

บทที่ 7

การใช้งานระบบเว็บเซิร์ฟเวอร์แบบฝังตัว

การใช้งานระบบเว็บเซิร์ฟเวอร์แบบฝังตัวนั้น เริ่มต้นด้วยการการตั้งค่าต่างๆ ของระบบเครือข่ายและการควบคุมการใช้งานในระดับโปรแกรมประยุกต์ ซึ่งวัตถุประสงค์ของการตั้งค่าดังกล่าว เพื่อให้ระบบเว็บเซิร์ฟเวอร์แบบฝังตัวสามารถติดตั้งเข้ากับระบบเครือข่ายทั่วไปซึ่งมีการกำหนดหมายเลขระบบเครือข่ายที่แตกต่างกัน พร้อมทั้งตรวจสอบการทำงานของเว็บเซิร์ฟเวอร์แบบฝังตัวในการสื่อสารข้อมูลผ่านระบบเครือข่ายไปพร้อมกันด้วย และการตั้งค่าในระดับโปรแกรมประยุกต์คือการเตรียมระบบเว็บเซิร์ฟเวอร์แบบฝังตัว เพื่อให้ปรับเปลี่ยนและควบคุมอุปกรณ์ต่อพ่วงต่างๆ โดยการกำหนดภาษาสคริปต์และระบบรักษาความปลอดภัย

7.1. การกำหนดหมายเลข IP ให้กับระบบเว็บเซิร์ฟเวอร์แบบฝังตัว

เนื่องจากระบบเครือข่ายโดยทั่วไปนั้น แต่ละระบบเครือข่ายจะมีหมายเลข IP ที่แตกต่างกัน ดังนั้นเพื่อให้ระบบเว็บเซิร์ฟเวอร์แบบฝังตัวสามารถทำงานได้บนระบบเครือข่ายทุกแห่ง ในที่นี้จึงให้ผู้ใช้งานทำการกำหนดหมายเลข IP ให้กับระบบเว็บเซิร์ฟเวอร์แบบฝังตัวด้วยตัวเอง การกำหนดหมายเลข IP นั้นทำได้โดยการแก้ไขตาราง ARP ให้กับเครื่องไคลเอ็นต์ที่ต้องการเรียกใช้ เช่น กรณีต้องการให้เว็บเซิร์ฟเวอร์แบบฝังตัวมีหมายเลข IP เป็น 10.0.1.250 สามารถทำได้ดังนี้

บนระบบ Unix

```
arp -s 10.0.1.250 00:40:05:50:4f:4b
```

บนระบบ Window

```
arp -s 10.0.1.250 00-40-05-50-4f-4b
```

ผลลัพธ์ที่ได้จากการทำงานของคำสั่งดังกล่าวทำให้เครื่องไคลเอ็นต์ กำหนดตาราง ARP เพื่อชี้หมายเลขตำแหน่ง IP 10.0.1.250 ไปยัง หมายเลขตำแหน่ง MAC ของระบบเว็บเซิร์ฟเวอร์แบบฝังตัว โดยในที่นี้หมายเลขดังกล่าวคือ 00:40:05:50:4f:4b หลังจากนั้นเมื่อมีการส่ง IP Packet เข้าสู่ระบบเว็บเซิร์ฟเวอร์แบบฝังตัว ระบบจะทำการจดจำหมายเลขตำแหน่ง IP ไว้เพื่อกำหนดเป็นหมายเลขตำแหน่ง IP ของเว็บเซิร์ฟเวอร์แบบฝังตัวต่อไป ซึ่ง IP Packet ในที่นี้อาจเป็น Ping หรือ TCP Packet ก็ได้

7.2. การทดสอบการทำงานในระดับเครือข่าย

การทดสอบการทำงานของระบบเครือข่ายนั้น วิธีการที่ง่ายที่สุดคือการทดสอบด้วยโปรแกรม PING ซึ่งพบทั้งบน Unix และ Window โดยทำการ Ping หมายเลข IP ของระบบเว็บเซิร์ฟเวอร์แบบฝังตัว เช่นหากต้องการตรวจสอบว่าระบบเว็บเซิร์ฟเวอร์แบบฝังตัวยังคงทำงานติดต่อกับระบบเครือข่ายหรือไม่ ให้ทดสอบโดยการใช้คำสั่ง ping ตามด้วยหมายเลข IP ที่ต้องการทดสอบ

หากระบบยังคงเชื่อมต่ออยู่ผลลัพธ์ที่ได้ (ดูในภาพที่ 32) จะแสดงความเร็วในการตอบรับของระบบเว็บเซิร์ฟเวอร์แบบฝังตัว ซึ่งการทดสอบพบว่า ระบบเว็บเซิร์ฟเวอร์จะตอบรับข้อมูลโดยใช้เวลาเฉลี่ย 20.7ms เมื่อทำงานปกติ ในกรณีที่ระบบไม่ตอบรับนั้น ให้ทำการตรวจสอบสถานะของการเชื่อมต่อและสายสัญญาณต่างๆ ของระบบเครือข่าย

```
$ ping 10.0.1.250
PING 10.0.1.250 (10.0.1.250) from 10.0.1.10 : 56 data
bytes
64 bytes from 10.0.1.250: icmp_seq=0 ttl=64 time=20.7 ms
64 bytes from 10.0.1.250: icmp_seq=1 ttl=64 time=20.7 ms
64 bytes from 10.0.1.250: icmp_seq=2 ttl=64 time=20.7 ms
```

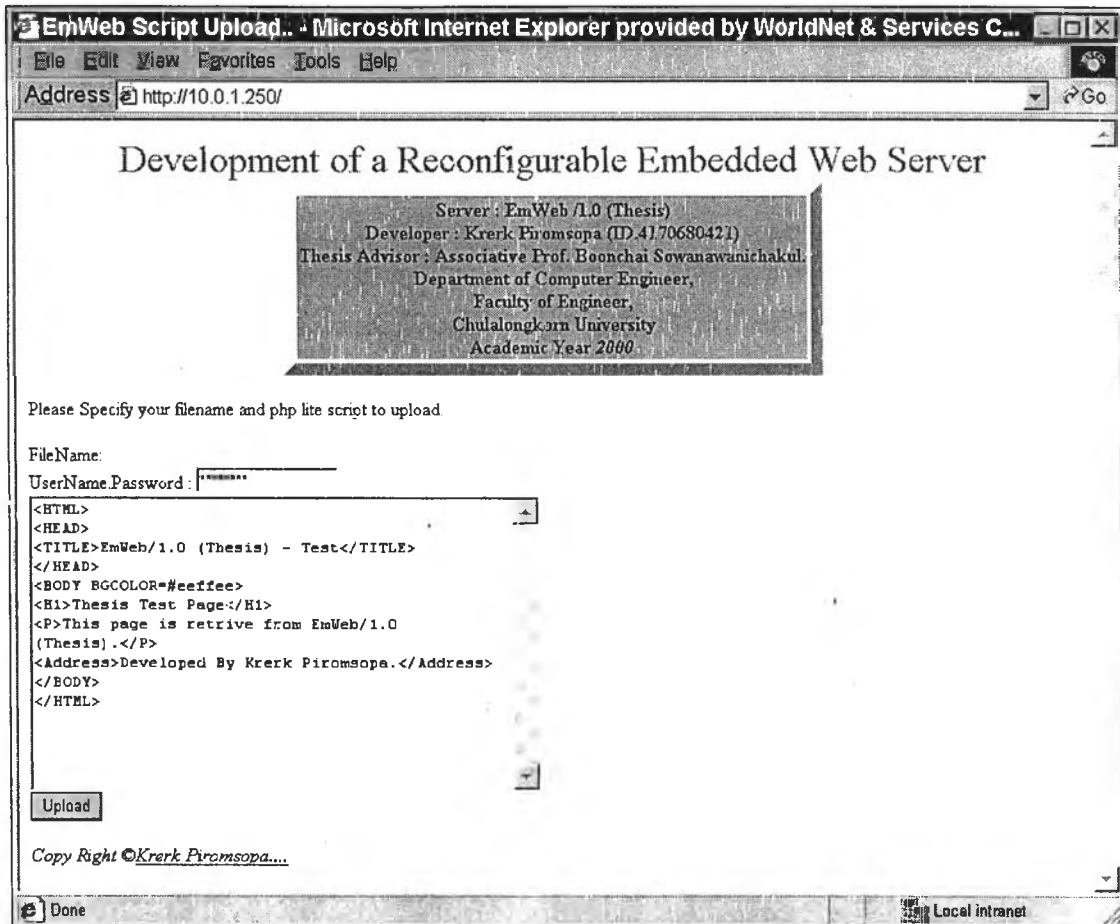
ภาพที่ 32 ผลลัพธ์จากการทดสอบสถานะทางเครือข่ายด้วยคำสั่ง ping

7.3. การตั้งค่าสคริปต์เข้าสู่ระบบเว็บเซิร์ฟเวอร์แบบฝังตัว

ในการตั้งค่าภาษาสคริปต์นั้น ผู้ใช้งานจะพัฒนาภาษาสคริปต์และหน้าเว็บโดยใช้โปรแกรมแก้ไขข้อมูลทั่วไป เช่น NOTEPAD หรือ PICO จากนั้นให้ส่งค่าที่ได้ดังกล่าวเข้าสู่ระบบโดยการกำหนด URL บนโปรแกรมค้นผ่านเว็บ ให้ชี้ไปยังหมายเลข IP ของระบบเว็บเซิร์ฟเวอร์แบบฝังตัว เช่น <http://10.0.1.250/> เป็นต้น

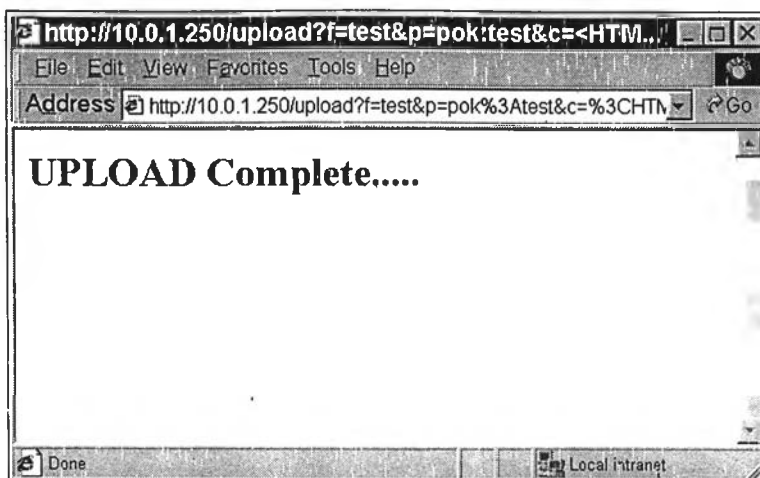
ระบบเว็บเซิร์ฟเวอร์แบบฝังตัว จะตอบรับกลับมาเป็นฟอร์ม (ดูตัวอย่างภาพที่ 32) ให้ผู้ใช้ตั้งค่า Username, Password และสคริปต์ลงในช่องว่างที่กำหนด โดยชื่อผู้ใช้และรหัสผ่านนั้น จะกำหนดในช่อง Username:Password โดยใช้เครื่องหมาย ":" (Colon) เป็นตัวขึ้น เช่นกรณีต้องการกำหนดชื่อผู้ใช้เป็น pok และ รหัสผ่านเป็น test ให้กำหนด Username:Password เป็น "pok:test" และใส่ข้อมูลภาษาสคริปต์ที่ต้องการลงในช่องว่างข้างล่าง (ทั้งนี้ผู้ใช้อาจอาศัย

โปรแกรมประยุกต์อื่นๆ เพื่อช่วยในการสร้างหน้าเว็บก็ได้ เช่น โปรแกรม Microsoft Front Page หรือ โปรแกรม Dreamweaver) เมื่อเสร็จสิ้นสมบูรณ์แล้วจึงกดปุ่ม Upload เพื่อส่งข้อมูลทั้งหมดเข้าสู่ระบบ



ภาพที่ 33 การส่งสคริปต์เข้าสู่ระบบเว็บเซิร์ฟเวอร์แบบฝังตัว

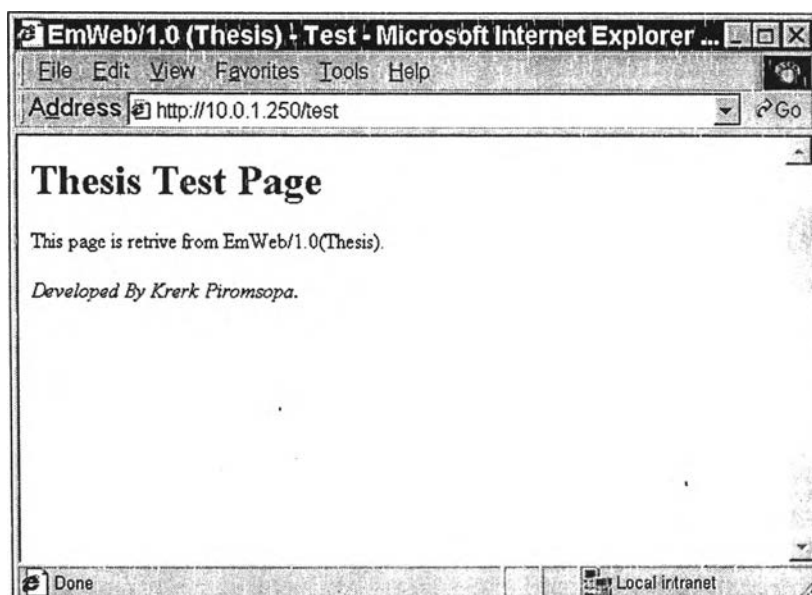
เมื่อระบบได้รับข้อมูลแล้วจะตอบรับการรับข้อมูลด้วยความ Upload Complete ดังแสดงในภาพที่ 34 อย่างไรก็ตามเนื่องจากระบบดังกล่าว สามารถรองรับได้เพียงการส่งข้อมูลแบบ Method Get เท่านั้น ทำให้มีข้อจำกัดในเรื่องของความยาวของข้อมูลที่ส่งเข้าสู่ระบบ ซึ่งหากระบบไม่ทำการตอบรับใดๆ หมายความว่าสคริปต์ที่ส่งเข้าสู่ระบบนั้นอาจมีความยาวมากเกินไป การแก้ไขสามารถกระทำได้โดยการปรับสคริปต์ให้สั้นลง และส่งเข้าสู่ระบบใหม่อีกครั้งหนึ่ง



ภาพที่ 34 ผลลัพธ์จากการส่งข้อมูลเข้าสู่ระบบเว็บเซิร์ฟเวอร์แบบฝังตัว

7.4. การเรียกใช้งานสคริปต์บนระบบเว็บเซิร์ฟเวอร์แบบฝังตัว

เมื่อผู้ใช้งานได้ทำการส่งสคริปต์เข้าสู่ระบบเว็บเซิร์ฟเวอร์แบบฝังตัวแล้ว ผู้ใช้สามารถเรียกใช้งานสคริปต์ดังกล่าวได้โดยการกำหนด URL ให้ชี้ไปยังไฟล์ชื่อ test ของระบบเช่น <http://10.0.1.250/test> (ดูตัวอย่างผลลัพธ์ในภาพที่ 35) ซึ่งระบบจะทำการประมวลผลผลลัพธ์ที่ได้จากสคริปต์ดังกล่าวและแสดงผลออกมาทางหน้าเว็บ ทั้งนี้โปรแกรมค้นผ่านเว็บจะถาม Username และ Password เพื่อเข้าถึงข้อมูลดังกล่าว (ตามตัวอย่างในบทที่ 5) ให้ผู้ใช้ใส่ชื่อผู้ใช้และรหัสผ่านตามที่กำหนดไว้ระหว่างการส่งสคริปต์เข้าสู่ระบบ ซึ่งหากระบบเว็บว่าชื่อผู้ใช้และรหัสผ่านผิดพลาด ระบบเว็บเซิร์ฟเวอร์แบบฝังตัวจะไม่อนุญาตให้ผู้ใช้เรียกดูข้อมูลดังกล่าว



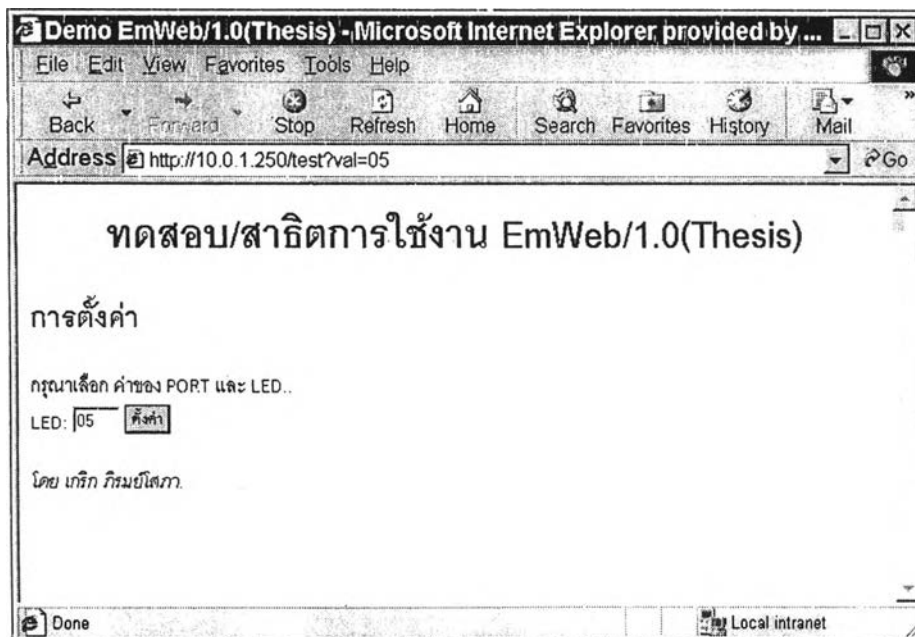
ภาพที่ 35 ตัวอย่างผลลัพธ์จากการเรียกใช้งานสคริปต์

ทั้งนี้เนื่องจากระบบเว็บเซิร์ฟเวอร์แบบฝังตัวถูกกำหนดให้มีการตั้งชื่อผู้ใช้และรหัสผ่านทุกครั้ง เมื่อมีการตั้งค่าสคริปต์เข้าสู่ระบบ ดังนั้นในการเรียกใช้งานสคริปต์เพื่อแสดงผลลัพธ์นี้ หากมีการเปลี่ยนแปลงรหัสผ่าน หรือเป็นการเรียกใช้งานครั้งแรกเมื่อมีการเปิดโปรแกรมผ่านเว็บนั้น ผู้ใช้จะต้องระบุชื่อผู้ใช้และรหัสผ่านเสมอ แต่หากเคยมีการระบุชื่อผู้ใช้และรหัสผ่านที่ถูกต้องไว้ก่อนแล้ว โปรแกรมผ่านเว็บจะส่งค่ารหัสผ่านนั้นในการเรียกขอข้อมูลทุกครั้งจนกว่าจะปิดโปรแกรม (ดูตัวอย่างการใส่ชื่อผู้ใช้และรหัสผ่านได้ในภาพที่ 28 บทที่ 5)

7.5. การรับค่าผ่านฟอร์มเพื่อใช้ประมวลผลในภาษาสคริปต์

การรับค่าผ่านฟอร์มเพื่อใช้สำหรับการประมวลผลในภาษาสคริปต์นั้น จัดทำขึ้นเพื่อให้ผู้ใช้งานสามารถโต้ตอบกับระบบควบคุมแบบฝังตัวได้โดยอาศัยการทำงานของโปรแกรมผ่านเว็บ โดยผู้ใช้งานสามารถอ้างถึงตัวแปรต่างๆ บนหน้าเว็บเพื่อใช้งานในสคริปต์ได้ทันที (ดังตัวอย่างในบทที่ 6) อย่างไรก็ตามข้อควรระวังในการอ้างอิงตัวแปรบนฟอร์มไปใช้งานในภาษาสคริปต์คือ ข้อมูลทุกตัวจะถูกจัดเก็บเป็นตัวแปรประเภทสายอักขระมิใช่ตัวแปรประเภทจำนวน ซึ่งมีควรใช้ค่าดังกล่าวเพื่อการคำนวณใดทางคณิตศาสตร์โดยไม่ทำการแปลงค่าก่อน

ตัวอย่างเช่น เราต้องการรับค่าจาก TEXT บนฟอร์ม HTML ชื่อ val เพื่อนำค่าที่ได้แสดงออกทางพอร์ตของระบบควบคุมแบบฝังตัว (ดังตัวอย่างในภาพที่ 36) ผู้ใช้สามารถพัฒนาสคริปต์ โดยการอ้างอิงตัวแปร val ด้วย \$val และแสดงผลลัพธ์ออกยังพอร์ต 0x2000 ด้วยประโยค outp ซึ่งตัวอย่างสคริปต์เพื่อสามารถผลลัพธ์ดังกล่าว สามารถแสดงดังภาพที่ 36 ทั้งนี้รายละเอียดเพิ่มเติมเกี่ยวกับคำสั่งต่างๆ ในภาษา PHP Lite นั้น สามารถดูได้ในบทที่ 6



ภาพที่ 36 ตัวอย่างหน้าเว็บในการรับค่าผ่านฟอร์ม

```

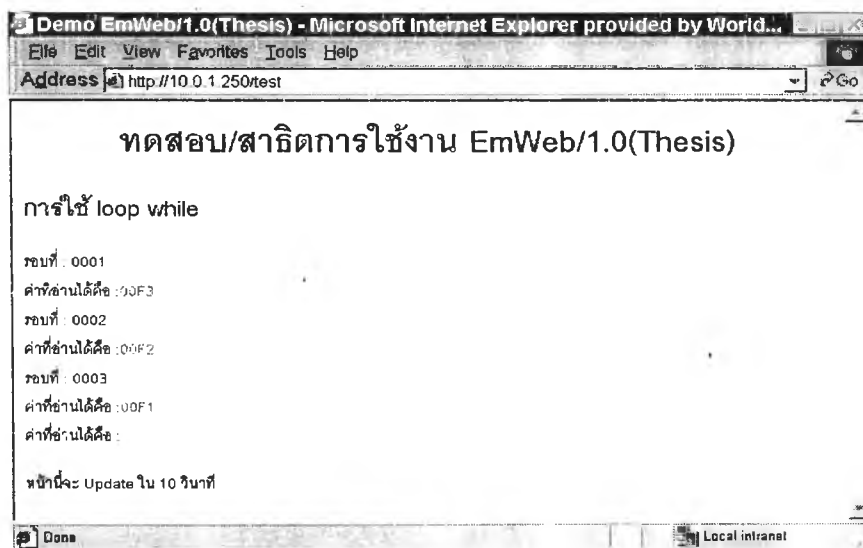
<HTML>
<HEAD><TITLE>Demo EmWeb/1.0(Thesis)</TITLE></HEAD>
<BODY BGCOLOR=lightyellow>
<H1 ALIGN=CENTER>ทดสอบ/สาธิตการใช้งาน EmWeb/1.0(Thesis)</H1>
<H2><FONT COLOR=BLUE>การตั้งค่า</FONT></H2>
<FORM METHOD=GET>
กรุณาเลือก ค่าของ PORT และ LED..<BR>
LED: <INPUT TYPE=TEXT NAME=val VALUE="00" SIZE=4>
<INPUT TYPE=SUBMIT VALUE="ตั้งค่า">
</FORM>
<%outp(0x2000,$val);%>
<Address>โดย เกริก ภิรมย์โสภะ.</Address>
</BODY>
</HTML>

```

ภาพที่ 37 ตัวอย่างสคริปต์ในการรับค่าผ่านฟอร์ม

7.6. การทำงานของภาษาสคริปต์ที่ซับซ้อนยิ่งขึ้น

ในการทำงานที่ซับซ้อนยิ่งขึ้น ผู้ใช้อาจอาศัยการ Refresh ของโปรแกรมค้นผ่านเว็บเพื่อดึงข้อมูลสถานะของระบบตามช่วงเวลาที่กำหนด หรืออาศัยการทำงานแบบวนรอบเพื่อตั้งค่าหรืออ่านค่าจากอุปกรณ์ต่อพ่วงหลายตัวพร้อมกันดังตัวอย่างในภาพที่ 38 แสดงการวนรอบเพื่ออ่านข้อมูล ซึ่งโปรแกรมค้นผ่านเว็บจะทำการร้องขอข้อมูลทุกๆ 10 วินาที



ภาพที่ 38 ตัวอย่างผลลัพธ์จากการ Refresh และการทำงานแบบวนรอบ

การ Refresh นั้น สามารถกระทำได้หลายแนวทางเช่นการใช้ Meta Tag หรือการใช้ JavaScript ทั้งนี้การ Refresh ที่ง่ายที่สุดคือการใช้ Meta Tag สำหรับการวนรอบนั้น ในภาษาสคริปต์ที่กำหนดขึ้นนี้มีการวนรอบแบบ while ให้ผู้พัฒนาเลือกใช้ โดยตัวอย่างการพัฒนาสคริปต์เพื่อการทำงานแบบวนรอบและการใช้ Meta Tag ในการ Refresh นั้น สามารถแสดงได้ดังภาพที่

39

```
<HEAD>
<TITLE>Demo EmWeb/1.0(Thesis)</TITLE>
<META HTTP-EQUIV="Refresh" CONTENT="10;URL=/test">
</HEAD>
<H1 ALIGN=CENTER>ทดสอบ/สาริตการใช้งาน EmWeb/1.0(Thesis)</H1>
<H2><FONT COLOR=BLUE>การใช้ loop while</FONT></H2>
<%
$I=3;
while ($i) {
$s=4-$i; str($s);
echo("รอบที่ : ".$s."<BR>");
outp(0x2000,$i);inp(0x2000,$val); str ($val);
$I=$i-1;
}%>
ค่าที่อ่านได้คือ :<FONT COLOR=RED>
<% echo($val):%>
<BR></FONT>
<% } %>
<P>หน้านี้จะ Update ใน 10 วินาที</P>
```

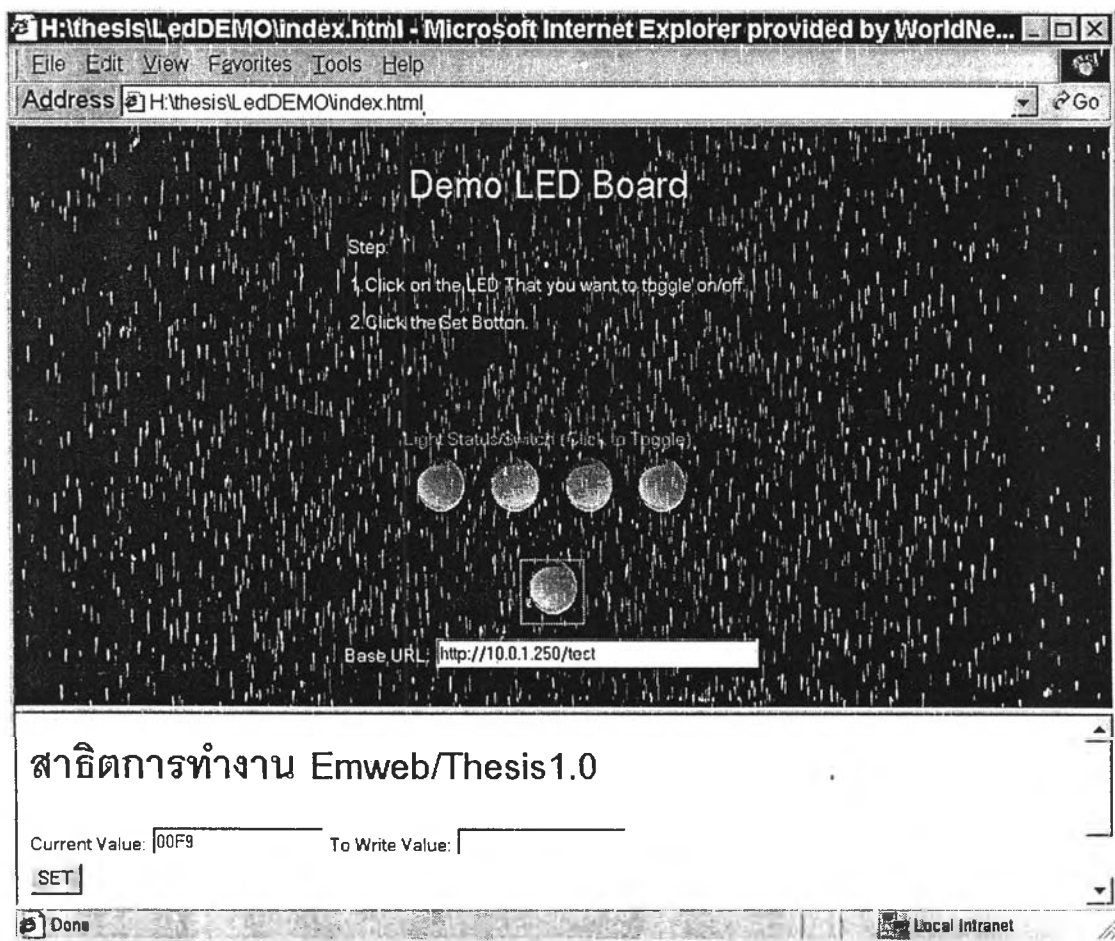
ภาพที่ 39 ตัวอย่างสคริปต์การทำงานแบบมีการ Refresh และการทำงานแบบวนรอบ

7.7. การทำงานร่วมกับภาษาอื่น ๆ เพื่อการควบคุมที่ซับซ้อนยิ่งขึ้น

ในการทำงานที่ซับซ้อนยิ่งขึ้น ภาษาสคริปต์ที่ฝังอยู่บนระบบควบคุมแบบฝังตัว อาจไม่เพียงพอต่อการควบคุม ทั้งนี้เนื่องจากข้อจำกัดต่างๆ ของภาษาสคริปต์ และระบบควบคุมแบบฝังตัวที่มีหน่วยความจำจำกัด ไม่สามารถเก็บข้อมูลที่มีขนาดใหญ่ ดังนั้นจึงอาจมีการเขียนโปรแกรมหรือสคริปต์ด้วยภาษาอื่น เพื่อเป็น Component ที่สนับสนุนการทำงานของระบบเว็บ

เซิร์ฟเวอร์แบบฝังตัวโดยฝากไว้ที่เซิร์ฟเวอร์อื่นๆ ซึ่งทำหน้าที่เป็นที่เลี้ยงช่วยในการเก็บข้อมูลขนาดใหญ่

ตัวอย่างของ Component ในที่นี้พัฒนาด้วย JavaScript เพื่อใช้ในการทำงานร่วมกับอุปกรณ์ทดสอบ เพื่อทำการควบคุมการเปิดปิดสวิตช์ไฟ สคริปต์ประกอบด้วย 2 ส่วนคือ JavaScript ซึ่งเก็บอยู่ที่เว็บเซิร์ฟเวอร์ภายนอกหรือเครื่องคอมพิวเตอร์ของผู้ใช้งาน และ สคริปต์เพื่อฝังในเว็บเซิร์ฟเวอร์แบบฝังตัวเพื่อทำงานร่วมกับ Component นั้นๆ ทั้งนี้ผู้ใช้งานจะต้องทำการส่งสคริปต์ส่วนที่เป็น PHP Lite เข้าสู่ระบบตามขั้นตอนปกติก่อน แล้วจึงเรียก URL ไปยังหน้าเว็บที่เก็บส่ง Component นั้นๆ ซึ่งลักษณะการควบคุมด้วย Component ดังกล่าว สามารถอธิบายได้ดังภาพที่ 40 โดยผู้ใช้งานสามารถเปิด-ปิด LED ได้โดยการกดที่รูป LED ทั้ง 4 ดวง จากนั้นจึงทำการกดที่รูป LED ดวงกลางเพื่อปรับเปลี่ยนสถานะของอุปกรณ์ทดสอบ



ภาพที่ 40 ตัวอย่างผลลัพธ์การประยุกต์ใช้ JavaScript บนเว็บเซิร์ฟเวอร์แบบฝังตัว

เพื่อให้สามารถควบคุมได้ดังภาพที่ 40 จะต้องมีการพัฒนา Component เป็น JavaScript เพื่อช่วยรองรับ Event ที่เกิดจากการกดที่รูป LED และเตรียมข้อมูลก่อนส่งเข้าสู่ระบบเว็บเซิร์ฟเวอร์แบบฝังตัว โดยสคริปต์ทั้งหมดแบ่งออกเป็น 2 ส่วนคือ สคริปต์ส่วนที่ฝังอยู่ในระบบเว็บเซิร์ฟเวอร์แบบฝังตัว พัฒนาด้วย PHP Lite และ หน้าเว็บส่วนที่ฝากไว้ยังเว็บเซิร์ฟเวอร์ภายนอก ซึ่งประกอบด้วยรูปภาพต่างๆ และ JavaScript Component สคริปต์ทั้ง 2 ส่วนนี้ สามารถแสดงได้ดังภาพที่ 41 และ 42 ตามลำดับ

```
<BODY BGCOLOR="lightyellow">
<H1>สาริตถการทำงาน Emweb/Thesis1.0</H1>
<form method="get">
<%
outp(0x2000,$wr);
inp(0x2000,$val); str($val);
%>
Current Value:
<INPUT TYPE=TEXT NAME=rd VALUE=<% echo ($val);%>><RB>
To Write Value:
<INPUT TYPE=TEXT NAME=wr><BR>
<INPUT TYPE=SUBMIT VALUE="SET">
</form>
<address>โดย เกริก ภิรมย์โสภา</address>
</BODY>
```

ภาพที่ 41 ตัวอย่างสคริปต์ส่วนที่ฝังในเว็บเซิร์ฟเวอร์แบบฝังตัว

จากภาพที่ 41 นั้นจะพบว่าสคริปต์จะรับค่าจากตัวแปร wr เพื่อเขียนออกสู่พอร์ต 0x2000 ดังนั้นหน้าที่ของสคริปต์ส่วนที่เป็น Component คือการสร้าง URL เพื่อร้องขอข้อมูลจากระบบเว็บเซิร์ฟเวอร์แบบฝังตัวให้มีการส่งผ่านค่าตัวแปร wr ตามต้องการเช่น กรณีต้องการตั้งค่า wr เป็น 5 สคริปต์ส่วน Component จะต้องสร้าง URL เป็น <http://10.0.1.250/test?wr=05> เป็นต้น จากการทำงานในลักษณะดังกล่าว ผู้พัฒนาอาจพัฒนา Component นี้ด้วย JavaScript ดังแสดงในภาพที่ 42

```

<HTML><HEAD><TITLE>LED Module</TITLE></HEAD><BODY BGCOLOR="BLACK">
<SCRIPT LANGUAGE="JavaScript1.2">
function setimage(newval)
{ // Set Picture
document.led3.src= (newval & 8)? "1.gif":"0.gif"; document.led2.src= (newval & 4)? "1.gif":"0.gif";
document.led1.src= (newval & 2)? "1.gif":"0.gif"; document.led0.src= (newval & 1)? "1.gif":"0.gif"; }
function toggleval(val)
{ StrConst = new Array();
var newval;
StrConst[0] = "00";StrConst[1] = "01";StrConst[2] = "02";StrConst[3] = "03";
StrConst[4] = "04";StrConst[5] = "05";StrConst[6] = "06";StrConst[7] = "07";
StrConst[8] = "08";StrConst[9] = "09";StrConst[10] = "0J";StrConst[11] = "0K";
StrConst[12] = "0L";StrConst[13] = "0M";StrConst[14] = "0N";StrConst[15] = "0O";
Document.forms[0].dec.value ^= val; Newval = document.forms[0].dec.value;
Document.forms[0].hex.value =StrConst[newval]; Setimage(newval); }
Function setvalue()
{ self.parent.frames["sub"].location = document.forms[0].burl.value + "?wr=" + document.forms
[0].hex.value; }
</SCRIPT><CENTER><FONT COLOR="YELLOW" SIZE=-3>Demo LED Board</FONT><BR>
<TABLE><TR><TD><FONT COLOR="CYAN"><P ALIGN=LEFT>Step:<BR>1.Click on the LED That
you want to toggle on/off.<BR>2.Click the Set Botton.</P></FONT></TD></TR></TABLE><BR>
<BR><FONT COLOR="RED">Light Status/Switch (Click to Toggle):</FONT><BR>
<A HREF="javascript:toggleval(8);"><IMG NAME=led3 SRC="" BORDER=0></A>&nbsp;
<A HREF="javascript:toggleval(4);"><IMG NAME=led2 SRC="" BORDER=0></A>&nbsp;
<A HREF="javascript:toggleval(2);"><IMG NAME=led1 SRC="" BORDER=0></A>&nbsp;
<A HREF="javascript:toggleval(1);"><IMG NAME=led0 SRC="" BORDER=0></A>
<BR><FORM><A HREF="javascript:setvalue();"><IMG SRC="1.gif" ALT="SET" BORDER=0></a>
</CENTER><INPUT TYPE=HIDDEN NAME="dec" VALUE="0"><INPUT TYPE=HIDDEN NAME="hex"
VALUE="00"><CENTER><FONT COLOR="ORANGE">Base URL: </FONT><INPUT TYPE=TEXT
NAME="burl" SIZE=40></CENTER></FORM>
<SCRIPT LANGUAGE="JavaScript1.2">
setimage(document.forms[0].dec.value);
</SCRIPT></BODY></HTML>

```

ภาพที่ 42 ตัวอย่าง JavaScript เพื่อช่วยในการควบคุมที่ซับซ้อน

นอกจากนี้เพื่อความสะดวกในการแสดงผลของหน้าเว็บส่วน Component ที่พัฒนาโดย JavaScript และส่วนที่ฝังในระบบเว็บเซิร์ฟเวอร์แบบฝังตัว จึงอาศัยการแบ่งเฟรม และแปลงค่าจาก Event ต่างๆ ของผู้ใช้เป็นค่าที่ต้องการเพื่อส่งผ่านไปยังระบบเว็บเซิร์ฟเวอร์แบบฝังตัว โดยการแบ่งเฟรมสามารถทำได้โดยการใช้ HTML Tag ดังแสดงเป็นตัวอย่างในภาพที่ 43

```
< frameset rows="*,150">  
    <frame name="main" src="ledmod.html">  
    <frame name="sub" src="none.html">  
</frameset>
```

ภาพที่ 43 ตัวอย่างหน้าหลักเพื่อทำการแบ่งเฟรม