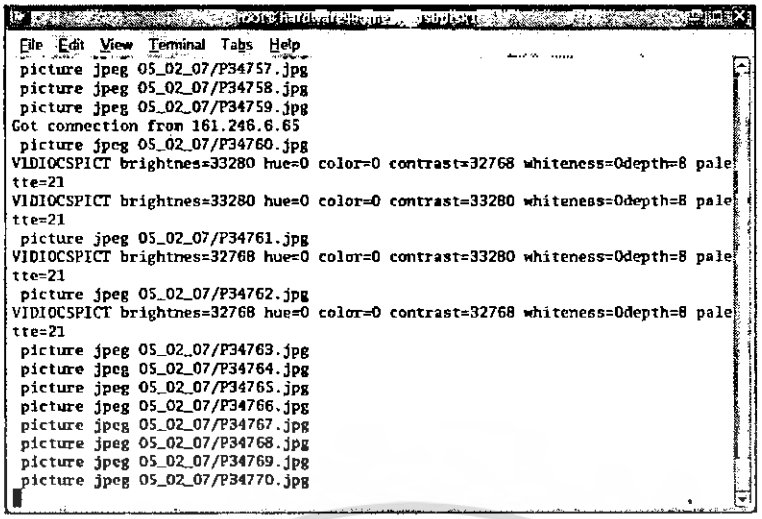
ในการทำงานบน PC ด้วยนั้น เราจะสามารถ Compile Source Code โดยใช้ Tool ที่มีใน Linux อยู่แล้วอย่าง gcc ก็จะได้ Execute File ที่เราจะใช้ run บน PC ได้เลย ดังที่ได้แสดงในรูปด้านล่าง



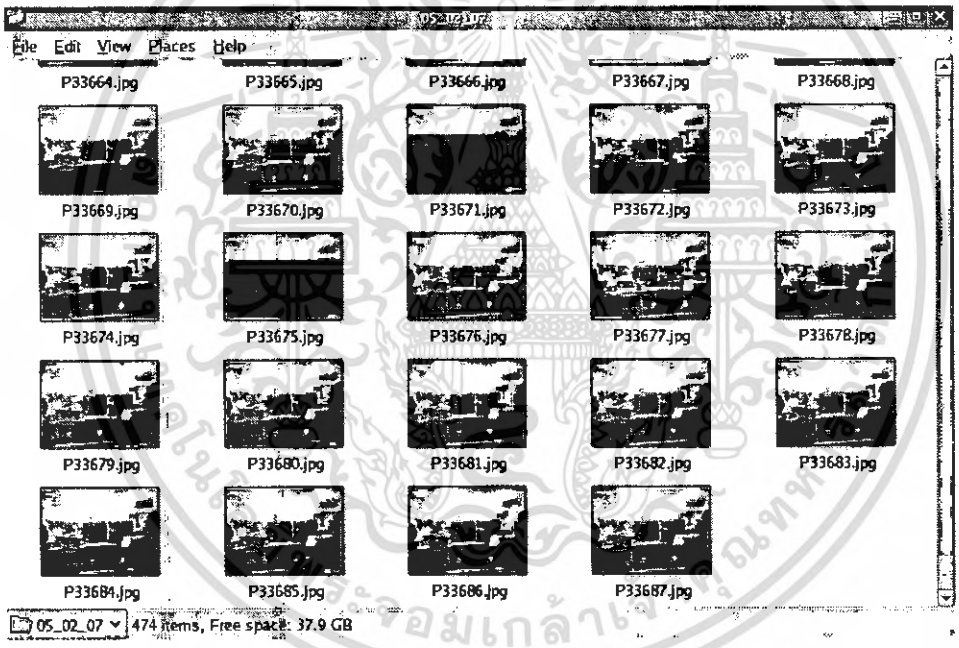
รูปที่ 4.1 แสดงผลการบันทึกภาพทางฝั่ง Server

รูปที่ 4.2 แสดงผลการแสดงภาพทางฝั่ง Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงให้ทราบว่ามีการเชื่อมเข้ามาที่ Server จะทำการเปิดให้บริการทันที



รูปที่ 4.4 แสดงผล ของภาพที่ถูกบันทึกลงบน Hard disk

สรุปผลการทดลอง

จากโครงสร้างที่ได้ออกแบบไว้ในส่วนของการออกแบบนั้น เราสามารถ พัฒนาโปรแกรมออกมาได้ตรงกับที่วางไว้ โดยโปรแกรมจะทำงานได้สอง Mode พร้อมกัน โดยจะทำการบันทึกภาพในทันทีที่เริ่มการใช้งานโปรแกรม ซึ่งในขณะเดียวกันก็ทำการเปิดรอรับการเชื่อมต่อทาง port 7070 ด้วย โดยรวมของการพัฒนาโปรแกรมในส่วน PC สามารถทำงานออกมาได้ครบตามวัตถุประสงค์ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลองโปรแกรมบนบน Embedded Board

ในการทำงานบน Embedded Board นั้นจะต้องผ่านการ Cross Compile โดย Tool ที่แล้วติดตั้งเอาไว้ จากนั้นก็ทำการทดสอบโดยการนำไปใช้งาน บน Embedded Board เพื่อง่ายต่อการ compile เราจะใช้ Makefile ช่วยในการ compile โดยมีขั้นตอนการ Cross Compile ดังนี้

```
CC=arm-linux-gcc
CPP=arm-linux-g++

SERVLIBS= $(MATH_LIB) -lpthread -static
```

```
[root@hardware44 Test-arm]# make
arm-linux-gcc -O2 -DLINUX  -c -o test.o test.c
arm-linux-gcc -O2 -DLINUX  -c -o spcav4l.o spcav4l.c
arm-linux-gcc -O2 -DLINUX  -c -o utils.o utils.c
arm-linux-gcc -O2 -DLINUX  -c -o picture.o picture.c
arm-linux-gcc -O2 -DLINUX  -c -o pargpio.o pargpio.c
arm-linux-gcc -O2 -DLINUX  -c -o encoder.o encoder.c
arm-linux-gcc -O2 -DLINUX  -c -o huffman.o huffman.c
arm-linux-gcc -O2 -DLINUX  -c -o marker.o marker.c
arm-linux-gcc -O2 -DLINUX  -c -o quant.o quant.c
arm-linux-gcc -O2 -DLINUX  -c -o tcputils.o tcputils.c
arm-linux-gcc -O2 -DLINUX  -o test test.o spcav4l.o utils.o picture.o pargpio.o encod
er.o huffman.o marker.o quant.o tcputils.o       -lm  -lpthread -static
```

และเมื่อทำการ make ทำการ compile code ที่ใช้ทั้งหมดออกมาเป็น executable ของ ARM ซึ่งสามารถตรวจสอบได้โดยใช้คำสั่ง file

```
[root@hardware44 Test-arm]# file test
test: ELF 32-bit LSB executable, ARM, version 1 (ARM), statically linked, not stripped
```

เมื่อนำมาทดสอบ ให้ทำการ mount Harddisk เพื่อเข้าไปใช้งาน executable ไฟล์ ที่จะทดสอบ

```
# mount /dev/sda5 /mnt/
# cd mnt/
# ls
          atestv1    atestv4    spcacat    test-p~1   test-p~4   test-p~7
          atestv2    load.sh               test-p~2   test-p~5   testver5
atestv0   atestv3               test-pc    test-p~3   test-p~6   thumbs.db
# ./atestv4 _
```

เมื่อทำการทดสอบ ไฟล์ executable ของ Arm บน Embedded Board จะได้ผลออกมาดังนี้ โดยโปรแกรมจะแสดงการบันทึกภาพลง barddisk ครั้งละ 1 วินาที ซึ่งโปรแกรมจะทำได้ผลออกมาเหมือนกันกับบน Pc และเมื่อมีการเชื่อมต่อเข้าโปรแกรมจะทำการเปิดให้บริการการเชื่อมต่อทันทีและจะทำจนกว่าจะหยุดการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
picture jpeg 31_12_1969/P61432.jpg
picture jpeg 31_12_1969/P61433.jpg
picture jpeg 31_12_1969/P61434.jpg
picture jpeg 31_12_1969/P61435.jpg
picture jpeg 31_12_1969/P61436.jpg
picture jpeg 31_12_1969/P61437.jpg
picture jpeg 31_12_1969/P61438.jpg
picture jpeg 31_12_1969/P61439.jpg
picture jpeg 31_12_1969/P61440.jpg
picture jpeg 31_12_1969/P61441.jpg
picture jpeg 31_12_1969/P61442.jpg
picture jpeg 31_12_1969/P61443.jpg
picture jpeg 31_12_1969/P61444.jpg
picture jpeg 31_12_1969/P61445.jpg
picture jpeg 31_12_1969/P61446.jpg
picture jpeg 31_12_1969/P61447.jpg
picture jpeg 31_12_1969/P61448.jpg
picture jpeg 31_12_1969/P61449.jpg
picture jpeg 31_12_1969/P61450.jpg
picture jpeg 31_12_1969/P61451.jpg
picture jpeg 31_12_1969/P61452.jpg
picture jpeg 31_12_1969/P61453.jpg
picture jpeg 31_12_1969/P61454.jpg
```

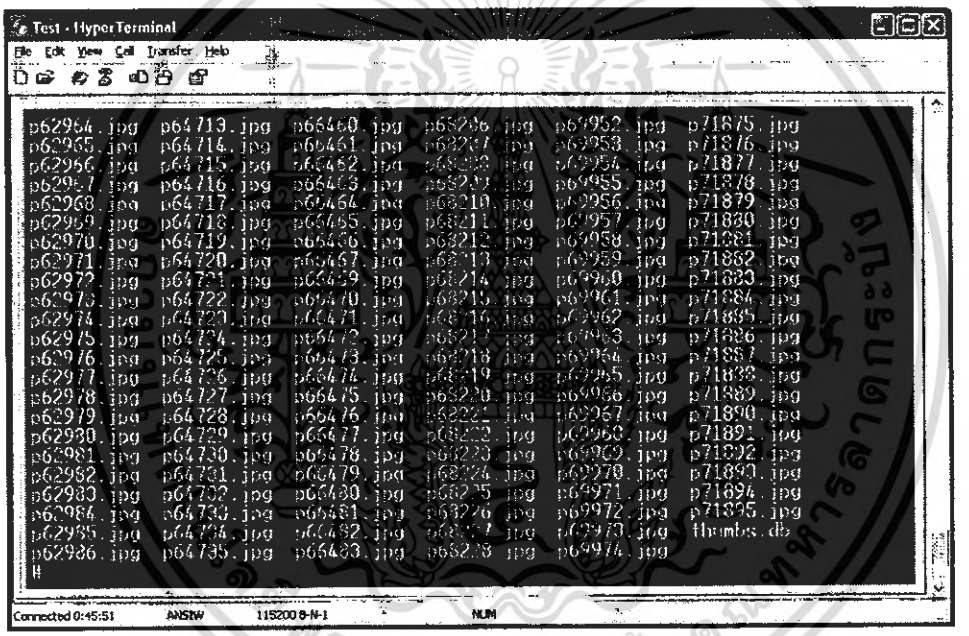
รูปที่ 4.5 แสดงผลการบันทึกภาพทางฝั่ง Server

รูปที่ 4.6 แสดงผลการแสดงภาพทางฝั่ง Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

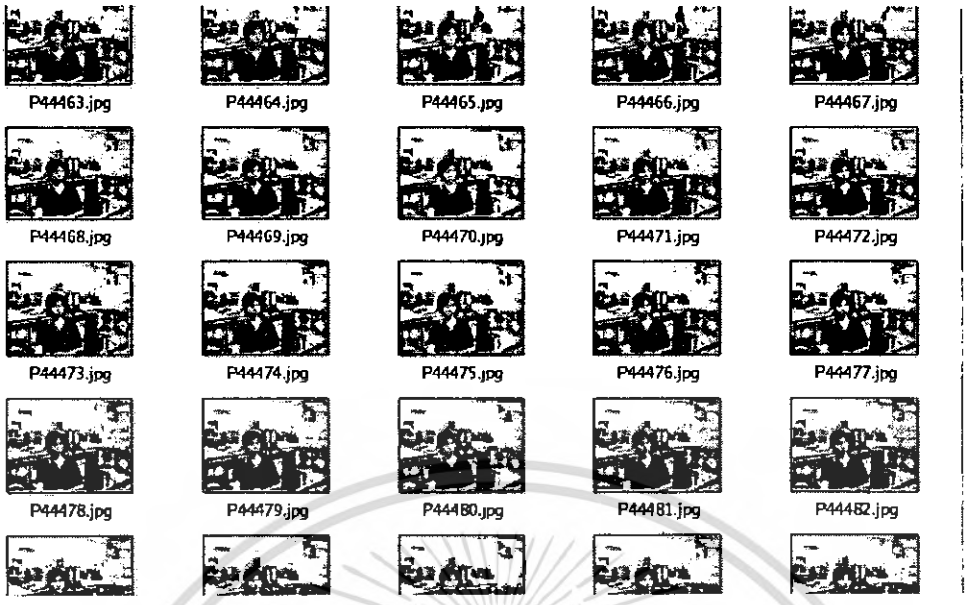
```
 picture jpeg 31_12_1969/P61690.jpg
 picture jpeg 31_12_1969/P61691.jpg
 picture jpeg 31_12_1969/P61692.jpg
 picture jpeg 31_12_1969/P61693.jpg
 picture jpeg 31_12_1969/P61694.jpg
 picture jpeg 31_12_1969/P61695.jpg
 picture jpeg 31_12_1969/P61696.jpg
 picture jpeg 31_12_1969/P61697.jpg
Got connection from 161.246.6.65
VIDIOCSPICT brightnes=33280 hue=49170 color=0 contrast=32768 whiteness=58463dept
h=8 palette=21
VIDIOCSPICT brightnes=33280 hue=49170 color=0 contrast=33280 whiteness=58463dept
h=8 palette=21
VIDIOCSPICT brightnes=32768 hue=49170 color=0 contrast=33280 whiteness=58463dept
h=8 palette=21
VIDIOCSPICT brightnes=32768 hue=49170 color=0 contrast=32768 whiteness=58463dept
h=8 palette=21
 picture jpeg 31_12_1969/P61698.jpg
 picture jpeg 31_12_1969/P61699.jpg
 picture jpeg 31_12_1969/P61700.jpg
 picture jpeg 31_12_1969/P61701.jpg
 picture jpeg 31_12_1969/P61702.jpg
 picture jpeg 31_12_1969/P61703.jpg
```

รูปที่ 4.7 แสดงให้ทราบว่ามีการเชื่อมเข้ามาที่ server จะทำการเปิดให้บริการทันที



รูปที่ 4.8 แสดงผล ของภาพที่กำลังบันทึกลงบน Hard disk

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงผล ของภาพที่ถูกบันทึกลงบน Hard disk

รูปที่ 4.10 แสดงผล การใช้งาน Application ที่ใช้งานจากบน windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 สรุปผลการทดลอง

เมื่อนำโปรแกรมที่พัฒนาบน Pc นำมา Cross Compile แล้วนำมาใช้งานบน Embedded Board ผลของการทำงานจะได้ผลออกมาเหมือนกันดังนี้

1 . สามารถบันทึกภาพลง Harddisk ได้อย่างต่อเนื่อง

2 . บอร์ดทดลอง สามารถจำลองตัวเองเป็น server ส่งภาพออกไปให้เครื่องที่เข้ามาติดต่อได้ตลอดเวลา

3 . สามารถเปิดภาพจาก Harddisk เพื่อดูเหตุการณ์ ย้อนหลังได้

และโปรแกรมที่พัฒนามาก็ตรงกับวัตถุประสงค์ของ โครงงาน ดังนั้นจึงไม่ต้องมีการแก้ไขโปรแกรมในส่วนอื่นๆ ซึ่งสามารถติดต่อเพื่อเข้าดูภาพผ่าน Network socket ได้ โปรแกรมที่พัฒนามายังสามารถเปิดดู ภาพจาก Harddisk ที่บันทึกเอาไว้ได้ ซึ่งพัฒนาโดย C#

Hardware Specification

- สามารถบันทึกภาพลง Harddisk ได้
- ขนาดภาพ 320x240
- รูปแบบ JPEG file
- ความเร็ว 1 frame / sec
- สามารถบันทึกลง Harddisk ขนาด 40 GBytes ได้ 16 วัน
- สามารถทำการบันทึกวนทับได้

Application Specification

มีการทำงาน 2 Mode

1 ขณะทำการเชื่อมต่อ

- ดูภาพผ่าน Network ได้ พร้อมกัน user ตั้งแต่ 2-6 user ขึ้นไป
- บันทึก ภาพในขณะเชื่อมต่อได้
- สามารถปรับ ความสว่าง และ ความเข้มของภาพ ได้

2. ขณะที่ไม่ทำการเชื่อมต่อ

- เปิดดูภาพจาก Harddisk ที่บันทึก ได้
- สามารถเลือกวันและเวลาที่จะทำการดูย้อนหลังได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

5.1 สรุปผลการทำงาน

จากการศึกษาและทำงานในโครงงานนี้สามารถสรุปผลออกมาได้เป็นหัวข้อดังนี้

1 สามารถบันทึกภาพลง Harddisk ได้อย่างต่อเนื่อง

2 บอร์ดทดลอง สามารถจำลองตัวเองเป็น server ส่งภาพออกไปให้เครื่องที่เข้ามาติดต่อได้ตลอดเวลา

3 สามารถเปิดภาพจาก Harddisk เพื่อดูเหตุการณ์ ย้อนหลังได้

4 มีโปรแกรมที่พัฒนาเพื่อ ใช้ในการติดต่อเพื่อเข้าดูภาพผ่านบอร์ดได้ โปรแกรมที่พัฒนามายังสามารถเปิดดู ภาพจาก Harddisk ที่บันทึกเอาไว้ได้ ซึ่งพัฒนาโดย C#

5.2 ปัญหาและอุปสรรคในการทำงาน

ปัญหาและอุปสรรคที่พบในการทำโครงงาน ได้แก่

1. บอร์ดที่ใช้ในการทำโครงงานมีเพียงบางรุ่นเท่านั้นที่มีขายภายในประเทศ และมีราคาค่อนข้างสูง ทำให้เราไม่สามารถหาซื้อเองได้ จึ่งต้องใช้เวลานานในการติดต่อขอยืม ซึ่งจากปัญหานี้ ทำให้การพัฒนาโครงงาน เป็นไปได้ช้าและไม่ตรงตามกำหนด
2. บอร์ดที่ใช้ในการทำโครงงานไม่ตรงกับการออกแบบ เนื่องจากบอร์ดที่ใช้ในโครงงานมีการใช้งานกันน้อยภายในประเทศ และส่วนใหญ่ก็อยู่ในช่วงระหว่างการพัฒนา ดังนั้นเป็นไปได้ยากที่จะติดต่อขอยืมเพื่อนำมาทดสอบในโครงงานดังนั้นจึงจำเป็นที่จะต้องใช้บอร์ดรุ่นอื่นๆทดแทน จึงทำให้ผลของการทดสอบอาจผิดพลาดไป

5.3 แนวทางแก้ไข

เนื่องจากบอร์ดที่ใช้งานไม่ตรงกับที่ได้คาดการณ์ไว้ ดังนั้นถ้ามีการใช้งานในบอร์ด คาดการณ์เอาไว้คาดว่าจะผลออกมาครบตามวัตถุประสงค์ ทำงานก็ยังออกมาได้แต่ไม่เป็นที่น่าพอใจ เนื่องจากยังมีปัญหา ทาง Hard ware บางอย่างที่ไม่สามารถแก้ไขได้

ดังนั้นถ้าให้สามารถใช้งานได้อย่างมีประสิทธิภาพ ควรจะใช้งานบอร์ดให้ตรงกับที่ออกแบบไว้ นั้นคือมีการรองรับการใช้งาน USB อย่างเต็มประสิทธิ์ภาพ และหากนำมาผลิตเป็นชิ้นงานสามารถลดค่าใช้จ่ายต่อชิ้นลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 บทวิจารณ์โครงงาน

เนื่องจากการทำงานกับบอร์ด Embedded จำเป็นต้องใช้งานระบบปฏิบัติการ Linux และจำเป็นต้องมีการใช้งาน Tool ที่ใช้ในการพัฒนา ดังนั้นจำเป็นที่จะต้องมีการศึกษาการทำงานและขั้นตอนการพัฒนา ซึ่งใช้ระยะเวลาในการศึกษามากสำหรับผู้เริ่มต้น และข้อมูลที่มีให้ศึกษามีจำกัด รวมไปถึงตัว บอร์ด Embedded เอง ก็ มีการใช้งานอยู่อย่างไม่แผ่หลาย และส่วนใหญ่จะอยู่ในขั้นตอนการพัฒนา และอาจมีความผิดพลาดที่ยังไม่ทำการแก้ไข ดังนั้นการพัฒนาโครงงานจึงล้าช้าและทำให้การพัฒนาโครงงานมีขอบเขตจำกัด

5.5 แนวทางการพัฒนา

- แนวทางการทำงานที่ผ่านมา สามารถใช้งานโปรแกรมในการทำงานทั้งระบบได้จริงทั้งการทำงานแบบ การบันทึกภาพลง Harddisk ได้อย่างต่อเนื่องและสามารถจัดการในเรื่องของการบันทึกภาพได้ในระดับเบื้องต้นอีกทั้งยังสามารถจำลองตัวเองเป็นserver เพื่อทำการกระจายภาพให้กับ ผู้ใช้งานผ่านทางระบบ network ได้โดยจำเป็นที่จะต้องมีโปรแกรมซึ่งได้พัฒนาร่วมกับการทำโครงงานเพื่อใช้ในการติดต่อเข้าไปเพื่อดูภาพได้

- แนวทางการพัฒนาต่อ แก้ไขและพัฒนารูปแบบการทำงานให้มีความสามารถเพิ่มขึ้นในส่วนของการเชื่อมต่อกับอุปกรณ์ชนิดอื่นๆเช่น ใช้อุปกรณ์ไร้สายเพื่อเชื่อมต่อกับเครือข่าย หรือ การนำจอ LCD เพื่อแสดงภาพที่บันทึกอยู่ได้ รวมไปถึงนำแบตเตอรี่มาต่อเพื่อให้ระบบมีความปลอดภัยเพิ่มขึ้น และสามารถพัฒนาให้ทำงานในลักษณะ ไม่จำเป็นต้องใช้สาย เพื่อนำไปใช้ในงานด้านความปลอดภัยได้อย่างสมบูรณ์ มากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ARM THE ARCHITECTURE FOR THE DIGITAL WORLD.

Available : http://www.arm.com

[2] USB.ORG

Available: http://www.usb.org.

[3] Wikipedia, the free encyclopedia.

Available : http://www.wikipedia.org

[4] Busybox

Available : http://www.busybox.net

[5] A Free World LibLand

Available : http://mxhaard.free.fr/spca5le.html

[6] ThaiGerTec Co.,LTD.

Available : http://thaigertec.com/product.html

[7] Creative Technology Ltd. All rights reserved

Available : http://www.creative.com/

[8] The Linux Kernel Archives

Available : http://www.kernel.org/

[9] ดร. ธนารักษ์ ธีระมั่นคง. "*เทคโนโลยีสมองกลฝังตัว (Embedded Technology)*"

กรุงเทพ ฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย - ญี่ปุ่น) 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code Main Program

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <syslog.h>
#include <linux/parport.h>
#include <linux/ppdev.h>
#include <sys/time.h>
#include "spcaframe.h"
#include "spcav4l.h"
#include "utils.h"
#include "version.h"
#include "tcputils.h"
#include <stdio.h>
#include <termios.h>
#include <unistd.h>

volatile int pietFlag = 0;
static int Oneshoot = 1;
static int numshoot = 0;
static int modshoot = 0;
static int overwrite = 0;
void grab (void);
void grab2 (void);
static int set_timer(int interval);
void service (void *ir);
void sigchld_handler(int s);
int fd = 0;
struct vdIn videoIn;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int
main (int argc, char *argv[])
{
        int videoon = 0;
        char *videodevice = NULL;
        char *partdevice = NULL;
        int usepartport = 0;
        int err;
        int grabmethod = 1;
        int width = 320;
        int height = 240;
        char *separateur;
        char *sizestring = NULL;
        char *mode = NULL;
        int interval = 1000;
        int i;
        int serv_sock,new_sock;
        pthread_t w1;
        pthread_t server_th;
        int sin_size;
        unsigned short serverport = 7070;
        struct sockaddr_in their_addr;
        struct sigaction sa;
        videodevice = "/dev/video";
        int format = VIDEO_PALETTE_JPEG;
        numshoot = 1;
        modshoot = 1;

  for (i = 1; i < argc; i++)
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* skip bad arguments */
        if (argv[i] == NULL || *argv[i] == 0 || *argv[i] != '-')
        {
            continue;
        }
        if (strcmp (argv[i], "-s") == 0) {
            if (i + 1 >= argc) {
                printf ("No parameter specified with -s, aborting.\n");
                exit (1);
            }
            sizestring = strdup (argv[i + 1]);
            width = strtoul (sizestring, &separateur, 10);
        if (*separateur != 'x') {
            printf ("Error in size use -s widthxheight \n");
            exit (1);
        } else {
            ++separateur;
            height = strtoul (separateur, &separateur, 10);
            if (*separateur != 0)
                    printf ("hmm.. dont like that!! trying this height \n");
            printf (" size width: %d height: %d \n",width, height);
            }
        }
             numshoot = 1;
             if (numshoot <= 0) numshoot = 1; // in case :)
             if(!interval) interval = 1000;
             modshoot = 1;
         if (strcmp (argv[i], "-p") == 0) {
            if (argv[i + 1]) {
              interval = (atoi(argv[i + 1])); // timer works on ms
              if ( interval < 50) interval = 50;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            } else {
              printf ("No parameter specified with -p, aborting.\n");
              exit (1);
            }
          }
        if (strcmp (argv[i], "-h") == 0)
        {
          printf ("usage: cdse [-h -s -p ] \n");
          printf ("-h print this message \n");
          printf ("-s  widthxheight      use specified input size \n");
          printf ("-p x ms       take a picture every x ms minimum is set to 50ms \n");
          printf ("-o overwrite picture, each picture come with the same name SpacPict.jpg \n");
          exit (0);
        }
  }

/*------------------- main code ----------------------*/
  printf(" %s \n",version);
  if (videodevice == NULL || *videodevice == 0)
    {
          videodevice = "/dev/video";
          exit (0);
    }
  memset (&videoIn, 0, sizeof (struct vdIn));

  if (init_videoIn (&videoIn, videodevice, width, height, format,grabmethod) != 0)
          if((err= pthread_create (&w1, NULL, (void *) grab, NULL)) != 0){
              printf("thread grabbing error %d \n",err);
              close_v4l (&videoIn);
              exit(1);
          }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
serv_sock = open_sock(serverport);
signal(SIGPIPE, SIG_IGN);      /* Ignore sigpipe */
sa.sa_handler = sigchld_handler;
sigemptyset(&sa.sa_mask);
sa.sa_flags = SA_RESTART;
syslog(LOG_ERR,"Spcaserv Listening on Port  %d\n",serverport);
pthread_create (&w1, NULL, (void *) grab2, NULL);
err = set_timer(interval);
printf("Waiting .... for Incoming Events. CTrl_c to stop !!!! \n");

        /* main wait loop */
        while(videoIn.signalquit){
         sin_size = sizeof(struct sockaddr_in);
          if ((new_sock = accept(serv_sock, (struct sockaddr *)&their_addr, &sin_size)) == -1)
          {
              continue;
          }
        syslog(LOG_ERR,"Got connection from %s\n",inet_ntoa(their_addr.sin_addr));
        printf("Got connection from %s\n",inet_ntoa(their_addr.sin_addr));

        pthread_create(&server_th, NULL, (void *)service, &new_sock);
        }
 pthread_join (w1, NULL);
 close(serv_sock);
 close_v4l (&videoIn);

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//--------------------------Function Check Video In--------------
void
grab (void)
{
int err = 0;
 for (;;)
   {
    err = v4lGrab(&videoIn);
    if (!videoIn.signalquit || (err < 0)){
            break;
            }
   }
}
//-------------------------------------------------------------
typedef void (*sighandler_t)(int);
static sighandler_t original_sighandler;
static void take_snap(int x)
{
            pictFlag=1;
}
static int set_timer(int interval)
{
             struct itimerval itimer;
              itimer.it_interval.tv_sec = interval/1000;
              itimer.it_interval.tv_usec = (interval%1000)*1000;
              itimer.it_value.tv_sec = interval/1000;
              itimer.it_value.tv_usec =(interval%1000)*1000 ;
              original_sighandler = signal(SIGALRM, take_snap);
              setitimer(ITIMER_REAL, &itimer, NULL);
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-------------------Function Server----------------------
void
service (void *ir)
{
        int *id = (int*) ir;
        int frameout = 1;
        struct frame_t *headerframe;
         int ret;
        int sock;
         int framecount=0;
          int retsize=0;
          int ack = 0;
          int err;
          unsigned char x = 0;
          unsigned char y = 0;
          unsigned char wakeup = 0;
          unsigned char cmd = 0;
          unsigned short bright;
          unsigned short contrast;
          struct client_t message;
          sock = *id;
         bright = upbright(&videoIn);
         contrast = upcontrast(&videoIn);
         bright = downbright(&videoIn);
         contrast = downcontrast(&videoIn);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for (;;)
 {
        memset(&message,0,sizeof(struct client_t));
        ret = read(sock,(unsigned char*)&message,sizeof(struct client_t));
        if (ret < 0) {
              printf(" Client vaporished !! \n");
              break;
        }
        if (!ret) break;
        if(ret && (message.message[0] != 'O'))break;
        if(fd) {
              if ( x != message.x ){
                    x = message.x; ;
                    err = port_setdata1(fd,x);
              }
              if ( y != message.y){
                    y = message.y;
                    err = port_setdata2(fd,y);
              }
        }
        if (message.updobright){
              switch (message.updobright){
                    case 1: bright = upbright(&videoIn);
                          break;
                    case 2: bright = downbright(&videoIn);
                          break;
              }
              ack = 1;
        } else if (message.updocontrast){
              switch (message.updocontrast){
                    case 1: contrast = upcontrast(&videoIn);
                          break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 2: contrast = downcontrast(&videoIn);
                    break;
}
        ack = 1;
} else if (message.updoexposure){
        switch (message.updoexposure){
                case 1: spcaSetAutoExpo(&videoIn);
                    break;
                case 2:;
                    break;
  }
        ack = 1;
} else if (message.updosize){
        switch (message.updosize){
                case 1: qualityUp(&videoIn);
                    break;
                case 2: qualityDown(&videoIn);
                    break;
        }
        ack = 1;
} else if (message.fps){
        switch (message.fps){
                case 1: timeDown(&videoIn);
                    break;
                case 2: timeUp(&videoIn);
                    break;
  }
  ack = 1;
} else if (message.sleepon){
  if(fd){
        switch (message.sleepon){
        case 1:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                wakeup = !wakeup;
                break;
                case 2:
                cmd = 1;
                brcak;
                }
            }
            ack = 1;
          } else ack =0;
redo:
    while ((frameout == videoIn.frame_cour) && videoIn.signalquit) usleep(1000);
    if (videoIn.signalquit){
          videoIn.framelock[frameout]++;
          headerframe = (struct frame_t *) videoIn.ptframe[frameout];

           headerframe->acknowledge = ack;
           headerframe->bright = bright;
           headerframe->contrast = contrast;
           headerframe->wakeup = wakeup;
           if(headerframe->size == 0){
            videoIn.framelock[frameout]--;
            frameout = (frameout+1)% OUTFRMNUMB;
           goto redo;
           }
          ret = write_sock(sock, (unsigned char *)headerframe, sizeof(struct frame_t)) ;

          if(!wakeup)
          ret = write_sock(sock,(unsigned char*)(videoIn.ptframe[frameout]+sizeof(struct
frame_t)),headerframe->size);
          else if (fd)
           if ( !read_portstatus15(fd)) {
           wakeup = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }


            if(fd )
            if(cmd ){ //relay command on bit autofeed
              port_toggleAutoFeedbit(fd);
              usleep(100000);
              port_toggleAutoFeedbit(fd);
              cmd = 0;
            }
           videoIn.framelock[frameout]--;
           frameout = (frameout+1)% OUTFRMNUMB;
    } else {
           break;
    }
//---------- end Server code----------


   }
 close_sock(sock);pthread_exit(NULL);
 pthread_exit(NULL);
}

void sigchld_handler(int s)
{
           videoIn.signalquit = 0;
}
void
grab2 (void)
{
int err = 0;
int iframe =0;
unsigned char *pictureData =NULL;
struct frame_t *headerframe;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for (;;)
  {

    err = v4lGrab(&videoIn);
    if (!videoIn.signalquit || (err < 0)){
            break;}

          if(pictFlag){
           iframe=(videoIn.frame_cour +(OUTFRMNUMB-1))% OUTFRMNUMB;
           videoIn.framelock[iframe]++;
           headerframe=(struct frame_t*)videoIn.ptframe[iframe];
           pictureData = videoIn.ptframe[iframe]+sizeof(struct frame_t);
           if(overwrite) {
             getJpegPicture(pictureData,headerframe->w,headerframe-
>h,VIDEO_PALETTE_JPEG,headerframe->size,PICTWRD,NULL);

           }else {
             getJpegPicture(pictureData,headerframe->w,headerframe-
>h,VIDEO_PALETTE_JPEG,headerframe->size,PICTURE,NULL);}
          pictFlag = 0;
          videoIn.framelock[iframe]--;
          }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

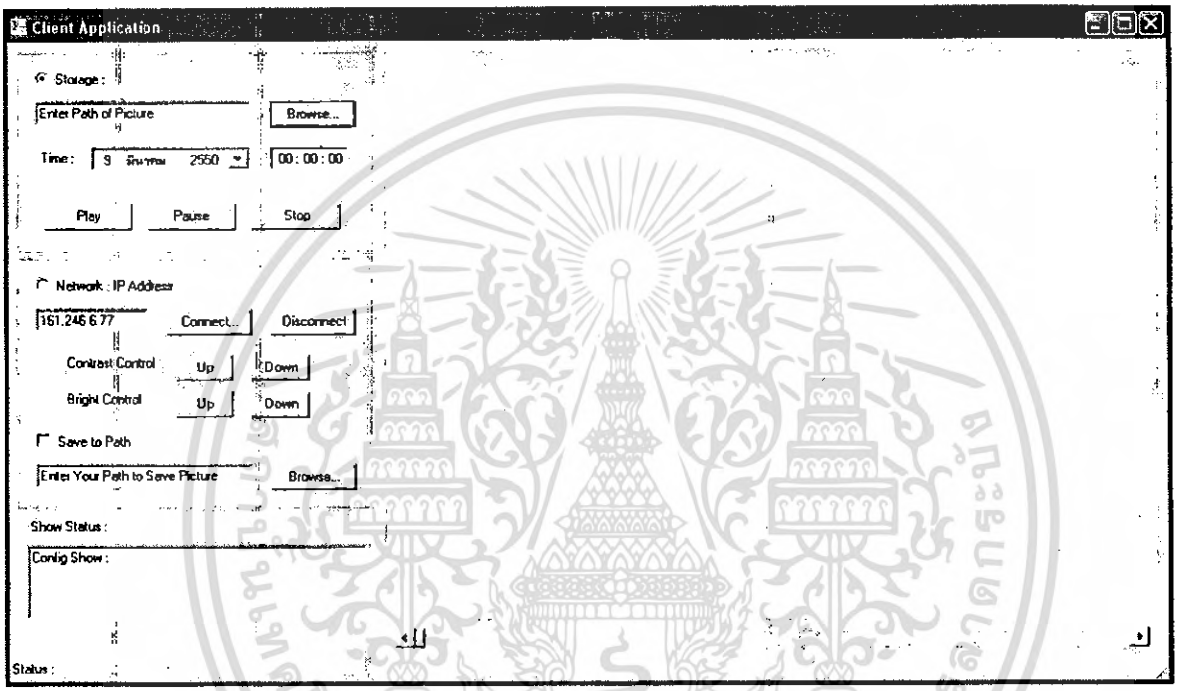
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งานโปรแกรม

ขั้นแรกเมื่อทำการเรียกใช้งานโปรแกรม โปรแกรมจะอยู่ในรูปที่ 1 โดยจะมี Mode การทำงานให้เลือกใช้งานอยู่สอง Mode คือ

1 การเปิดภาพจาก Harddisk

2. การเปิดภาพผ่าน Network แบบ Real time

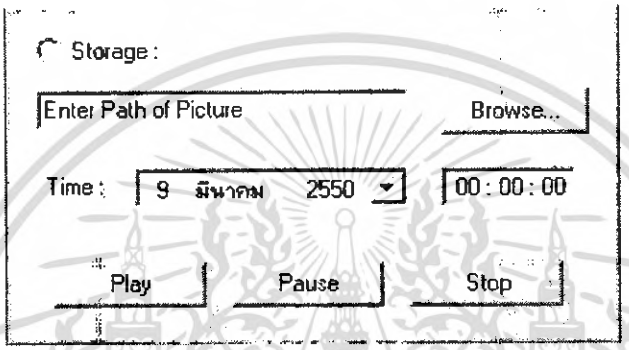


รูปที่ ข-1 ภาพรวมของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

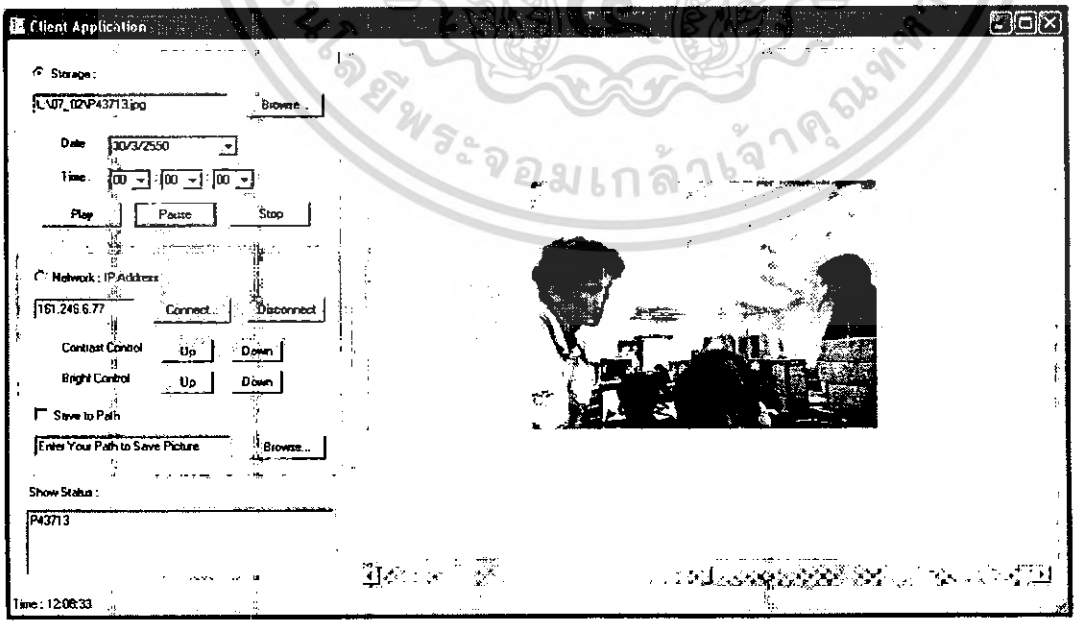
1. **Mode การเปิดภาพจาก Harddisk**

   - เมื่อให้ทำการเลือก Mode โดยการกด เลือกที่ Radio button หน้า Storage
   - จากนั้นให้ทำการเลือกไฟล์ที่จะทำการเล่นโดยใช้ button ที่ชื่อว่า Browse เพื่อทำการเลือกไฟล์จาก Harddisk ที่บันทึกเอาไว้ในวันที่ผ่านๆมา
   - หรือให้ทำการเลือก ใน List ที่มีชื่อว่า Time ซึ่งโปรแกรมจะทำการค้นหา ว่าไฟล์ที่อยู่ใน Harddisk นั้นมีการบันทึกเอาไว้ ในเวลาเท่าใด แล้วแสดงออกมาใน List เพื่อให้ผู้ใช้งานเลือก



รูปที่ ข-2 mode การเปิดภาพจาก Storage

- เมื่อทำการเลือกไฟล์ที่จะทำการเปิดเรียบร้อยแล้วให้ทำการกดปุ่ม Play เพื่อให้ทำการเล่นภาพ และกดปุ่ม Pause เพื่อทำการหยุดชั่วคราว และ Stop เพื่อทำการหยุดเล่น

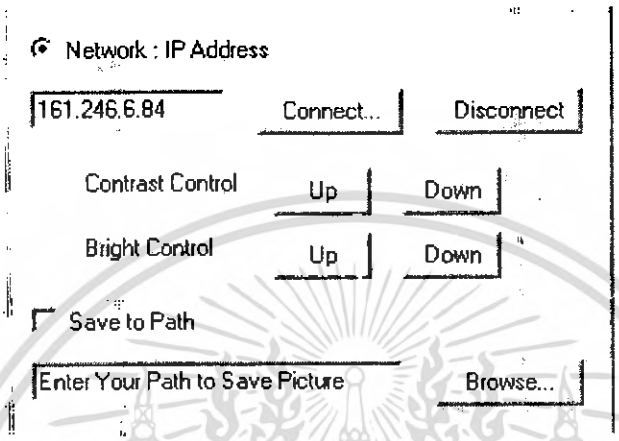


รูปที่ ข-3 แสดงการทำงานของ Mode การเปิดภาพจาก Storage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การเปิดภาพผ่าน Network แบบ Real time

- เมื่อให้ทำการเลือก Mode โดยการกด เลือกที่ Radio button หน้า Network : IP Address
- จากนั้นให้ทำการใส่ค่า IP Address ของBoard ที่เปิดบริการ อยู่เพื่อนำไปใช้ในการเชื่อมต่อ



รูปที่ ข-4 mode การเชื่อมต่อผ่าน network

- ให้ทำการกดปุ่ม Connect เพื่อเป็นการเริ่มทำการเชื่อมต่อและสามารถทำการปรับความสว่างของภาพได้ และหากต้องการเลิกใช้งานให้ทำการกด Disconnect

รูปที่ ข-5 แสดงการทำงานของ Mode การเชื่อมต่อผ่าน network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้