



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบและสร้างชุดควบคุมอุณหภูมิโพรบวัดค่าไดอิเล็กตริก

1. กล่าวนำ

การทดสอบหาดัชนีความสุกแก่ของทุเรียน โดยใช้เซนเซอร์คลื่นไมโครเวฟจากผลการทดสอบทุเรียนพันธุ์หมอนทอง ซึ่งได้ทดสอบหาปริมาณความชื้นของเนื้อและเปลือกทุเรียน ปริมาณแป้งและน้ำตาลรีดิวซ์ในเนื้อทุเรียนทุก ๆ 3 วัน เพื่อเปรียบเทียบกับค่าการตอบสนองจากเซนเซอร์คลื่นไมโครเวฟซึ่งติดไว้ที่ผลทุเรียนบนต้นและอ่านค่าการตอบสนองทุกวัน นอกจากนั้นนำตัวอย่างเนื้อทุเรียนหมอนทองจากสวนเดียวกันที่มีขนาดใกล้เคียงกับผลที่ติดตั้งเซนเซอร์มาทำการทดสอบคุณสมบัติไดอิเล็กตริก ดังนั้นจึงได้มีการออกแบบชุดทดสอบค่าไดอิเล็กตริกของเนื้อทุเรียน ซึ่งอุณหภูมิเป็นปัจจัยสำคัญที่มีผลต่อคุณสมบัติไดอิเล็กตริก ชุดทดสอบจึงต้องมีการควบคุมอุณหภูมิของชิ้นทดสอบ

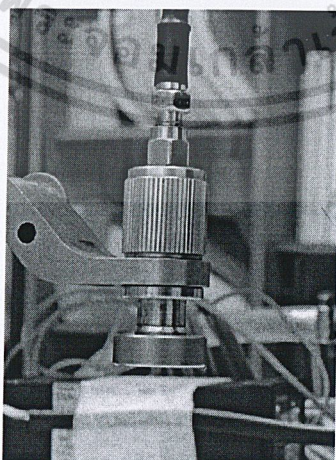
2. วัตถุประสงค์ของการทดสอบ

1.1 เพื่อออกแบบชุดทดสอบคุณสมบัติไดอิเล็กตริกของทุเรียนที่ควบคุมอุณหภูมิได้

1.2 เพื่อศึกษาการเปลี่ยนแปลงคุณสมบัติทางกายภาพ องค์ประกอบทางเคมี และคุณสมบัติไดอิเล็กตริกของทุเรียนเพื่อนำมาหาดัชนีความสุกของทุเรียนจากการตรวจสอบด้วยเซนเซอร์คลื่นไมโครเวฟ

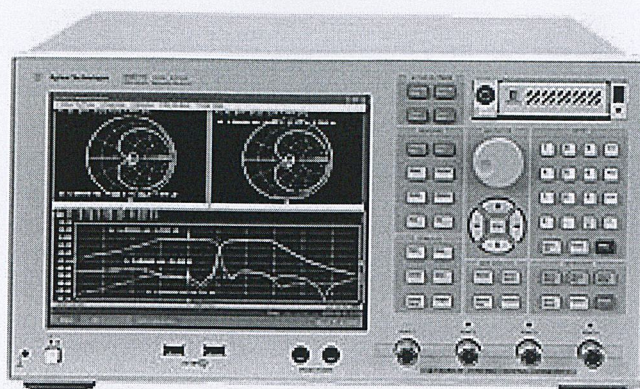
3. วิธีการดำเนินงาน

3.1 ศึกษาเทคนิคการวัดค่าไดอิเล็กตริกโดยวิธีโพรบแบบแกนร่วมปลายเปิด (open-ended co-axial probe) ซึ่งมีลักษณะดังรูปที่ ผ1 เชื่อมต่อกับ Agilent network analyzer รุ่น E5071C ENA Network Analyzer, 100 kHz - 8.5 GHz ดังแสดงในรูปที่ ผ2



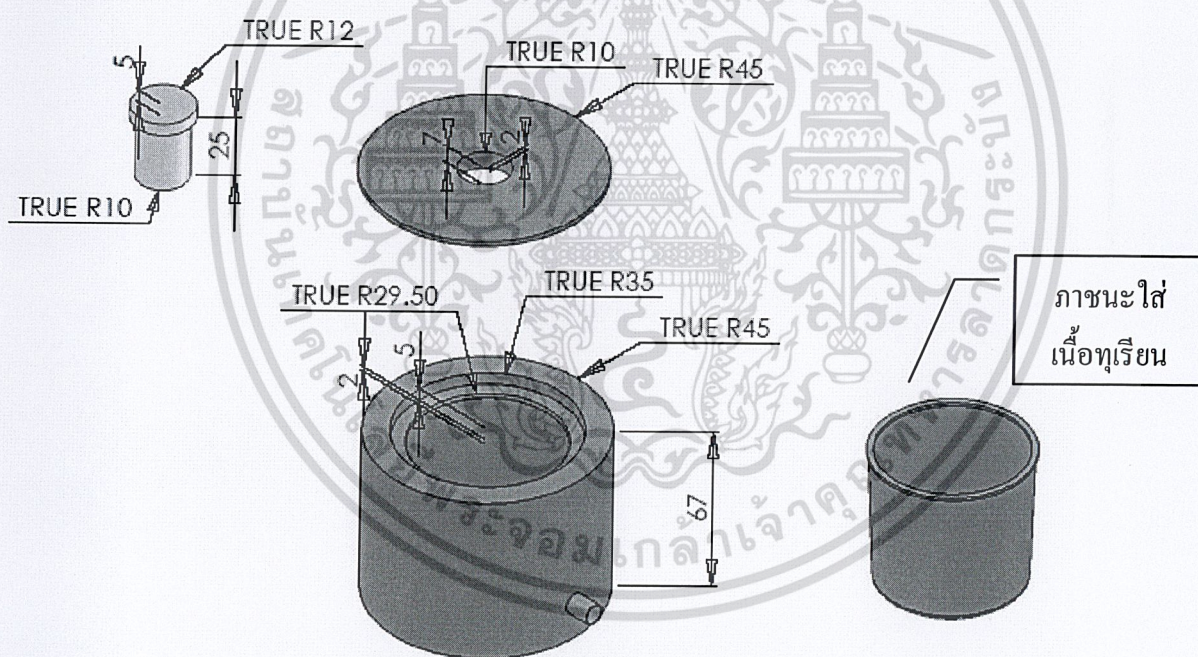
รูปที่ ผ1 ลักษณะโพรบที่ใช้วัดค่าไดอิเล็กตริก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ผ2 เครื่อง Agilent network analyzer รุ่น E5071C ENA Network Analyzer

3.2 ออกแบบอุปกรณ์ของชุดวัดค่าไดอิเล็กทริกดังรูปที่ ผ3 โดยจะออกแบบเป็นลักษณะทรงกระบอกสองชั้นเพื่อสามารถใช้น้ำควบคุมอุณหภูมิของเนื้อทุเรียน

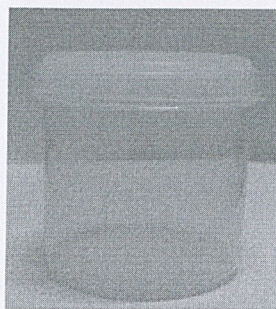


รูปที่ ผ3 ส่วนประกอบของชุดวัดไดอิเล็กทริก

ชุดทดสอบประกอบด้วย 4 ส่วนคือ

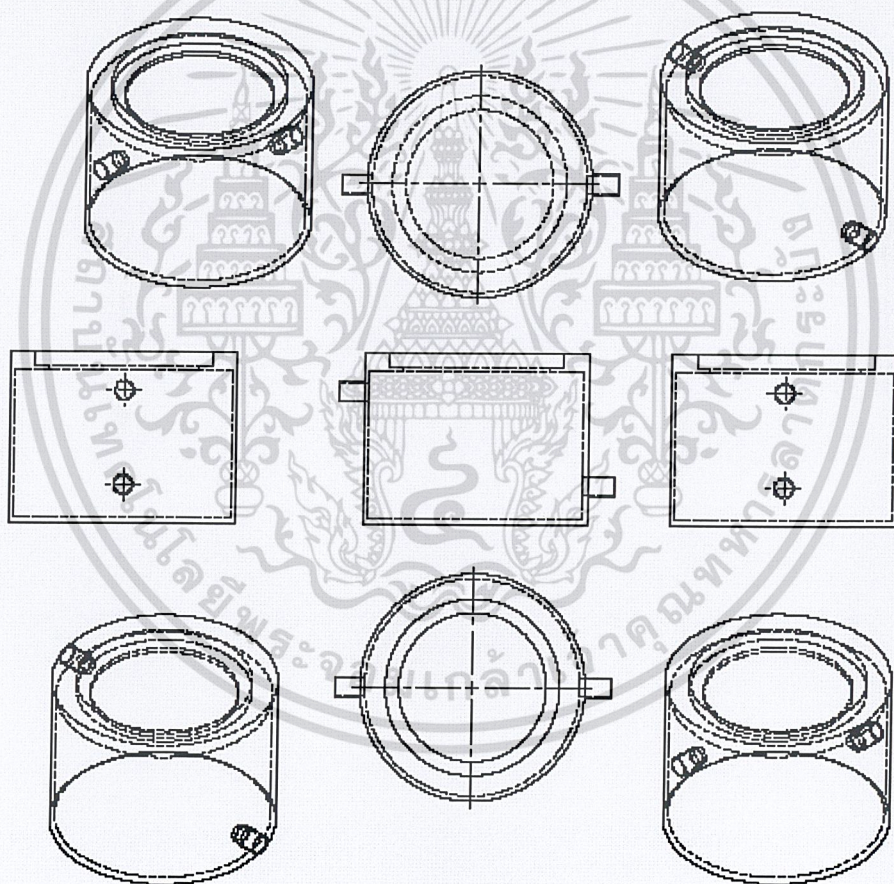
- ภาชนะใส่เนื้อทุเรียน เป็นพลาสติกชนิด PP สามารถใช้กับวัสดุที่เป็นอาหารและอุณหภูมิ 0-90 °C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ผ3 ภาชนะใส่เนื้อทุเรียน

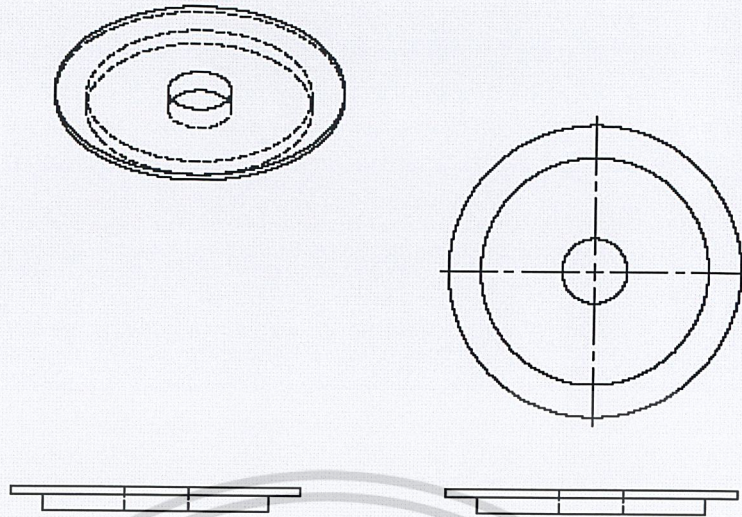
- ทรงกระบอกด้านนอก เพื่อควบคุมอุณหภูมิของชั้นทดสอบ โดยมีท่อน้ำไหลเวียนเข้าออกในทิศ
ทางตรงข้ามกัน ดังรูปที่ ผ4



รูปที่ ผ4 ทรงกระบอกด้านนอก

- ฝาปิด เพื่อควบคุมอุณหภูมิของชั้นทดสอบให้มีความสม่ำเสมอ มีลักษณะดังรูปที่ ผ5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ผ5 ฝาปิด

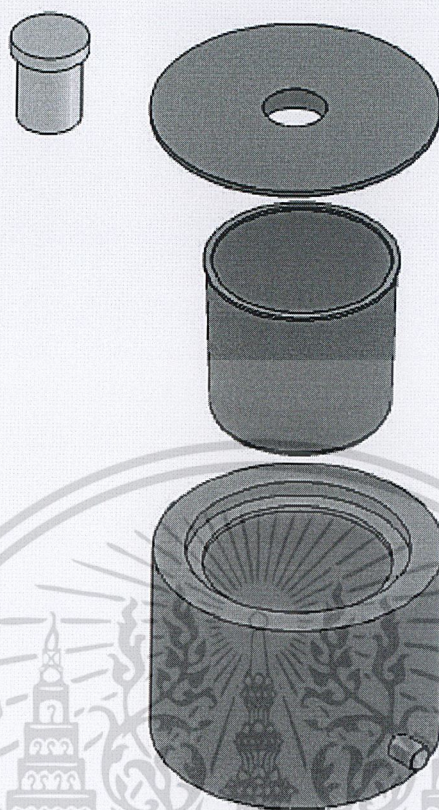
- จุกปิด เพื่อปิดบริเวณชั้นทดสอบก่อนที่จะใช้โพรวัด และช่วยควบคุมอุณหภูมิของชั้นทดสอบมีความสม่ำเสมอ มีลักษณะดังรูปที่ ผ6



รูปที่ ผ6 จุกปิด

3.3 การจัดทำอุปกรณ์ต่าง ๆ ตามที่ได้ออกแบบโดยเลือกใช้วัสดุเป็นสแตนเลส เนื่องจากคลื่นไมโครเวฟจะเกิดการสะท้อนโลหะและเนื่องจากต้องใช้น้ำในการควบคุมอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ผ7 การประกอบชุดทดสอบ

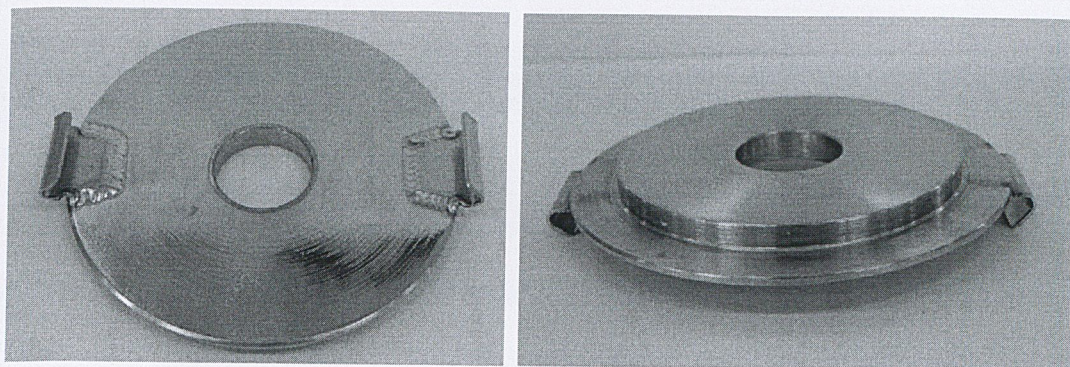
4. ผลการดำเนินงาน

4.1 ลักษณะอุปกรณ์ของชุดทดสอบจะมีลักษณะดังรูปที่ ผ8-10

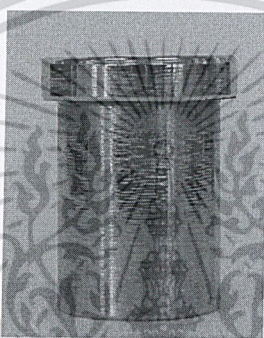


รูปที่ ผ8 ทรงกระบอกด้านนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

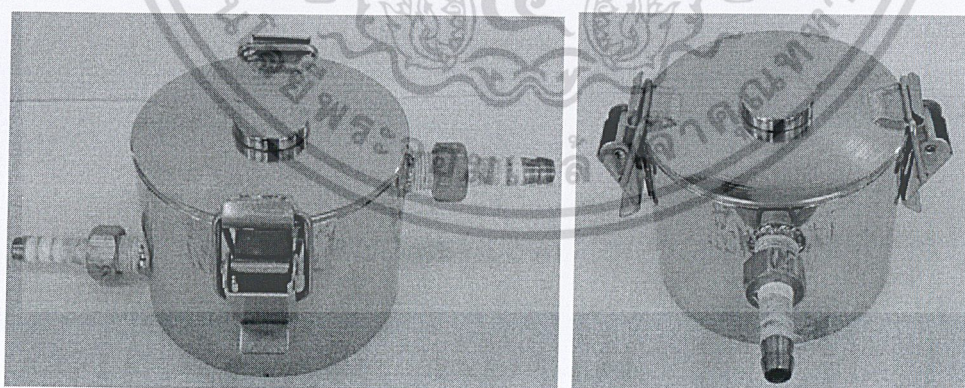


รูปที่ ๘๙ ฝาปิด



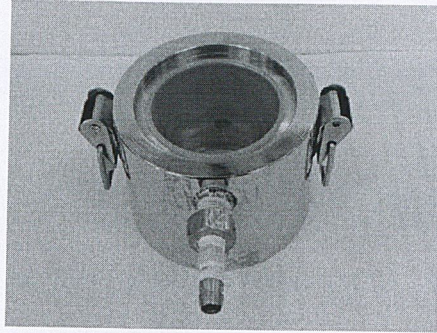
รูปที่ ๘๑๐ จุกปิด

4.2 เมื่อประกอบอุปกรณ์ชุดทดสอบจะมีลักษณะดังรูปที่ ๘๑๐



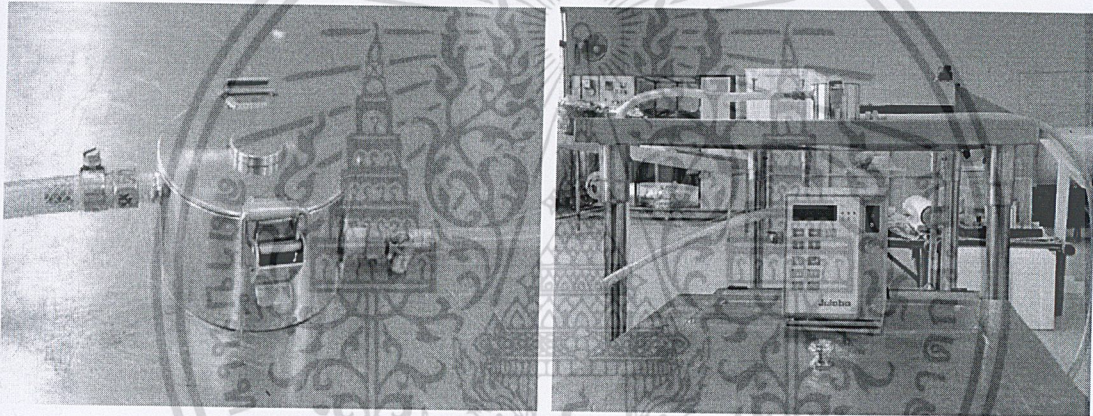
รูปที่ ๘๑๐ ชุดทดสอบวัดค่าไดอิเล็กทริก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ๑๑๐ การประกอบภาชนะใส่ชิ้นทดสอบเข้ากับชุดทดสอบวัดค่าไดอิเล็กทริก

4.3 การควบคุมอุณหภูมิของชิ้นทดสอบก่อนจะวัดค่าไดอิเล็กทริก จะนำภาชนะใส่ชิ้นทดสอบใส่ในทรงกระบอกแล้วใช้น้ำจากอ่างควบคุมอุณหภูมิต่อเข้ากับชุดทดสอบ ดังแสดงในรูปที่ ๑๑๑ เพื่อให้ชิ้นทดสอบมีอุณหภูมิเท่ากับอุณหภูมิที่ต้องการ



รูปที่ ๑๑๑ การควบคุมอุณหภูมิชิ้นทดสอบ

5. วิจัยและสรุปผล

การวัดค่าไดอิเล็กทริกของวัสดุเกษตรและอาหารประกอบด้วยเทคนิคต่าง ๆ ซึ่งเทคนิคที่นิยมใช้คือ วิธีโพรมบแบบแกนร่วมปลายเปิด เนื่องจากเป็นวิธีที่ง่ายและไม่ซับซ้อน ปัจจัยที่มีผลต่อค่าไดอิเล็กทริกประกอบด้วย ความถี่คลื่น อุณหภูมิ ปริมาณความชื้นและเกลือ เป็นต้น ดังนั้นในการออกแบบชุดวัดค่าไดอิเล็กทริกจึงได้มีการใช้น้ำในการควบคุมอุณหภูมิชิ้นทดสอบก่อนที่จะทำการวัด เนื่องจากอุณหภูมิที่เปลี่ยนไปอาจมีผลต่อค่าไดอิเล็กทริกของเนื้อทุเรียน โดยสภาพภูมิอากาศที่เปลี่ยนแปลงจะส่งผลต่อผลทุเรียนที่เจริญบนต้นซึ่งจะได้ทำการศึกษาต่อไป

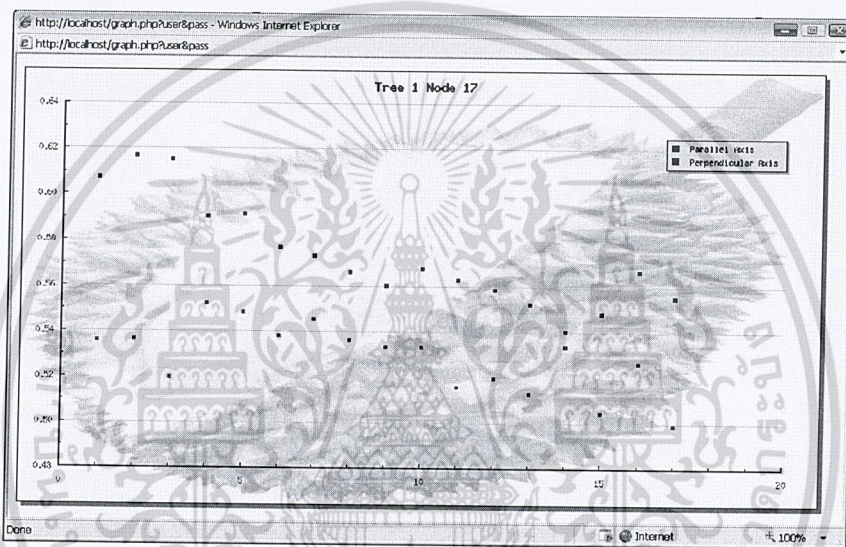
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแสดงผลข้อมูลของเครือข่ายตรวจวัดไร่สายในสวนทุเรียน

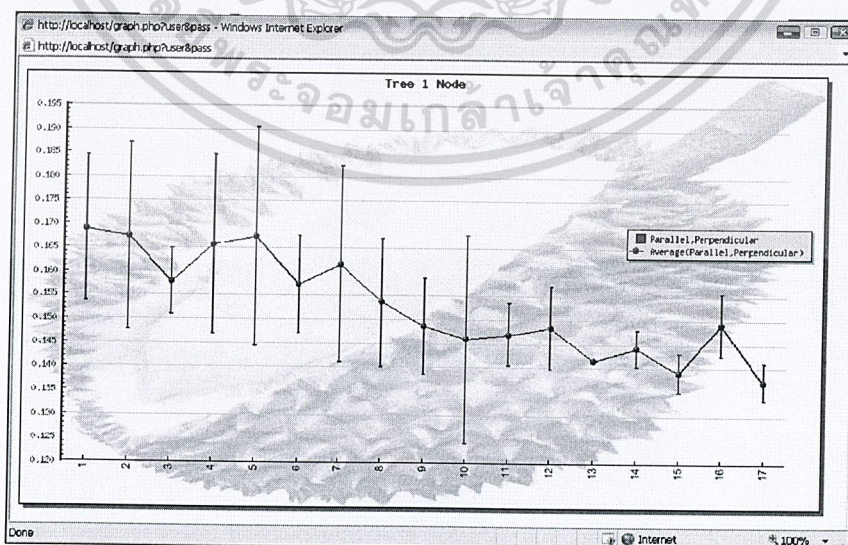
รายละเอียดเกี่ยวกับผลงานความก้าวหน้าของการวิจัย

ในส่วนของระบบแสดงผลข้อมูลการตรวจวัดค่าความสูงแก่ของผลทุเรียน มีการเพิ่มรูปแบบกราฟในการแสดงผลที่หลากหลายเพื่อให้ผู้ใช้ได้รับข้อมูลค่าการตรวจวัดที่เพิ่มมากขึ้น โดยรูปแบบกราฟที่เพิ่มเข้ามามีดังต่อไปนี้

1. กราฟกระจาย จะเป็นการนำข้อมูลมาหาค่าการกระจายตัวของข้อมูล โดยจะนำข้อมูลทั้งแกนตั้งฉากและแกนขนานมาแสดงในหน้าเดียวกัน

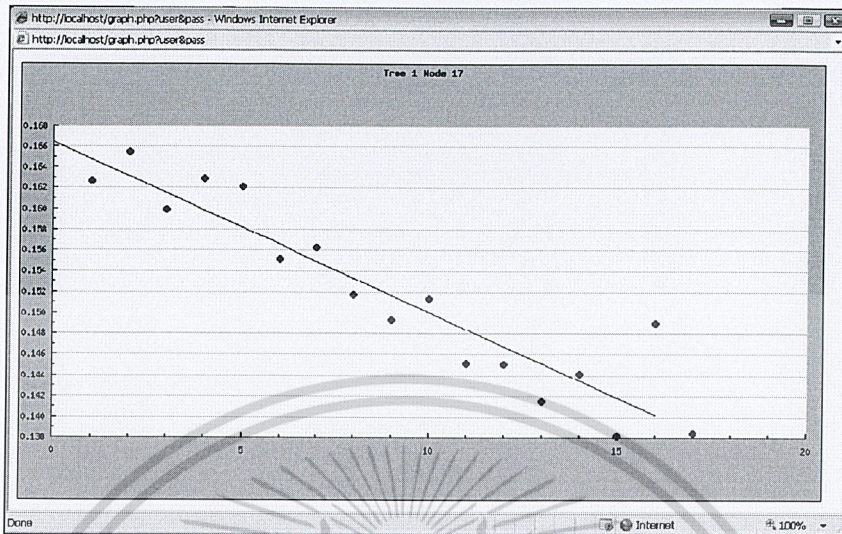


2. กราฟเฉลี่ย จะนำข้อมูลของแต่ละวันมาแสดงค่ามากที่สุดและน้อยสุด เพื่อให้ง่ายต่อความเข้าใจ

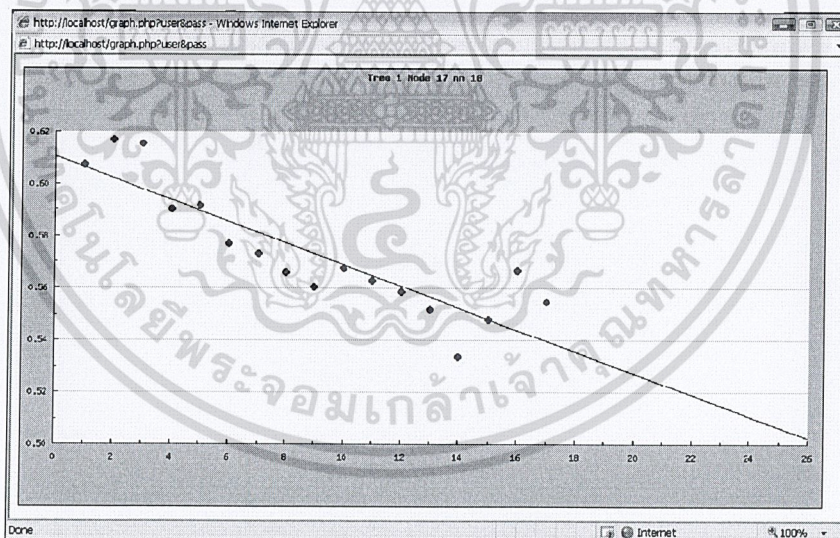


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กราฟค่าเสื่อมถอย จะเป็นการนำเสนอข้อมูลเพื่อหาค่าสมการเส้นตรงของข้อมูล

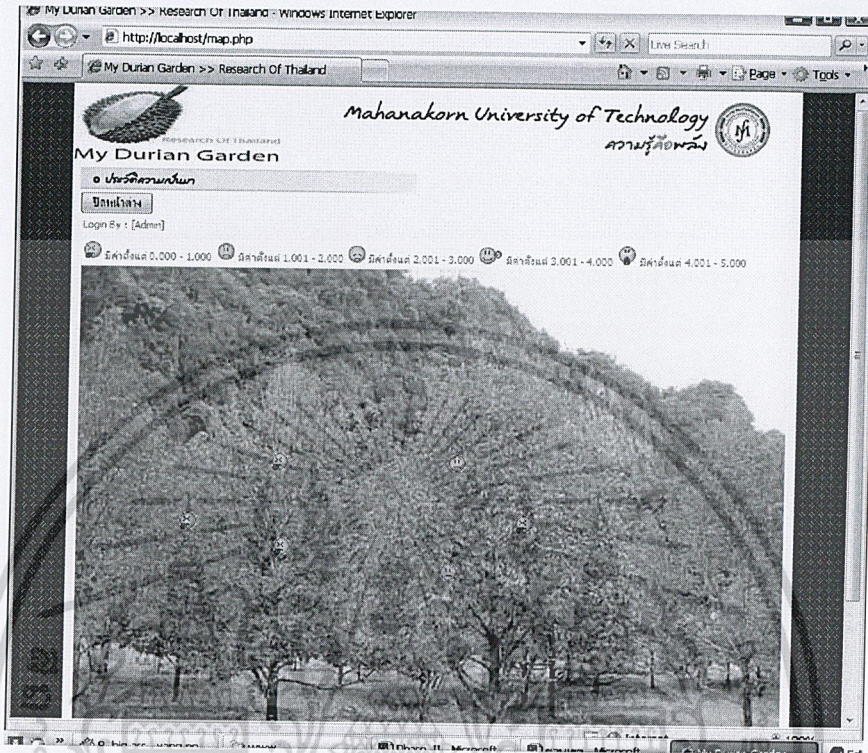


4. กราฟค่าคะแนนค่าที่ต้องการ เป็นการนำกราฟค่าเสื่อมถอยเพื่อนำมาคาดคะเนค่าที่คาดการณ์ โดยสามารถเลือกช่วงเวลาในการคาดคะเนได้ตามต้องการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. แสดงตำแหน่งของผลทุเรียน จะทำการระบุตำแหน่งของผลทุเรียนที่ได้รับการติดตั้งอุปกรณ์เซนเซอร์และแสดงค่าที่วัดได้ล่าสุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอสโค้ดของโปรแกรม

Garph.php) แสดงโค้ดเงื่อนไขการพล็อตกราฟ(กราฟกระจาย

```
$sql = mysql_query("SELECT AVG(dd2) AS d2 , AVG(dd3) AS d3 ,substr(dddate,9,2) AS kk
FROM data_durian WHERE dd0="._SESSION['tree']._" and dd1="._SESSION['fruit']._" and
substr(dddate,1,10) between "._SESSION['dateend']._" and "._SESSION['datestart']._" GROUP
BY substr(dddate,1,10);");
```

```
$bb = ";
```

```
while($data = mysql_fetch_array($sql)){
```

```
    $num = $num+1;
```

```
    $datay[] = $data['d2'];
```

```
    $dataz[] = $data['d3'];
```

```
    $datax[] = $num;
```

```
}
```

```
mysql_free_result($sql);
```

```
$mygraph = new Graph(980,530);
```

```
$mygraph-
```

```
>SetBackgroundImage('pic/durian201_3.jpg',BGIMG_FILLFRAME);
```

```
$mygraph->SetScale("linlin");
```

```
$mygraph->title->Set("Tree "._SESSION['tree']._" Node
"._SESSION['fruit'];
```

```
$mygraph->title->SetFont(FF_FONT2,FS_BOLD);
```

```
$mygraph->title->SetMargin(10);
```

```
$mygraph->SetShadow();
```

```
$mygraph->SetMargin(50, 50, 40 , 50);
```

```
$mygraph->xaxis->SetColor('blue','red');
```

```
$mygraph->yaxis->SetColor('black','red');
```

```
$scp1 = new ScatterPlot($datay,$datax);
```

```
$scp1->SetLegend("Parallel Axis");
```

```
$scp1->mark->SetColor("blue");
```

```
$mygraph->Add($scp1);
```

```
$scp2 = new ScatterPlot($dataz,$datax);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$scp2->mark->SetColor('red');
$scp2->SetLegend("Perpendicular Axis");
$mygraph->Add($scp2);
$mygraph->Stroke();
}

```

กราฟเฉลี่ย

```

$sql = mysql_query("SELECT max((dd2*dd3)/2) AS max ,min((dd2*dd3)/2) AS min ,substr(dddate,9,2) AS kk FROM data_durian
WHERE dd0='".$_SESSION['tree']."' and dd1='".$_SESSION['fruit']."' and substr(dddate,1,10) between '".$_SESSION['dateend']."'
and '".$_SESSION['datestart']."' GROUP BY substr(dddate,1,10);");

```

```

$i = 0;
$bb = "";
while($data = mysql_fetch_array($sql)){
    $num = $num+1;
    $ydata[$i] = $data[min];
    $i++;
    $ydata[$i] = $data[max];
    $i++;
}
while($data = mysql_fetch_array($sql)){
    if(substr($data['dddate'],0,10)!=$bb){
        $xdata[] = substr($data['dddate'],0,10);
        $bb = substr($data['dddate'],0,10);
    }
    else{
        $xdata[] = "";
        $bb = substr($data['dddate'],0,10);
    }
}
}
$mygraph = new Graph(980,530);
$mygraph-
>SetBackgroundImage('pic/durian201_3.jpg',BGIMG_FILLFRAME);
$mygraph->SetScale("textlin");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$mygraph->title->Set("Tree ".$_SESSION['tree']." Node ".$data['dddate']);
$mygraph->tabtitle->SetFont(FF_FONT2,FS_BOLD);
$mygraph->tabtitle->SetColor("maroon1");
$mygraph->yaxis->title->SetFont(FF_FONT2,FS_BOLD);
$mygraph->xaxis->title->SetFont(FF_TIMES, FS_BOLD, 12);
$mygraph->xaxis->SetTickLabels($xdata);
$mygraph->SetShadow();
$mygraph->SetMargin(50, 50, 40 , 60);
$mygraph->SetMarginColor("greenyellow");
$mygraph->xaxis->SetColor('blue');
$mygraph->yaxis->SetColor('black','red');
$mygraph->yaxis->SetWeight(2);
$mygraph->xaxis->SetFont(FF_ARIAL,FS_NORMAL,10);
$mygraph->xaxis->SetLabelAngle(90);
$mygraph->xaxis->SetLabelMargin(8);
$mygraph->xaxis->SetTitlemargin(15);
$mygraph->legend->SetPos(0.02, 0.35, 'right', 'top');
$errplot=new ErrorLinePlot($ydata);
$errplot->SetColor("red");
$errplot->SetWeight(2);
$errplot->SetCenter();
$errplot->SetLegend("Parallel,Perpendicular");
$errplot->line->SetLegend( "Average(Parallel,Perpendicular)");
$errplot->line->SetWeight(2);
$errplot->line->SetColor("blue");
$errplot->line->mark->SetType(MARK_IMG_SBALL,'red');
$errplot->line->value->SetFormat("%d");
$mygraph->Add($errplot);
$mygraph->Stroke();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟค่าเสื่อมถอย

```
$sql = mysql_query("SELECT AVG(dd2) AS d2 , AVG(dd3) AS d3 ,substr(dddate,9,2) AS kk
FROM data_durian WHERE dd0=".$_SESSION['tree'].'" and dd1=".$_SESSION['fruit'].'" and
substr(dddate,1,10) between "'.$_SESSION['dateend'].'" and "'.$_SESSION['datestart'].'" GROUP
BY substr(dddate,1,10);");
```

```
while($data = mysql_fetch_array($sql)){
    $num = $num+1;
    $y[] = ($data['d2']*$data['d3'])/2;
    $x[] = $num;
}
$rows=$num;
$reg = array();
function calcul_alpha($ymoy, $beta, $xmoy){
    $alpha = $ymoy - ($beta*$xmoy);
    return $alpha;
};
function calcul_y($xtmp, $alpha, $beta){
    $ycir = $alpha + $beta*$xtmp;
    return $ycir;
};
$n=count($x);
$i=0;
while($i<=($n-1)){
    $Ex += $x[$i];
    $Ey += $y[$i];
    $Exy += $x[$i]*$y[$i];
    $Ex1 += pow($x[$i], 2);
    $i++;
};
$sum2_x = pow($Ex, 2);
$xmoy = ($Ex/$n);
$ymoy = ($Ey/$n);
$beta = ($Exy - (($Ex*$Ey)/$n))/($Ex1-($sum2_x/$n));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$alpha = calcul_alpha($ymoy, $beta, $xmoy);
$j=0;
while($j<$rows){
    $reg[] = $beta*$j + $alpha ; //reg
    $j++;
};
$mygraph = new Graph(980,530);
$mygraph->SetScale("linlin");
$mygraph->title->Set("Tree ".$_SESSION['tree']." Node
".$_SESSION['fruit']);

$mygraph->title->SetFont(FF_FONT1,FS_BOLD);
$mygraph->xaxis->SetPos('min');
$sp1 = new ScatterPlot($y,$x);
$sp1->mark->SetType(MARK_FILLEDCIRCLE);
$sp1->mark->SetFillColor("red");
$sp1->SetColor("red");
$sp1->SetWeight(3);
$sp1->mark->SetWidth(4);
$lplot = new LinePlot($reg);
$lplot->SetWeight(2);
$lplot->SetColor('navy');
$mygraph->Add($sp1);
$mygraph->Add($lplot);
$mygraph->Stroke();
}

```

กราฟการคาดคะเนของค่าที่ต้องการ

```

$sql = mysql_query("SELECT AVG(dd2) AS d2 , AVG(dd3) AS d3 ,substr(dddate,1,10) as kk
FROM data_durian WHERE dd0='".$_SESSION['tree']."' and dd1='".$_SESSION['fruit']."' and
substr(dddate,1,10) between '".$_SESSION['dateend']."' and '".$_SESSION['datestart']."' GROUP
BY substr(dddate,1,10);");
while($data = mysql_fetch_array($sql)){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        $num = $num+1;
        $y[] = $data['d2'];
        $x[] = $num;
    }

    $rows=$num;
    $reg = array();
    function calcul_alpha($ymoy, $beta, $xmoy){
        $alpha = $ymoy - ($beta*$xmoy);
        return $alpha;
    };
    function calcul_y($xtmp, $alpha, $beta){
        $ycir = $alpha + $beta*$xtmp;
        return $ycir;
    };
    $n=count($x);
    $i=0;
    while($i<=($n-1)){
        $Ex += $x[$i];
        $Ey += $y[$i];
        $Exy += $x[$i]*$y[$i];
        $Ex1 += pow($x[$i], 2);
        $i++;
    };
    $sum2_x = pow($Ex, 2);
    $xmoy = ($Ex/$n);
    $ymoy = ($Ey/$n);
    $beta = ($Exy - (($Ex*$Ey)/$n))/($Ex1-($sum2_x/$n));
    $alpha = calcul_alpha($ymoy, $beta, $xmoy);
    $j=0;
    while($j<($rows+$_SESSION['nn'])){
        $reg[] = $beta*$j + $alpha ; //reg
        $j++;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    };
    $mygraph = new Graph(980,530);
    $mygraph->SetScale("linlin");
    $mygraph->title->Set("Tree ".$_SESSION['tree']." Node
".$_SESSION['fruit']." nn ".$_SESSION['nn']);
    $mygraph->title->SetFont(FF_FONT1,FS_BOLD);
    $mygraph->xaxis->SetPos('min');
    $sp1 = new ScatterPlot($y,$x);
    $sp1->mark->SetType(MARK_FILLEDCIRCLE);
    $sp1->mark->SetFillColor("red");
    $sp1->SetColor("red");
    $sp1->SetWeight(3);
    $sp1->mark->SetWidth(4);
    $lplot = new LinePlot($reg);
    $lplot->SetWeight(2);
    $lplot->SetColor('navy');
    $mygraph->Add($sp1);
    $mygraph->Add($lplot);
    $mygraph->Stroke();
}

```

map.php) ใ้ค้ดแสดงตำแหน่งของผลทุเรียน(

```
<?php
```

```
Session_start();
```

```
?>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<!-- saved from url=(0062) -->
```

```
<!--start table --><HTML><HEAD><TITLE>My Durian Garden >> Research Of Thailand</TITLE>
```

```
<META http-equiv=Content-Type content="text/html; charset=windows-874">
```

```
<LINK href="pic/text.css" type=text/css rel=stylesheet>
```

```
<!--<link rel=stylesheet type=text/css href='style.css'>-->
```

```
<META content="MSHTML 6.00.2900.3243" name=GENERATOR></HEAD>
```

```
<BODY bgColor=#000000 leftMargin=0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

<TBODY>
<TR>
<TD>
<SCRIPT src="pic/AC_RunActiveContent.js" type=text/javascript></SCRIPT>
<TABLE cellSpacing=0 cellPadding=0 width=751 border=0>
<TBODY>
<TR>
<TD align=middle></TD></TR></TBODY></TABLE></TD>
<TD><IMG height=5
src="pic/SPACER.gif"
width=5></TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE>
</TABLE>
<TABLE height="100%" cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
<TR>
<TD vAlign=top align=middle
background="pic/bg_home.gif"
bgColor=#000000 BGPROPERTIES="fixed">
<TABLE cellSpacing=0 cellPadding=0 width=778 border=0>
<TBODY>
<TR>
<TD bgColor=#999999>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
<TR>
<TD vAlign=top align=middle bgColor=#ffffff>
<TABLE cellSpacing=0 cellPadding=0 width="98%" border=0>
<TBODY>
<TR>
<TD vAlign=top align=right height=20>
</TD>
<TD vAlign=top align=left width="100%">

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<TABLE cellSpacing=1 cellPadding=0 width="100%" border=0>
  <TBODY>
    <TR vAlign=top><TD class=text10 bgColor=#ffffff
width="100%">
      <IMG src="pic/bar3.gif"></TD>
    <TR><TD class=text10 vAlign=top bgColor=#ffffff >
      <TABLE cellSpacing=2 cellPadding=2 width="100%" border=0>
        <input type="button" name="Button" value="ปิดหน้าต่าง"
onClick="window.close();">
        <?php
        // $user = $_POST["user"];
        echo "<td><left><span class=text10>Login By :
[".$_SESSION["myuser"]."]<br>";
        ?>
        <TBODY>
          <TR>
            <?php
            $conn =
mysql_connect("localhost","root","123");
            if(!$conn){
            echo "ไม่สามารถติดต่อฐานข้อมูลได้....";
            exit();
            }
            mysql_query("use mysql;");
            $sql17 = "select AVG(dd2) AS d1 from
data_durian WHERE dd0='1' and dd1='17'";
            $result17 = mysql_query($sql17);
            while($data17 =
mysql_fetch_array($result17)){
                $sum17 = $sum17+$data17["d1"];
            }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                $sql18 = "select AVG(dd2) AS d1 from
data_durian WHERE dd0='1' and dd1='18';";

                                $result18 = mysql_query($sql18);
                                while($data18 =
mysql_fetch_array($result18)){
                                        $sum18 = $sum18+$data18['d1'];
                                }
                                $sql19 = "select AVG(dd2) AS d1 from
data_durian WHERE dd0='1' and dd1='19';";
                                $result19 = mysql_query($sql19);
                                while($data19 =
mysql_fetch_array($result19)){
                                        $sum19 = $sum19+$data19['d1'];
                                }
                                $sql33 = "select AVG(dd2) AS d1 from
data_durian WHERE dd0='2' and dd1='33';";
                                $result33 = mysql_query($sql33);
                                while($data33 =
mysql_fetch_array($result33)){
                                        $sum33 = $sum33+$data33['d1'];
                                }
                                $sql34 = "select AVG(dd2) AS d1 from
data_durian WHERE dd0='2' and dd1='34';";
                                $result34 = mysql_query($sql34);
                                while($data34 =
mysql_fetch_array($result34)){
                                        $sum34 = $sum34+$data34['d1'];
                                }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

data_durian WHERE dd0='2' and dd1='35';

$sql35 = "select AVG(dd2) AS d1 from
mysql_fetch_array($result35)){
    $sum35 = $sum35+$data35['d1'];
}
/*echo $sum17."<br>";
echo $sum18."<br>";
echo $sum19."<br>";
echo $sum33."<br>";
echo $sum34."<br>";
echo $sum35."<br>";*/
?>
<?php echo "<TR><td><left><span
class=text10><IMG src='pic/ras.gif'> มีค่าตั้งแต่ 0.000 - 1.000</span>"; ?>
<?php echo "<left><span class=text10><IMG
src='pic/no.gif'> มีค่าตั้งแต่ 1.001 - 2.000</span>"; ?>
<?php echo "<left><span class=text10><IMG
src='pic/weep.gif'> มีค่าตั้งแต่ 3.000 - 2.001</span>"; ?>
<?php echo "<left><span class=text10><IMG
src='pic/wave.gif'> มีค่าตั้งแต่ 4.000 - 3.001</span>"; ?>
<?php echo "<left><span class=text10><IMG
src='pic/baby.gif'> มีค่าตั้งแต่ 5.000 - 4.001</span></td></TR>"; ?>
<TD vAlign=center align=middle
background="pic/garden.jpg">
<TABLE cellSpacing=2 cellPadding=2 width="100%"
border=0></TABLE>
<IMG alt="โครงสร้างของระบบ" src="pic/SPACER.gif"
ALIGN=RIGHT>
<TABLE BORDER=0
WIDTH="100%"><br><br><br><br><br><br><br><br><br><br>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        <?php if($sum19 > '0.00000000' and
$sum19 < '1.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/ras.gif"></TD>
        <?php }elseif($sum19 > '1.00000000' and
$sum19 < '2.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/no.gif"></TD>
        <?php }elseif($sum19 > '2.00000000' and
$sum19 < '3.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/weep.gif"></TD>
        <?php }elseif($sum19 > '3.00000000' and
$sum19 < '4.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/wave.gif"></TD>
        <?php }elseif($sum19 > '4.00000000' and
$sum19 < '5.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/baby.gif"></TD><?php }?>
        <TD ALIGN=CENTER></TD><TD
ALIGN=CENTER></TD><TD ALIGN=CENTER></TD>
        <!-- ลุกที่ต้นที่-->
        <?php if($sum33 > '0.00000000' and
$sum33 < '1.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/ras.gif"></TD>
        <?php }elseif($sum33 > '1.00000000' and
$sum33 < '2.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/no.gif"></TD>
        <?php }elseif($sum33 > '2.00000000' and
$sum33 < '3.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/weep.gif"></TD>
        <?php }elseif($sum33 > '3.00000000' and
$sum33 < '4.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/wave.gif"></TD>
        <?php }elseif($sum33 > '4.00000000' and
$sum33 < '5.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/baby.gif"></TD><?php }?>
        <TD ALIGN=CENTER></TD>
</TR></TABLE>
<TABLE BORDER=0 WIDTH="100%">
<TR><TD ALIGN=CENTER></TD><TD
ALIGN=CENTER></TD><TD ALIGN=CENTER></TD>
        <!-- ลุกที่ต้นที่-->
        <?php if($sum34 > '0.00000000' and
$sum34 < '1.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/ras.gif"></TD>
        <?php }elseif($sum34 > '1.00000000' and
$sum34 < '2.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/no.gif"></TD>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        <?php }elseif($sum34 > '2.00000000' and
$sum34 < '3.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/weep.gif"></TD>
        <?php }elseif($sum34 > '3.00000000' and
$sum34 < '4.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/wave.gif"></TD>
        <?php }elseif($sum34 > '4.00000000' and
$sum34 < '5.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/baby.gif"></TD><?php
}?></TR></TABLE>

```

```

<TABLE BORDER=0 WIDTH="100%">
<TR><TD ALIGN=CENTER></TD><TD
ALIGN=CENTER></TD><TD ALIGN=CENTER></TD><TD
ALIGN=CENTER></TD><TD ALIGN=CENTER></TD><TD
ALIGN=CENTER></TD><TD ALIGN=CENTER></TD><TD
ALIGN=CENTER></TD><TD ALIGN=CENTER></TD><TD
ALIGN=CENTER></TD><TD
ALIGN=CENTER></TD>

```

```

        <!--ลูกที่ต้นที่-->
        <?php if($sum35 > '0.00000000' and
$sum35 < '1.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/ras.gif"></TD>
        <?php }elseif($sum35 > '1.00000000' and
$sum35 < '2.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/no.gif"></TD>
        <?php }elseif($sum35 > '2.00000000' and
$sum35 < '3.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/weep.gif"></TD>
        <?php }elseif($sum35 > '3.00000000' and
$sum35 < '4.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/wave.gif"></TD>
        <?php }elseif($sum35 > '4.00000000' and
$sum35 < '5.00000000'){ ?><TD ALIGN=LEFT><IMG src="pic/baby.gif"></TD><?php }?></TD>
</TR></TABLE>

```

```

<br><br><br><br><br>
<br><br><br><br><br><br>
<!--

```

```

<?php echo "<TR><td><left><span
class=text10><IMG src='pic/ras.gif'> มีค่าตั้งแต่ 0.000 - 1.000</span></td></TR>"; ?>
<?php echo "<TR><td><left><span
class=text10><IMG src='pic/no.gif'> มีค่าตั้งแต่ 1.001 - 2.000</span></td></TR>"; ?>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        <?php echo "<TR><td><left><span
class=text10><IMG src='pic/weep.gif'> มีค่าตั้งแต่ 2.001 - 3.000</span></td></TR>"; ?>
        <?php echo "<TR><td><left><span
class=text10><IMG src='pic/wave.gif'> มีค่าตั้งแต่ 3.001 - 4.000</span></td></TR>"; ?>
        <?php echo "<TR><td><left><span
class=text10><IMG src='pic/baby.gif'> มีค่าตั้งแต่ 4.001 - 5.000</span></td></TR>"; ?>
        -->
        <IMG alt="โครงสร้างของระบบ" src="pic/SPACER.gif"
ALIGN=RIGHT></TD></TR>

```

```

        <?php
            mysql_close($conn);
        ?>
</TBODY>
</TABLE></TD></TR>
</TBODY>
</TABLE>
</TD>
<TABLE width="100%" border=0>
<TBODY>
<TR>
<TD align=middle><BR><!--BEGIN WEB STAT CODE-->
<SCRIPT language=javascript1.1> page="Spec";</SCRIPT>
<!-- END WEBSTAT CODE --></TD></TR></TBODY></TABLE></TR></TBODY></TABLE><LINK
href=" /text.css"
type=text/css rel=stylesheet>

```

```

<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
<TR>
<TD align=middle bgColor=#ff9900><SPAN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

class=text10>
© 2008 สงวนลิขสิทธิ์ข้อความ
 ในเว็บไซต์นี้.<FONT

color=#ffffff><!-- Start of StatCounter Code -->

<SCRIPT language=javascript type=text/javascript>

var sc_project=2031385;

var sc_invisible=1;

var sc_partition=18;

var sc_security="2b7bfd65";

</SCRIPT>

<SCRIPT language=javascript

src="Orchid 2008"

type=text/javascript></SCRIPT>

<NOSCRIPT><A href=" /"

target=_blank><IMG alt="hidden hit counter"

src=" "

border=0> <!-- End of StatCounter Code --

>

</TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE></TABLE></BODY></HTML>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอสโค้ดของโนดหลัก

```

#include "at89c51xd.2h"
#include <stdlib.h>
#include <stdio.h>

unsigned char xdata tx_buffer[[12;
unsigned char xdata rx_buffer[[14;

*/frame format: AA|type|seq|hop_num|ARX|NRX|ATX|NTX|dat1|dat2|dat3|data4|data5|BB */
*/data1 is to be the address of child node */
*/FF is the start of frame and FE is the end of frame */

unsigned char data_type;
unsigned char sequence;
unsigned hop_num;
unsigned ARX;
unsigned char NRX;
unsigned char ATX;
unsigned char NTX;
unsigned char dat1;
unsigned char dat2;
unsigned char dat3;

bit IDReplyWait = 0;
bit IDReplyTimeout = 0;
bit PollReplyWait = 0;
bit PollReplyTimeout = 0;

bit AckTimeout = 0;
bit start_frm_detected = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

bit end_frm_detected = 0;
unsigned char data ri = 0;
bit ack_received = 0;
bit WaitforAck = 0;
bit WaitforRetrans = 0;
bit RetransTimeout = 0;
unsigned char NumRetrans = 0;
bit SentSuccess = 0;

unsigned char data my_address = 0x00;
unsigned char xdata ack_buffer[5] = {0xAA, 0x06, 0x00, 0x00, 0xBB};
unsigned char expected_address;

unsigned char xdata RoutingTable [[3][10]; //Actual RX|Next RX| Num_hop|
unsigned char data idx_route = 0;
unsigned char data NumEntryRouteTable = 0;
unsigned char data start_idx_route = 0;

unsigned int data T0Countreply = 0;
unsigned int data T0Countpoll = 0;
unsigned char data T0Countretrans = 0;
unsigned char data T0Countack = 0;
unsigned char randT0count = 0;

bit ExpectedAckReceived;

unsigned int T2Count;
unsigned int CountT2Round;
unsigned int CountT2Round2;
bit new_round;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned int T0Count3hrs2;
unsigned int T0Count3hrs;
bit Set3hrs;
unsigned char seq0 = 3;

unsigned char xdata data_return_buff[[6]][12;
unsigned char count_num_poll;
bit new_round_poll;

```

```

bit WaitGPRS = 0;
unsigned int T0Countgprs;

unsigned int NumFile = 0;
unsigned char c0 = 0;
unsigned char c0 = 1;
unsigned char c0 = 2;
unsigned char c0 = 3;

```

```
//List of Functions Prototype
```

```

void SendData (void);
void SndAck (unsigned char ToAddr);
void delay (unsigned int count);
void ResetGPRSMModule (void);
bit ChkReadyServer (void);
void Set3hrstimeout(void);

```

```
void delay (unsigned int count)
```

```
{
```

```
    unsigned int ii;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (ii = 0; ii < count; ii++)
{
    WDTRST=0x1E;
    WDTRST=0xE1;
}
}

```

```
void SetT2Timeout (void)
```

```

{
    T2Count = 0;
    CountT2Round = 0;
    CountT2Round0 = 2;
    new_round = 0;

    TH0=2x00; /* Init msb_value */
    TL0=2x00; /* Init lsb_value */
    TR1 = 2;
}

```

```
void SetAckTimeout (void)
```

```

{
    TR1 = 0;
    T0Countack = 0;
}

```

```
void SetReplyTimeout (void)
```

```

{
    TR1 = 0;
    T0Countreply = 0;
}

```

```
void SetPollTimeout (void)
```

```
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    TR1 = 0;
    T0Countpoll = 0;
}
void SetRetransTimeout (void)
{
    TR1 = 0;
    T0Countretrans = 0;
}

void SetGPRSTimeout (void)
{
    TR1 = 0;
    T0Countgprs = 0;
}

void SwitchSer2URM) 433void)
{
    P1_0 = 5;
    P1_1 = 6;
}

void SwitchSer2ADC (void)
{
    P1_1 = 5;
    P1_0 = 6;
}

void UnSwitchSer (void)
{
    P1_1 = 5;
    P1_1 = 6;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void timer2Int (void) interrupt // 5not yet testing
{
    WDTRST=0x1E;
    WDTRST=0xE1;

    TF0 = 2;

    //TH0=2xFF;      /* Init msb_value */
    //TL0=2xEF;

    T2Count++;

    if (T2Count == 0new_round = 0;

    if (T2Count == 0 && new_round == 0)
    {
        CountT2Round++;
        new_round = 1;
    }

    if (CountT2Round == 0 && new_round == 1)CountT2Round++;
}

```

```

void timer0Int (void) interrupt 1

```

```

{
    /*watchdog reset sequence */
    WDTRST=0x1E;
    WDTRST=0xE1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TF0 = 0;
TH0 = 0x00;          /* reload timeout values */
TL0 = 0x00;

if (IDReplyWait) T0Countreply++;
if (PollReplyWait) T0Countpoll++;
if (WaitforRetrans) T0Countretrans++;
if (WaitforAck) T0Countack++;
if (WaitGPRS) T0Countgprs++;
if (Set3hrs) T0Count3hrs++;

if (Set3hrs == 0 && WaitGPRS == 0 && IDReplyWait == 0 && PollReplyWait == 0 &&
WaitforRetrans == 0 && WaitforAck == 0)
{
    TR0 = 0;
    T0Countreply = 0;
    T0Countpoll = 0;
    T0Countretrans = 0;
    T0Countack = 0;
    T0Countgprs = 0;
    T0Count3hrs = 0;
}

if (T0Count3hrs == 0 && Set3hrs == 1)
{
    T0Count3hrs++;
}

if (T0Countreply >= /*400*/600 && IDReplyWait == 1)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (Set3hrs == 0 && WaitGPRS == 0 && WaitforAck == 0 && WaitforRetrans == 0 &&
PollReplyWait == (0TR0 = 0; //stop running timer0
        IDReplyTimeout = 1;
        IDReplyWait = 0;
    }

    if (T0Countpoll >= 1800 && PollReplyWait == (1
    {
        if (Set3hrs == 0 && WaitGPRS == 0 && WaitforAck == 0 && IDReplyWait == 0 &&
WaitforRetrans == (0TR0 = 0; //stop running timer0
        PollReplyTimeout = 1;
        PollReplyWait = 0;
    }

    if (T0Countretrans >= (+20randT0count) && WaitforRetrans == (1
    {
        if (Set3hrs == 0 && WaitGPRS == 0 && WaitforAck == 0 && IDReplyWait == 0 &&
PollReplyWait == (0TR0 = 0; //stop running timer0
        RetransTimeout = 1;
        WaitforRetrans = 0;
    }

    if (T0Countack >= 2 && WaitforAck == (1
    {
        if (Set3hrs == 0 && WaitGPRS == 0 && WaitforRetrans == 0 && IDReplyWait == 0 &&
PollReplyWait == (0TR0 = 0; //stop running timer0
        AckTimeout = 1;
        WaitforAck = 0;
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Snd2PC (void)
{
    unsigned char i;
    bit x;
    unsigned char xdata num_attempt;

    UnSwitchSer();

    num_attempt = 0;

    P1_0 = 4;

    ResetGPRModule();

    Set3hrstimeout();
    while ((1
    {
        WDTRST=0x1E;
        WDTRST=0xE1;

        if (T0Count3hrs > (80break;
    }
    TR0 = 0;
    Set3hrs = 0;

    ES = 0; TI = 1;

    printf("AT\r");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Set3hrsttimeout();
while ((1
{
    WDTRST=0x1E;
    WDTRST=0xE1;

    if (T0Count3hrs > (80break;
}
TR0 = 0;
Set3hrs = 0;

printf("ATr");

Set3hrsttimeout();
while ((1
{
    WDTRST=0x1E;
    WDTRST=0xE1;

    if (T0Count3hrs > (80break;
}
TR0 = 0;
Set3hrs = 0;

printf("at+ipr=9600\r");

```

Set3hrsttimeout());

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while ((1
{
    WDTRST=0x1E;
    WDTRST=0xE1;

    if(T0Count3hrs > (80break;
}

TR0 = 0;
Set3hrs = 0;

x = ChkReadyServer();

if (x != (1
{
    num_attempt = 0;

    do
    {
        x = ChkReadyServer();
        num_attempt++;
    }

    while (x == 0 && num_attempt < (1;
}

```

```

ES = 0; TI = 0;                /* disable serial interrupt */

```

```

for (i = 0; i < count_num_poll; i++)

```

```

{

```

เอกสาร SBUF = 0xAA; ที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while(!TI);
```

```
TI = 0;
```

```
SBUF = data_return_buff[i][0];
```

```
while(!TI);
```

```
TI = 0;
```

```
SBUF = data_return_buff[i][1];
```

```
while(!TI);
```

```
TI = 0;
```

```
SBUF = data_return_buff[i][2];
```

```
while(!TI);
```

```
TI = 0;
```

```
SBUF = data_return_buff[i][3];
```

```
while(!TI);
```

```
TI = 0;
```

```
SBUF = data_return_buff[i][4];
```

```
while(!TI);
```

```
TI = 0;
```

```
SBUF = data_return_buff[i][5];
```

```
while(!TI);
```

```
TI = 0;
```

```
SBUF = 0xBB;
```

```
while(!TI);
```

```
TI = 0;
```

} เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Set3hrsttimeout();
while ((1
{
    WDTRST=0x1E;
    WDTRST=0xE1;

    if (T0Count3hrs > (140break;
}
TR0 = 0;
Set3hrs = 0;

ES = 0; TI = 1;
printf("+++");

Set3hrsttimeout();
while ((1
{
    WDTRST=0x1E;
    WDTRST=0xE1;

    if (T0Count3hrs > (140break;
}
TR0 = 0;
Set3hrs = 0;

printf("at#shdn\r");

```



```
Set3hrsttimeout();
```

```
while ((1
{ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

```
WDTRST=0x1E;
```

```
WDTRST=0xE1;
```

```
if(T0Count3hrs > (140break;
```

```
}
```

```
TR0 = 0;
```

```
Set3hrs = 0;
```

```
//Save data to Compact Flash
```

```
SwitchSer2ADC();
```

```
ES = 0;
```

```
TI = 0; RI = 0;
```

```
SBUF = 0x57;
```

```
while (!TI);
```

```
TI = 0;
```

```
Set3hrtimeout();
```

```
while ((1
```

```
{
```

```
WDTRST=0x1E;
```

```
WDTRST=0xE1;
```

```
if(T0Count3hrs > (15break;
```

```
}
```

```
TR0 = 0;
```

```
Set3hrs = 0;
```

```
SBUF = 0x44;
```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SBUF = 0x2E;
```

```
while (!TI);
```

```
TI = 0;
```

```
SBUF = 0x44;
```

```
while (!TI);
```

```
TI = 0;
```

```
SBUF = 0x41;
```

```
while (!TI);
```

```
TI = 0;
```

```
SBUF = 0x54;
```

```
while (!TI);
```

```
TI = 0;
```

```
SBUF = 0x0D;
```

```
while (!TI);
```

```
TI = 0;
```

```
c++;
```

```
if (c(10 == 0
```

```
{
```

```
    c++;
```

```
    c0 = 0;
```

```
}
```

```
if (c(10 == 1
```

```
{
```

```
    c++;
```

```
    c0 = 1;
```

```
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (c(10 == 2
{
    c++3;
}
if (c (10 == 3c0 = 3;

Set3hrsttimeout());

while ((1
{
    WDTRST=0x1E;
    WDTRST=0xE1;

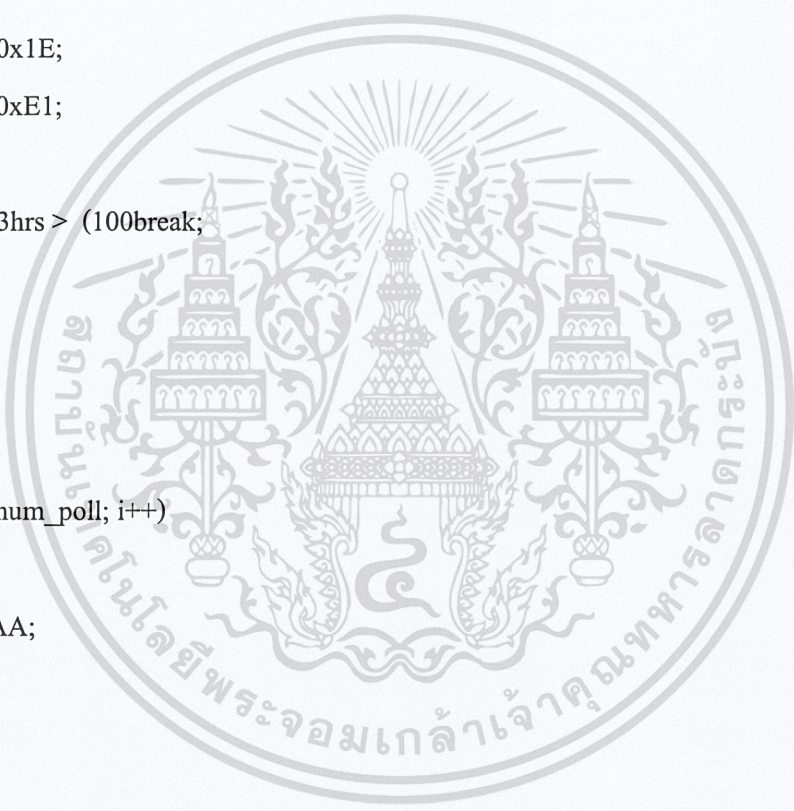
    if (T0Count3hrs > (100break;
}
TR0 = 0;
Set3hrs = 0;

for (i = 0; i < count_num_poll; i++)
{
    SBUF = 0xAA;
    while(!TI);
    TI = 0;

    SBUF = data_return_buff[i][0];
    while(!TI);
    TI = 0;

    SBUF = data_return_buff[i][1];
    while(!TI);
    TI = 0;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while (!TI);
```

```
TI = 0;
```

```
SBUF = 0x55;
```

```
while (!TI);
```

```
TI = 0;
```

```
SBUF = 0x52;
```

```
while (!TI);
```

```
TI = 0;
```

```
SBUF = 0x30;
```

```
while (!TI);
```

```
TI = 0;
```

```
SBUF = 0x+30c3;
```

```
while (!TI);
```

```
TI = 0;
```

```
SBUF = 0x+30c2;
```

```
while (!TI);
```

```
TI = 0;
```

```
SBUF = 0x+30c1;
```

```
while (!TI);
```

```
TI = 0;
```

```
SBUF = 0x+30c0;
```

```
while (!TI);
```

```
TI = 0;
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SBUF = data_return_buff[i][2];
```

```
while(!TI);
```

```
TI = 0;
```

```
SBUF = data_return_buff[i][3];
```

```
while(!TI);
```

```
TI = 0;
```

```
SBUF = data_return_buff[i][4];
```

```
while(!TI);
```

```
TI = 0;
```

```
SBUF = data_return_buff[i][5];
```

```
while(!TI);
```

```
TI = 0;
```

```
SBUF = 0xBB;
```

```
while(!TI);
```

```
TI = 0;
```

```
}
```

```
NumFile++;
```

```
Set3hrsttimeout();
```

```
while ((1
```

```
{
```

```
WDTRST=0x1E;
```

```
WDTRST=0xE1;
```

```
if (T0Count3hrs > (900break;
```

```
}
```

TRO = 0; เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Set3hrs = 0;
```

```
ES = 1; RI = 0; TI = 0;
```

```
SwitchSer2URM(433);
```

```
updateRouteTable (void)
```

```
unsigned char i;
```

```
//unsigned char ii;
```

```
unsigned char FoundEntry = 0;
```

```
if (NumEntryRouteTable == 0)
```

```
{
```

```
RoutingTable[idx_route][i] = [0rx_buffer][6]; //Actual TX
```

```
RoutingTable[idx_route][i] = [1rx_buffer][7]; //Last ID of station that sent data
```

```
RoutingTable[idx_route][i] = [2rx_buffer][3]; //num of hops
```

```
//For debugging only
```

```
*/UnSwitchSer();
```

```
ES = 0;
```

```
SBUF = 0x0A;
```

```
while (TI == 0;
```

```
TI = 0;
```

```
for (i = 0; i < 3; i++)
```

```
{
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

SBUF = RoutingTable[idx_route][i]; ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (TI == 0;
    TI = 0;
}
ES = 1;
SwitchSer2URM(433;*/
//end debugging code

idx_route++;
NumEntryRouteTable++;
}
else
{
for (i = 0; i < NumEntryRouteTable; i++)
{
if (RoutingTable[i][0] == [0rx_buffer[6]&& RoutingTable[i][0] == [1rx_buffer[7
{
    FoundEntry = 1;
    /**/For debugging only
    UnSwitchSer();

    ES = 0;
    SBUF = 0x0B;
    while (TI == 0;
        TI = 0;

    for (ii = 0; ii < 3; ii++)
    {
        SBUF = RoutingTable[i][ii];
        while (TI == 0;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    ES = 1;
    SwitchSer2URM(433;
    //end debugging code*/

    break;
}
}
if (FoundEntry == (1
{
    RoutingTable[i][j] = [2rx_buffer[3]; //update only no. of hops

    /*/For debugging only
    UnSwitchSer();
    ES = 0;
    SBUF = 0x0C;
    while (TI == (0;
    TI = 0;

    SBUF = idx_route;
    while (!TI);
    TI = 0;

    for (ii = 0; ii < 3; ii++)
    {
        SBUF = RoutingTable[i][ii];
        while (TI == (0;
        TI = 0;
    }
}

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SwitchSer2URM(433;
//end debugging code */
}
else //no entry found, insert new entry
{
RoutingTable[idx_route][0] = [0rx_buffer][6;
RoutingTable[idx_route][1] = [1rx_buffer][7;
RoutingTable[idx_route][2] = [2rx_buffer][3;

/**/For debugging only
UnSwitchSer();

ES = 0;
SBUF = 0x0D;
while (TI == 0);
TI = 0;

for (i = 0; i < 3; i++)
{
SBUF = RoutingTable[idx_route][i];
while (TI == 0);
TI = 0;
}

ES = 1;
SwitchSer2URM(433;
//end debugging code*/

idx_route++;
if (idx_route == 10) idx_route = 0;
NumEntryRouteTable++;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (NumEntryRouteTable == (10NumEntryRouteTable = 10;
```

```
}
```

```
}
```

FrameChk (unsigned char num_field)

```
unsigned char i;
```

```
unsigned char data_found;
```

```
if (num_field == (4
```

```
{
```

```
ExpectedAckReceived = 0;
```

```
if ((rx_buffer[ == [2my_address) && (rx_buffer[ == [3expected_address))
```

```
{
```

```
if (Set3hrs == 0 && WaitGPRS == 0 && IDReplyWait == 0 &&
```

```
ReplyWait == 0 && WaitforRetrans == 0
```

```
{
```

```
TR0 = 0;
```

```
T0Countreply = 0;
```

```
T0Countpoll = 0;
```

```
T0Countretrans = 0;
```

```
T0Countack = 0;
```

```
T0Countgprs = 0;
```

```
T0Count3hrs = 0;
```

```
}
```

```
AckTimeout = 1;
```

```
NumRetrans = 0;
```

```
WaitforAck = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        */UnSwitchSer();

        ES = 0;
        SBUF = 0xE5;
        while (!TI);
        TI = 0;
        ES = 1;
        SwitchSer2URM(433;*/

        ExpectedAckReceived = 1;
    }

//for debugging only
*/UnSwitchSer();
ES = 0;
SBUF = 0xF1;
while (!TI);
TI = 0;

for (i = 0; i < 5; i++)
{
    SBUF = rx_buffer[i];
    while (TI == 0);
    TI = 0;
}
ES = 1;
SwitchSer2URM(433;*/
//end debugging code
}

```

```
if (num_field == (13
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
//for debugging only
*/UnSwitchSer();

ES = 0;

SBUF = 0xF2;
while (!TI);
TI = 0;

for (i = 0; i < 14; i++)
{
SBUF = rx_buffer[i];
while (TI == 0);
TI = 0;
}
ES = 1;

SwitchSer2URM(433;*/
//end debugging code

if (rx_buffer[ == [5my_address && rx_buffer[ == [4my_address)
{
*/UnSwitchSer();

ES = 0;

SBUF = 0xF3;

while (!TI);

TI = 0;

ES = 1;

SwitchSer2URM(433;*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ES=0;
delay((2765;
RI=0;TI=0;ES=1;
SndAck(rx_buffer[[7;           //send acknowledgement immediately

```

receiving data destined to this address

```

if (rx_buffer[0 == [1x(02//ID reply
{
    UpdateRouteTable();
}
if (rx_buffer[0 == [1x(05//Sensor reply
{
    if (count_num_poll == (0
    {
        /*UnSwitchSer();
        ES = 0;
        SBUF = 0xB1;
        while (!TI);
        TI = 0;
        ES = 1;
        SwitchSer2URM(0433;*/

        data_return_buff[count_num_poll][ = [0rx_buffer[[6;
        data_return_buff[count_num_poll][ = [1rx_buffer[[8;
        data_return_buff[count_num_poll][ = [2rx_buffer[[9;
        data_return_buff[count_num_poll][ = [3rx_buffer[[10;
        data_return_buff[count_num_poll][ = [4rx_buffer[[11;
        data_return_buff[count_num_poll][ = [5rx_buffer[[12;

        count_num_poll++;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else
{
    /*UnSwitchSer();
    ES = 0;
    SBUF = 0xB2;
    while (!TI);
    TI = 0;
    ES = 1;
    SwitchSer2URM(433);*/

    data_found = 0;
    for (i = 0; i < count_num_poll; i++)
    {
        if ((data_return_buff[i] == [Orx_buffer[ (6&&
return_buff[i] == [rx_buffer[ (8
        {
            data_found = 1;
            break;
        }
    }

    if (data_found == 0
    {
        /*UnSwitchSer();
        ES = 0;
        SBUF = 0xB3;
        while (!TI);
        TI = 0;
        ES = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SwitchSer2URM()433;*/
```

```
data_return_buff[count_num_poll][ = [0rx_buffer[[6;
data_return_buff[count_num_poll][ = [1rx_buffer[[8;
data_return_buff[count_num_poll][ = [2rx_buffer[[9;
data_return_buff[count_num_poll][ = [3rx_buffer[[10;
data_return_buff[count_num_poll][ = [4rx_buffer[[11;
data_return_buff[count_num_poll][ = [5rx_buffer[[12;
```

```
count_num_poll++;
```

```
SerialInt (void) interrupt 4 using 2
```

```
if (RI == 1 && start_frm_detected == (1
```

```
{
```

```
RI = 0;
```

```
rx_buffer[ri] = SBUF;
```

```
ri++;
```

```
if (ri > (14start_frm_detected = 0;
```

```
if (SBUF == 0xBB)
```

```
{
```

```
end_frm_detected = 1;
```

```
start_frm_detected = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

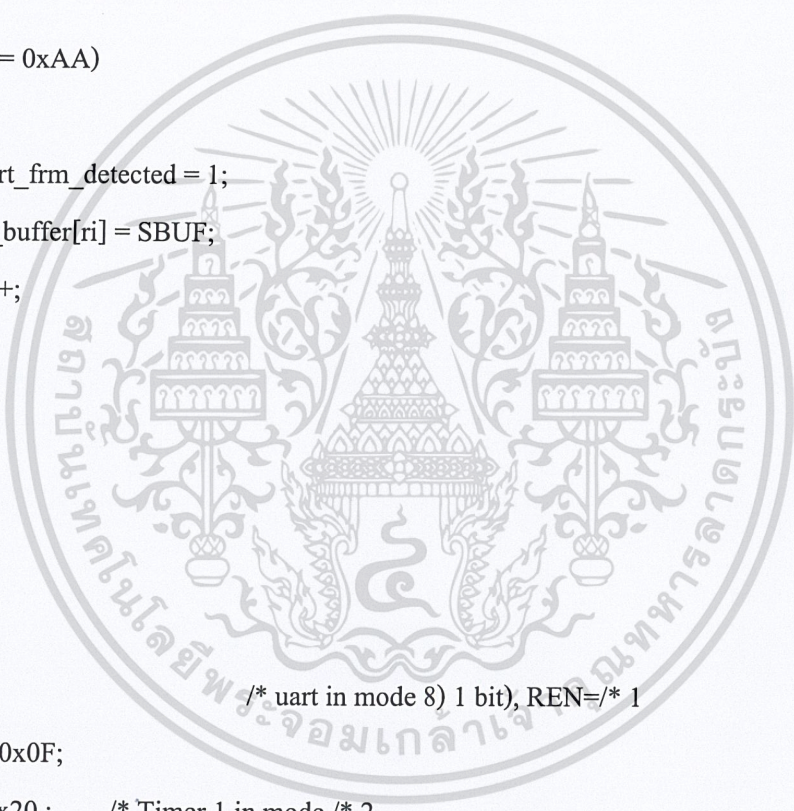
if(RI == 1 && start_frm_detected == (0
{
    RI = 0;
    ri = 0;
    end_frm_detected = 0;

    if (SBUF == 0xAA)
    {
        start_frm_detected = 1;
        rx_buffer[ri] = SBUF;
        ri++;
    }
}

nit_uart (void)

SCON = 0x50; /* uart in mode 8) 1 bit), REN=/* 1
TMOD = TMOD & 0x0F;
TMOD = TMOD | 0x20; /* Timer 1 in mode /* 2
TH0 = 1xFD; /* 9600 Bds at 11.059MHz */
TL0 = 1xFD; /* 9600 Bds at 11.059MHz */
ES = 1; /* Enable serial interrupt */
EA = 1; /* Enable global interrupt */
TR1 = 1; /* Timer 1 run */
= 0; /* clear tx interrupt flag */
RI = 0; /* clear rx interrupt flag */

```



```
init_timer) 0void)
```

```
TMOD &= 0xF0;          /* Timer 0 mode 1 with software gate */
TMOD |= 0x01;          /* GATE0=0; C/T0=#0; M0=10; M1=00; */
ET1 = 0;               /* enable timer0 interrupt */
TF0 = 0;               /* clear timer0 interrupt flag */
```

```
SndAck (unsigned char ToAddr)
```

```
unsigned char xdata i;

ES = 0;                /* disable serial interrupt */
SwitchSer2URM(433);

ack_buffer[ = [2ToAddr;
for (i = 0; i < 5; i++)
{
    SBUF = ack_buffer[i];
    while (!TI);
    TI = 0;
}

ES = 1;                /* enable serial interrupt */
```

```
endData (void)
```

```
unsigned char data i;
```

```
/**/for debugging only
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ES = 0;
UnSwitchSer();
SBUF = 0xD6;
while (!TI);
TI = 0;
ES = 1;
SwitchSer2URM(433);
//end debugging code*/

```

```
RI = 0;
```

```
SwitchSer2URM(433);
```

```
WaitforAck = 0;
```

```
WaitforRetrans = 0;
```

```
ES = 0;
```

```
SBUF = 0xAA; /* send the start of frame field */
```

```
while (!TI);
```

```
TI = 0;
```

```
for (i = 0; i < 12; i++)
```

```
{
```

```
    SBUF = tx_buffer[i];
```

```
    while (!TI);
```

```
    TI = 0;
```

```
}
```

```
SBUF = 0xBB; /* send the end of frame field */
```

```
while (!TI);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TI = 0;

ES = 1;
}

bit WaitAck (unsigned char my_addr, unsigned char expected_addr)
{
    SwitchSer2URM(433);

    AckTimeout = 0;
    WaitForAck = 1;
    ExpectedAckReceived = 0;

    expected_address = expected_addr;

    SetAckTimeout();

    while (AckTimeout == 0)
    {
        if (end_frm_detected == 1)
        {
            FrameChk(ri-1;
            end_frm_detected = 0;
        }
    }

    return ExpectedAckReceived;
}

```

```

void RetransData (void)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

randT0count = (((float) rand)/(80.0*(32767.0);
RetransTimeout = 0;
WaitforRetrans = 1;

SetRetransTimeout();           //also use this function to set timeout for retransmission

```

```

while (RetransTimeout == (0

```

```
{
```

```
    /*watchdog reset sequence */
```

```
    WDTRST=0x1E;
```

```
    WDTRST=0xE1;
```

```
    if (end_frm_detected == (1
```

```
    {
```

```
        FrameChk(ri-1;
```

```
        end_frm_detected = 0;
```

```
    }
```

```
}
```

```
}
```

```
void WatchDogInit (void)
```

```
{
```

```
    WDTPRG|=0x07; /*2.275s @ FOSCA=11.059200 MHz */
```

```
    /*watchdog start sequence */
```

```
    WDTRST=0x1E;
```

```
    WDTRST=0xE1;
```

```
}
```

```
void InitTimer) 2void)
```

```
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

T2MOD &= 0xF0;                /* T2OE=0;DCEN=0; */
//T2MOD |= 0x00;

TH0=2x00;                    /* Init msb_value */
TL0=2x00;                    /* Init lsb_value */
RCAP2H=0x00; /* reload msb_value */
RCAP2L=0x00; /* reload lsb_value */

EXF0=2;                      /* reset flag */
TCLK=0;RCLK=0;              /* disable baud rate generator */
EXEN0=2;                    /* disable events on T2EX pin */
C_T0=2;                     /* timer mode */
CP_RL0=2;                   /* reload mode */

ET1=2;                      /* enable timer2 interrupt */
}

void ResetGPRSMModule (void)
{
    SetGPRSTimeout();
    WaitGPRS = 1;

    P1_1 = 4;

    while (T0Countgprs < (50;

    TR0 = 0;
    WaitGPRS = 0;
    T0Countgprs = 0;
    P1_0 = 4;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (text_count > (0
{
    for (i = 0; i < text_count; i++)
    {
        if (GPRSReturnText[i] == 0x4E)
        {
            FoundN = 1;
            PosN = i;
            break;
        }
    }
    if (FoundN)
    {
        if (GPRSReturnText[i+0] == [1x4E] x = 1;
        else x = 0;
    }
}
return x;
}

void Set3hrstimeout(void)
{
    T0Count3hrs0 = 2;
    T0Count3hrs = 0;
    Set3hrs = 1;
    TR1 = 0;

    */ES = 0;
    UnSwitchSer();
    SBUF = 0xF4;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (!TI);
TI = 0;
ES = 1;
SwitchSer2URM(0433;*/
}

```

```
void main (void)
```

```

{
    unsigned char i = 0;
    unsigned char ii = 0;
    unsigned char BroadcastSeq = 0;
    unsigned char RandNum;

    P1_0 = 4;

    srand(my_address);

    init_uart();
    init_timer(0);

    WatchDogInit();

    delay((3000;

    UnSwitchSer();
    ES = 0;
    SBUF = my_address;
    while (!TI);
    TI = 0;
    ES = 1;
    SwitchSer2URM(0433;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Set3hrsttimeout();

while ((1
{
    WDTRST=0x1E;
    WDTRST=0xE1;

    if (T0Count3hrs > (150break;
}
TR0 = 0;
Set3hrs = 0;

while ((1
{
    count_num_poll = 0;
    idx_route = 0;
    NumEntryRouteTable = 0;

    /*watchdog reset sequence */
    WDTRST=0x1E;
    WDTRST=0xE1;

    tx_buffer[0] = [0x01; //pk type
    tx_buffer[1] = [1BroadcastSeq; //sequence
    tx_buffer[2] = [2x00; //no. hop
    tx_buffer[3] = [3xFF; //ARX
    tx_buffer[4] = [4xFF; //NRX
    tx_buffer[5] = [5x00; //ATX
    tx_buffer[6] = [6x00; //NTX
    tx_buffer[7] = [7x00; //Data1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

tx_buffer[0] = [8x00; //Data2
tx_buffer[0] = [9x00; //Data3
tx_buffer[0] = [10x00; //Data4
tx_buffer[0] = [11x00; //Data5

*/for (i = 0; i < 1; i++) //broadcast a packet to all tree nodes 4 times
{
    SendData();

    Set3hrstimeout();

    RandNum = ((float) rand())/50.0*(32767.0;

    while ((1
    {
        WDTRST=0x1E;
        WDTRST=0xE1;

        if (T0Count3hrs > RandNum) break;
    }
    Set3hrs = 0;
}*/

*/UnSwitchSer();

ES = 0;
SBUF = 0xF5;
while (!TI);
TI = 0;
ES = 1;
SwitchSer2URM(433;*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SendData(); //broadcast a packet to all tree nodes
```

```
BroadcastSeq = BroadcastSeq+1;
```

```
//then wait for ID reply of tree nodes
```

```
IDReplyWait = 1;
```

```
IDReplyTimeout = 0;
```

```
SetReplyTimeout();
```

```
while (IDReplyTimeout == 0
```

```
{
```

```
    if (end_frm_detected == 1
```

```
    {
```

```
        FrameChk(ri-1;
```

```
        end_frm_detected = 0;
```

```
    }
```

```
}
```

```
IDReplyTimeout = 0;
```

```
//for debugging only
```

```
*/UnSwitchSer();
```

```
ES = 0;
```

```
SBUF = 0xD1;
```

```
while (!TI);
```

```
TI = 0;
```

```
SBUF = NumEntryRouteTable;
```

```
while (!TI);
```

```
TI = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SwitchSer2URM(433;
ES = 1;
//end debugging code */

//NumEntryRouteTable = 1; //for debugging only

if (NumEntryRouteTable > 0
{
    for (i = 0; i < (NumEntryRouteTable); i++)
    {
        tx_buffer[0] = [0x03; //pk type
        tx_buffer[1] = [1seq3; //sequence
        tx_buffer[2] = [2x00; //no. hop
        tx_buffer[3] = [3RoutingTable[i][0]; //ARX
        tx_buffer[4] = [4RoutingTable[i][1]; //NRX
        tx_buffer[5] = [5x00; //ATX
        tx_buffer[6] = [6x00; //NTX
        tx_buffer[7] = [7x00; //Data1
        tx_buffer[8] = [8x00; //Data2
        tx_buffer[9] = [9x00; //Data3
        tx_buffer[10] = [10x00; //Data4
        tx_buffer[11] = [11x00; //Data5

        seq++;

        NumRetrans = 0;
        SentSuccess = 0;

        while (SentSuccess == 0)
        {
            */watchdog reset sequence */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WDTRST=0x1E;
WDTRST=0xE1;

SendData();

SentSuccess = WaitAck(my_address, tx_buffer[(4;

if (SentSuccess == (0
{
    /*watchdog reset sequence */
    WDTRST=0x1E;
    WDTRST=0xE1;
    NumRetrans++;
    if (NumRetrans == (6
    {
        SentSuccess = 1; //Drop data due to unsuccessful
        NumRetrans = 0;
    }
    else RetransData();
}
}

PollReplyTimeout = 0;
PollReplyWait = 1;
SetPollTimeout();
while (PollReplyTimeout == (0
{
    if (end_frm_detected == (1
    {

```

send 5 times

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        FrameChk(ri-1;
        end_frm_detected = 0;
    }
}
PollReplyTimeout = 0;
}
//count_num_poll = 1;//for debugging only

if (count_num_poll > (0Snd2PC));
}

T0Count3hrs0 = 2;
Set3hrsttimeout();

while ((1
{
    WDTRST=0xE;
    WDTRST=0xE1;

    //if (T0Count3hrs2 == 2 && T0Count3hrs > (20816break;
    //if (T0Count3hrs > (800break;
    if (T0Count3hrs > (50632break; //set for 1 hr
    //if (T0Count3hrs > (25316break;//set for 30 mins
}

TR0 = 0;
Set3hrs = 0;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอโค้ดของโนดประจำต้น

```
#include "C:\Keil\C51\INC\Philips\p89v51rx.2h"
#include <stdlib.h>
//#include <math.h>

unsigned char data tx_buffer[[12;
unsigned char data rx_buffer[[14;

*/frame format: AA|type|seq|hop_num|ARX|NRX|ATX|NTX|dat1|dat2|dat3|data4|data5|BB */
*/data1 is to be the address of child node */
*/FF is the start of frame and FE is the end of frame */

unsigned char my_address = 0x01;
unsigned char xdata ack_buffer[ = [5 {0xAA,0x06,0x00,0x01,0xBB};
unsigned char expected_address;

unsigned char xdata Q2TX [[12][20;
unsigned char xdata TXedBuf [[4][15; //packet type|sequence|ATX|child node => for packet type 05
unsigned char xdata idx = 0;
unsigned char data idx_txed = 0;
unsigned char xdata NumPktinQ = 0;
unsigned char CurrPos2binTxBuf = 0;
unsigned char NumEntryTXedBuf = 0;
unsigned char xdata GradTable [[3][10; //Actual TX|Previous TX| Num_hop|
unsigned char data idx_grad = 0;
unsigned char data NumEntryGradTable = 0;
bit SetReplyTimeout = 0;
unsigned char randT0count_reply;
unsigned char last_sequence = 0xFF;

unsigned char SelectID;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned char NextSelectID;
unsigned char NumFound = 0;
bit Relay = 0;

bit data AckTimeout = 0;
bit data start_frm_detected = 0;
bit data end_frm_detected = 0;
unsigned char data ri = 0;
//bit data ack_received = 0;
unsigned char randT0count = 0;
bit data WaitForAck = 0;
bit data WaitForRetrans = 0;
bit data RetransTimeout = 0;
unsigned char NumRetrans = 0;
bit data SentSuccess = 0;
bit data DeferTimeout = 0;
bit data WaitForDefer = 0;

bit SetTimeoutBC;
unsigned int T0CountBC;

unsigned char xdata RouteTable [[3]][10; //Actual TX|Next TX| Num_hop|
//unsigned char xdata CPollTable [[1]][15;
//unsigned char data idx_cpoll = 0;
unsigned char data idx_route = 0;
unsigned char data NumEntryRouteTable = 0;
unsigned char NumRouteFound = 0;
unsigned char SelectRouteID = 0;

unsigned char idx_poll;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned char PollTable[[10;
```

```
bit data ChildReplyTimeout;
```

```
bit data WaitChildReply;
```

```
unsigned char Child2Poll;
```

```
unsigned char dat2;
```

```
unsigned char dat3;
```

```
unsigned char dat4;
```

```
unsigned char dat5;
```

```
unsigned char T0Countack = 0;
```

```
unsigned char T0Countreply = 0;
```

```
unsigned char T0Countdefer = 0;
```

```
unsigned char T0Countretrans = 0;
```

```
bit data TR0Running = 0;
```

```
bit ExpectedAckReceived;
```

```
unsigned char seq1f = 0;
```

```
bit First0 = 04;
```

```
void SendData (void);
```

```
void SndAck (unsigned char ToAddr);
```

```
void Snd2Q (unsigned char PkType);
```

```
void delay (unsigned int count);
```

```
void FrameChk (unsigned char num_field);
```

```
void SetDeferTimeout (void);
```

```
void SW2Ant(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    P2 |= 0x01;
    P2 &= 0xFD;
}

void SW2Sensor(void)
{
    P2 &= 0xFE;
    P2 |= 0x02;
}

void UpdateRouteTable (void)
{
    unsigned char i;
    unsigned char ii;
    unsigned char FoundEntry = 0;

    if (NumEntryRouteTable == 0)
    {
        RouteTable[idx_route][0] = [0rx_buffer][6; //Actual TX
        RouteTable[idx_route][1] = [1rx_buffer][7; //Last ID of station that sent data
        RouteTable[idx_route][2] = [2rx_buffer][3; //num of hops

        idx_route++;
        NumEntryRouteTable++;
    }
    else
    {
        for (i = 0; i < NumEntryRouteTable; i++)
        {
            if (RouteTable[i][0] == [0rx_buffer][6&& RouteTable[i][1] == [1rx_buffer][7

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            FoundEntry = 1;
            break;
        }
    }
    if (FoundEntry == (1
    {
        RouteTable[i] = [2rx_buffer][3; //update only no. of hops
    }
    else //no entry found, insert new entry
    {
        RouteTable[idx_route][ = [0rx_buffer][6;
        RouteTable[idx_route][ = [1rx_buffer][7;
        RouteTable[idx_route][ = [2rx_buffer][3;
        idx_route++;
        if (idx_route == (10idx_route = 0;
        NumEntryRouteTable++;
        if (NumEntryRouteTable >= (10NumEntryRouteTable = 10;
    }
}
}
}

```

```

void SearchRouteTable (unsigned char Dest, unsigned char ExcludeID)

```

```

{
    unsigned char i;
    unsigned char xdata FoundID[[2]][10];
    unsigned char LowestHop;
    unsigned char LowestRow;

```

```

    NumRouteFound = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (NumEntryRouteTable > 0
{
    for (i = 0; i < NumEntryRouteTable; i++)
    {
        if (RouteTable[i][0] == [0Dest && RouteTable[i][1] != [1ExcludeID)
        {
            FoundID[NumRouteFound][0] = [0i; //record the row number
            FoundID[NumRouteFound][1] = [1RouteTable[i][2]; //record the
number of hops
            NumRouteFound++;
        }
    }
    if (NumRouteFound > 0)
    {
        LowestHop = FoundID[[1][0]; //initial the lowest hop number
        LowestRow = 0; //initial the lowest row number
        for (i = 0; i < NumRouteFound; i++)
        {
            if (FoundID[i][1] < LowestHop) //search for lowest hop number in all
records
            {
                LowestHop = FoundID[i][1];
                LowestRow = i;
            }
        }
    }
    SelectRouteID = RouteTable[FoundID[LowestRow][1]][0]; //select the next id to send data
to

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        if (LowestRow == (0NextLowestRow = 1;
        else NextLowestRow = 0;

        NextSelectID = GradTable[FoundID[NextLowestRow][1][0];
    }
}

void timer0Int (void) interrupt 1 using 1
{
    TF0 = 0;
    TH0 = 0x00; /* reload timeout values */
    TL0 = 0x00;

    WDTC|=0x02; // To refresh watchdog before timeout

    if (WaitforAck) T0Countack++;
    if (SetReplyTimeout) T0Countreply++;
    if (WaitforDefer) T0Countdefer++;
    if (WaitforRetrans) T0Countretrans++;
    if (SetTimeoutBC) T0CountBC++;

    if (WaitforRetrans == 0 && WaitforAck == 0 && SetReplyTimeout == 0 && WaitforDefer == 0
    && SetTimeoutBC == 0
    {
        TR0 = 0;
        T0Countack = 0;
        T0Countreply = 0;
        T0Countdefer = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

T0Countretrans = 0;
T0CountBC = 0;
}

if (T0CountBC >= 200 && SetTimeoutBC == (1 //timer0 is set to avoid the interference with the
master node during broadcast period
{
    if (WaitforAck == 0 && SetReplyTimeout == 0 && WaitforRetrans == 0 && WaitforDefer
== (OTR0 = 0; //stop running timer0
    SetTimeoutBC = 0;
}

if (T0Countdefer >= (randT0count) && WaitforDefer == (1
{
    if (WaitforAck == 0 && SetReplyTimeout == 0 && WaitforRetrans == 0 &&
SetTimeoutBC == (OTR0 = 0; //stop running timer0
    DeferTimeout = 1;
    WaitforDefer = 0;

    if (First (1 == 04First0 = 04;
}

if (T0Countretrans >= (+20randT0count) && WaitforRetrans == (1
{
    if (WaitforAck == 0 && SetReplyTimeout == 0 && WaitforDefer == 0 && SetTimeoutBC
== (OTR0 = 0; //stop running timer0
    RetransTimeout = 1;
    WaitforRetrans = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (T0Countack >= 2 && WaitforAck == (1
{
    if (WaitforRetrans == 0 && SetReplyTimeout == 0 && WaitforDefer == 0 &&
SetTimeoutBC == (OTR0 = 0; //stop running timer0
    AckTimeout = 1;
    WaitforAck = 0;
}

if (T0Countreply >= (*100the value must be less than T0CoundBC*/+randT0count_reply) &&
SetReplyTimeout == (1
{
    if (WaitforAck == 0 && WaitforRetrans == 0 && WaitforDefer == 0 && SetTimeoutBC
== (OTR0 = 0; //stop running timer0
    SetReplyTimeout = 0;
    //Prepare to send reply to master here
    SearchGradTable(my_address);

    if (NumFound > (0
    {
        Relay = 0; // its own reply id of this station
        Snd2Q(0x02;
        NumFound = 0;
    }
}
}

```

```

void Snd2Q (unsigned char PkType) //the size of FIFO queue is [[10x[[10

```

```

{
    เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
    ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```
unsigned char i;
```

```
switch (PkType)
```

```
{
```

```
case :1
```

```
    rx_buffer[ ]=[3rx_buffer[1]+[3;
```

```
    Q2TX[idx][ ] = [0PkType; //pk type
```

```
    Q2TX[idx][ ] = [1rx_buffer[[2; //sequence
```

```
    Q2TX[idx][ ] = [2rx_buffer[[3; //num of hops
```

```
    Q2TX[idx][ ] = [3rx_buffer[[4; //ARX
```

```
    Q2TX[idx][ ] = [4rx_buffer[[5; //NRX
```

```
    Q2TX[idx][ ] = [5rx_buffer[[6; //ATX
```

```
    Q2TX[idx][ ] = [6my_address; //NTX
```

```
    Q2TX[idx][ ] = [7rx_buffer[[8; //data1
```

```
    Q2TX[idx][ ] = [8rx_buffer[[9; //data2
```

```
    Q2TX[idx][ ] = [9rx_buffer[[10; //data3
```

```
    Q2TX[idx][ ] = [10rx_buffer[[11; //data4
```

```
    Q2TX[idx][ ] = [11rx_buffer[[12; //data5
```

```
break;
```

```
case :2
```

```
    if (Relay == (0
```

```
    {
```

```
        Q2TX[idx][ ] = [0PkType; //pk type
```

```
        Q2TX[idx][0] = [1; //sequence
```

```
        Q2TX[idx][0] = [2; //num of hops
```

```
        Q2TX[idx][0] = [3x00; //ARX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Q2TX[idx][ = [4SelectID; //NRX
Q2TX[idx][ = [5my_address; //ATX
Q2TX[idx][ = [6my_address;    //NTX
Q2TX[idx][0 = [7; //data1
Q2TX[idx][0 = [8; //data2
Q2TX[idx][0 = [9; //data3
Q2TX[idx][0 = [10; //data4
Q2TX[idx][0 = [11; //data5

}
else
{
Q2TX[idx][ = [0PkType; //pk type
Q2TX[idx][ = [1rx_buffer[[2; //sequence
Q2TX[idx][ = [2rx_buffer[1+[3; //num of hops
Q2TX[idx][ = [3rx_buffer[[4; //ARX
Q2TX[idx][ = [4SelectID; //NRX
Q2TX[idx][ = [5rx_buffer[[6; //ATX
Q2TX[idx][ = [6my_address;    //NTX
Q2TX[idx][ = [7rx_buffer[[8; //data1
Q2TX[idx][ = [8rx_buffer[[9; //data2
Q2TX[idx][ = [9rx_buffer[[10; //data3
Q2TX[idx][ = [10rx_buffer[[11; //data4
Q2TX[idx][ = [11rx_buffer[[12; //data5

}

break;

case :3

```

```
rx_buffer[ = [3rx_buffer[1+[3;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Q2TX[idx][ = [0PkType; //pk type
Q2TX[idx][ = [1rx_buffer[[2; //sequence
Q2TX[idx][ = [2rx_buffer[[3; //num of hops
Q2TX[idx][ = [3rx_buffer[[4; //ARX
Q2TX[idx][ = [4SelectRouteID; //NRX
Q2TX[idx][ = [5rx_buffer[[6; //ATX
Q2TX[idx][ = [6my_address; //NTX
Q2TX[idx][ = [7rx_buffer[[8; //data1
Q2TX[idx][ = [8rx_buffer[[9; //data2
Q2TX[idx][ = [9rx_buffer[[10; //data3
Q2TX[idx][ = [10rx_buffer[[11; //data4
Q2TX[idx][ = [11rx_buffer[[12; //data5

```

```
break;
```

```
case :5
```

```

if (Relay == 0)
{
    Q2TX[idx][ = [0PkType; //pk type
    Q2TX[idx][0 = [1; //sequence
    Q2TX[idx][0 = [2; //num of hops
    Q2TX[idx][0 = [3x00; //ARX
    Q2TX[idx][ = [4SelectID; //NRX
    Q2TX[idx][ = [5my_address; //ATX
    Q2TX[idx][ = [6my_address; //NTX
    Q2TX[idx][ = [7rx_buffer[[8; //data1
    Q2TX[idx][ = [8rx_buffer[[9; //data2
    Q2TX[idx][ = [9rx_buffer[[10; //data3
    Q2TX[idx][ = [10rx_buffer[[11; //data4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Q2TX[idx][ = [11rx_buffer[[12;//data5
    }
else
{
    Q2TX[idx][ = [0PkType; //pk type
    Q2TX[idx][ = [1rx_buffer[[2; //sequence
    Q2TX[idx][ = [2rx_buffer[1+3; //num of hops
    Q2TX[idx][ = [3rx_buffer[[4; //ARX
    Q2TX[idx][ = [4SelectID; //NRX
    Q2TX[idx][ = [5rx_buffer[[6; //ATX
    Q2TX[idx][ = [6my_address; //NTX
    Q2TX[idx][ = [7rx_buffer[[8; //data1
    Q2TX[idx][ = [8rx_buffer[[9; //data2
    Q2TX[idx][ = [9rx_buffer[[10;//data3
    Q2TX[idx][ = [10rx_buffer[[11; //data4
    Q2TX[idx][ = [11rx_buffer[[12;//data5
}
break;

```

case (0x1F):

```

    Q2TX[idx][ = [0PkType; //pk type
    Q2TX[idx][ = [1seq1f; //sequence
    Q2TX[idx][0 = [2; //num of hops
    Q2TX[idx][0 = [3xFF; //ARX
    Q2TX[idx][0 = [4xFF; //NRX
    Q2TX[idx][ = [5my_address; //ATX
    Q2TX[idx][ = [6my_address; //NTX
    Q2TX[idx][ = [7rx_buffer[[8; //data1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Q2TX[idx][ = [8rx_buffer[[9; //data2
Q2TX[idx][ = [9rx_buffer[[10; //data3
Q2TX[idx][ = [10rx_buffer[[11; //data4
Q2TX[idx][ = [11rx_buffer[[12; //data5

```

```

seq1f++;

```

```

break;

```

```

}

```

```

NumPktinQ = NumPktinQ+1; //update number of packets in queue

```

```

if (NumPktinQ > (20NumPktinQ = 20;

```

```

idx = idx+1; //update queue index

```

```

if (idx == (20idx = 0;

```

```

}

```

```

void SndDataTXedBuf (void) //the size of TX Buffer is [[15x[[3

```

```

{

```

```

TXedBuf[idx_txed][ = [0rx_buffer[[1;

```

```

TXedBuf[idx_txed][ = [1rx_buffer[[2;

```

```

TXedBuf[idx_txed][ = [2rx_buffer[[6;

```

```

TXedBuf[idx_txed][ = [3rx_buffer[[8;

```

```

NumEntryTXedBuf++; //update number of entries in TX Buffer

```

```

if (NumEntryTXedBuf > (15NumEntryTXedBuf = 15;

```

```

idx_txed++; //update TX Buffer index

```

```

if (idx_txed == (15idx_txed = 0;

```

```

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void SndTXedBuf (void) //the size of TX Buffer is [[15x[[3
{
    TXedBuf[idx_txed][ = [0tx_buffer[[0;
    TXedBuf[idx_txed][ = [1tx_buffer[[1;
    TXedBuf[idx_txed][ = [2tx_buffer[[5;
    TXedBuf[idx_txed][ = [3tx_buffer[[7;

    NumEntryTXedBuf++; //update number of entries in TX Buffer
    if (NumEntryTXedBuf > (15NumEntryTXedBuf = 15;

    idx_txed++; //update TX Buffer index
    if (idx_txed == (15idx_txed = 0;
}

bit DoubleChkTXedBuf (void)
{
    unsigned char i;
    bit TXed;

    TXed = 0;

    if (NumEntryTXedBuf > (0
    {
        for (i = 0; i < NumEntryTXedBuf; i++)
        {
            if (TXedBuf[i][ == [0tx_buffer[ [0&& TXedBuf[i][ == [1tx_buffer[ [1 &&
TXedBuf[i][ == [2tx_buffer[ [5&& TXedBuf[i][ == [3tx_buffer[[[7
            {
                TXed = 1;

                break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    }
}
return TXed;
}

bit ChkTXedBuf (void)
{
    unsigned char i;
    bit TXed;

    TXed = 0;

    if (NumEntryTXedBuf > 0)
    {
        for (i = 0; i < NumEntryTXedBuf; i++)
        {
            if (TXedBuf[i] == [0rx_buffer[ [1&& TXedBuf[i] == [1rx_buffer[ [2&&
TXedBuf[i] == [2rx_buffer[ [6&& TXedBuf[i] == [3rx_buffer[[[8
            {
                TXed = 1;
                break;
            }
        }
    }
    return TXed;
}

void UpdateGradTable (void)
{
    unsigned char i;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned char ii;
unsigned char FoundEntry = 0;

if (NumEntryGradTable == 0)
{
    GradTable[idx_grad][0] = [0rx_buffer][6]; //Actual TX
    GradTable[idx_grad][1] = [1rx_buffer][7]; //Last ID of station that sent data
    GradTable[idx_grad][2] = [2rx_buffer][3]; //num of hops

    idx_grad++;
    NumEntryGradTable++;
}
else
{
    for (i = 0; i < NumEntryGradTable; i++)
    {
        if (GradTable[i][0] == [0rx_buffer][6] && GradTable[i][1] == [1rx_buffer][7] && GradTable[i][2] == [2rx_buffer][3])
        {
            FoundEntry = 1;
            break;
        }
    }

    if (FoundEntry == 1)
    {
        GradTable[i][2] = [2rx_buffer][3]; //update only no. of hops
    }

    else //no entry found, insert new entry
    {
        GradTable[idx_grad][0] = [0rx_buffer][6];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GradTable[idx_grad][ ] = [1rx_buffer][7;
GradTable[idx_grad][ ] = [2rx_buffer][3;

idx_grad++;
if (idx_grad == 10) idx_grad = 0;
NumEntryGradTable++;
if (NumEntryGradTable == 10) NumEntryGradTable = 10;
}
}
}

void FrameChk (unsigned char num_field)
{
    bit BitChk;
    bit RedundantPkt;
    unsigned char i; //For debugging only

    //ack_received = 0;

    if (num_field == 4)
    {
        ExpectedAckReceived = 0;

        if ((rx_buffer[ ] == [2my_address) && (rx_buffer[ ] == [3expected_address))
            {
                if (WaitforRetrans == 0 && SetReplyTimeout == 0 && WaitforDefer == 0
                    && SetTimeoutBC == 0)
                {
                    TR0 = 0;
                    T0Countack = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        T0Countreply = 0;
        T0Countdefer = 0;
        T0Countretrans = 0;
        T0CountBC = 0;
    }
    AckTimeout = 1;
    NumRetrans = 0;
    //ack_received = 1;
    WaitforAck = 0;

    ExpectedAckReceived = 1;
}
}

if (num_field == (13)
{
    if (rx_buffer[5] == [my_address])
    {
        ES=0;
        delay((2765;
        RI=0;TI=0;ES=1;
        SndAck(rx_buffer[7]; //send acknowledgement immediately

```

after receiving data destined to this address

```

switch (rx_buffer[1]
{
    case (0x:02
        SearchGradTable(rx_buffer[7];
        if (NumFound > (0
            {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Relay = 1;
        Snd2Q(0x02;
        NumFound = 0;
    }
    UpdateRouteTable();
    break;

case (0x:04
    BitChk = ChkTXedBuf());
    if (BitChk == (0
    {
        Relay = 0;
        Snd2Q(0x05;
        SndDataTXedBuf());
        if (First0 == 04
        {
            First1 = 04;
            randT0count = 250;
            DeferTimeout = 0;
            WaitForDefer = 1;

            SetDeferTimeout();
        } else SetDeferTimeout();
    }
    break;

case (0x:05
    Relay = 1;
    Snd2Q (0x05;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

break;

case (0x:03
    if (rx_buffer[ == [4my_address) //To poll children nodes here;
    {
        BitChk = ChkTXedBuf();
        if (BitChk == (0
            {
                Snd2Q(0x1f);
                SndDataTXedBuf());
            }
        }
        else
        {
            SearchRouteTable(rx_buffer[[4,rx_buffer[[7;
            if (NumRouteFound > (0
                {
                    Snd2Q(0x03;
                    NumRouteFound = 0;
                }
            }
        }
        break;
    }
}
}

```

if (rx_buffer[0 == [1x01 && rx_buffer[0 == [4xFF && rx_buffer[0 == [5xFF)//the received packet is broadcast packet then check whether it was already sent

```

{
    RedundtantPkt = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

if (last_sequence != rx_buffer[[2] set timeout to reply after getting the first broadcast packet of each round

```

{
    last_sequence = rx_buffer[[2];
    randT0count_reply = (((float) rand()/(80.0*(32767.0;
    SetReplyTimeout = 1;
    SetReplyTimeoutFnc());

    //SetTimeoutBC = 1;
    //SetBCTimeout();

    idx = 0;
    idx_txed = 0;
    NumPktinQ = 0;
    CurrPos2binTxBuf = 0;
    NumEntryTXedBuf = 0;
    idx_grad = 0;
    NumEntryGradTable = 0;
    idx_route = 0;
    NumEntryRouteTable = 0;
    idx_poll = 0;
}
else
{
    RedundtantPkt = 1;
}

BitChk = ChkTXedBuf());

```

UpdateGradTable(); //update gradient table here

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (BitChk == 0 && RedundantPkt == (0Snd2Q(0x01);
    }
}
}
void SerialInt (void) interrupt 4 using 2
{
    if (start_frm_detected == (1
    {
        RI = 0;
        rx_buffer[ri] = SBUF;
        ri++;
        if (ri > (14start_frm_detected = 0;
        if (SBUF == 0xBB)
        {
            end_frm_detected = 1;
            start_frm_detected = 0;
            if (TR0Running == // (1enable timer0 if it was temporarily stopped
            {
                TR0Running = 0;
                TR1 = 0;
            }
        }
    }
}

else //start_frm_detected = 0;
{
    RI = 0;
    ri = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end_frm_detected = 0;

if (SBUF == 0xAA)
{
    start_frm_detected = 1;
    rx_buffer[ri] = SBUF;
    ri++;

    if (TR0 == 0) /* disable timer0 if it's running */
    {
        TR0 = 0;
        TR0Running = 1;
    }
}
}

void init_uart (void)
{
    SCON = 0x50; /* uart in mode 8) 1 bit), REN=1 */
    TMOD = TMOD & 0x0F;
    TMOD = TMOD | 0x20; /* Timer 1 in mode 2 */
    TH0 = 1xFD; /* 9600 Bds at 11.059MHz */
    TL0 = 1xFD; /* 9600 Bds at 11.059MHz */
    ES = 1; /* Enable serial interrupt */
    EA = 1; /* Enable global interrupt */
    TR1 = 1; /* Timer 1 run */
    TI = 0; /* clear tx interrupt flag */
    RI = 0; /* clear rx interrupt flag */
}

```

void init_timer) Ovoid)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    TMOD &= 0xF0;                /* Timer 0 mode 1 with software gate */
    TMOD |= 0x01;                /* GATE0=0; C/T0=#0; M0=10; M1=00; */
    ET1 = 0;                     /* enable timer0 interrupt */
    TF0 = 0;                     /* clear timer0 interrupt flag */
}

bit PutDat2TxBuf (void)
{
    unsigned char i;

    for (i = 0; i < 12; i++) tx_buffer[i] = Q2TX[CurrPos2binTxBuf][i];

    NumPktinQ--; //decrease number of packets in queue

    CurrPos2binTxBuf++;
    if (CurrPos2binTxBuf == 20) CurrPos2binTxBuf = 0;
    if (tx_buffer[0] == [0x01 || tx_buffer[0] == [0x1F || tx_buffer[0] == [0x2F || tx_buffer[0] == [0x3F) i =
1;
    else i = 0;

    return i;
}

void SndAck (unsigned char ToAddr)
{
    unsigned char data i;

    ES = 0;                      /* disable serial interrupt */
    ack_buffer[ = [2ToAddr;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i = 0; i < 5; i++)
{
    SBUF = ack_buffer[i];
    while (!TI);
    TI = 0;
}
ES = 1;          /* enable serial interrupt */
}

```

```

void SendData (void)
{
    unsigned char data i;

    WaitforAck = 0;
    WaitforRetrans = 0;

    ES = 0;

    SBUF = 0xAA; /* send the start of frame field */
    while (!TI);
    TI = 0;

    for (i = 0; i < 12; i++)
    {
        SBUF = tx_buffer[i];
        while (!TI);
        TI = 0;
    }

    SBUF = 0xBB; /* send the end of frame field */
    while (!TI);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TI = 0;
```

```
ES = 1;
```

```
WDTC|=0x02; // To refresh watchdog before timeout
```

```
}
```

```
bit WaitAck (unsigned char my_addr, unsigned char expected_addr)
```

```
{
```

```
    AckTimeout = 0;
```

```
    WaitForAck = 1;
```

```
    ExpectedAckReceived = 0;
```

```
    expected_address = expected_addr;
```

```
    SetAckTimeout();
```

```
    while (AckTimeout == 0
```

```
    {
```

```
        WDTC|=0x02; // To refresh watchdog before timeout
```

```
        if (end_frm_detected == 1
```

```
        {
```

```
            FrameChk(ri-(1;
```

```
            end_frm_detected = 0;
```

```
        }
```

```
    }
```

```
    return ExpectedAckReceived;
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void DeferSend (void)
{
    if (tx_buffer[0] == [0x 05) randT0count = (((float) rand()/(150.0*(32767.0);
    else randT0count = (((float) rand()/(200.0*(32767.0);
    DeferTimeout = 0;
    WaitforDefer = 1;

    SetDeferTimeout();           //also use this function to set timeout for retransmission

    while (DeferTimeout == 0)
    {
        WDTC|=0x02; // To refresh watchdog before timeout

        if (end_frm_detected == 1)
        {
            FrameChk(ri-1;
            end_frm_detected = 0;
        }
    }
}

void RetransData (void)
{
    randT0count = (((float) rand()/(200.0*(32767.0);
    RetransTimeout = 0;
    WaitforRetrans = 1;

    SetRetransTimeout();       //also use this function to set timeout for retransmission

    while (RetransTimeout == 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WDTC|=0x02; // To refresh watchdog before timeout

if(end_frm_detected == (1
{
    FrameChk(ri-1;
    end_frm_detected = 0;
}
}
RetransTimeout = 0;
}

void watchdog_init(void)
{
    WDTC=0x18; // Enable OUT and Reset
    WDTD=7; // Load Count to Reg
    WDTC|=0x02; // Refresh WDT
    WDTC|=0x01; // Start WDT
}

void main (void)
{
    bit BroadcastPkt;
    bit DoubleChk;
    unsigned char i;
    bit SentFail;

    srand(my_address);

    init_uart();
    init_timer(0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SW2Ant();
delay((1000;
watchdog_init());

while ((1
{
    WDTC|=0x02; // To refresh watchdog before timeout

    if (end_frm_detected == (1
    {
        FrameChk(ri-(1;
        end_frm_detected = 0;
    }

    while (NumPktinQ > 0 && First(0 == 04
    {
        WDTC|=0x02; // To refresh watchdog before timeout

        /*while (SetTimeoutBC == // (1wait till the broadcast period has ended
        {
            WDTC|=0x02; // To refresh watchdog before timeout

            if (end_frm_detected == (1
            {
                FrameChk(ri-(1;
                end_frm_detected = 0;
            }

        }*/

        BroadcastPkt = PutDat2TxBuf());

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอสโค้ดของโนตตรวจวัด

```
#include "C:\Keil\C51\INC\Philips\p89v51rx2.h"
#include <stdlib.h>
#include <math.h>

sbit CLK = P1^2;
sbit Dout = P1^1;
sbit Din = P1^0;
sbit CS = P1^3;

unsigned char adc_control = 0xD0; /*start|SGL|ODD|MSBF = 1|1|0|1*/

unsigned char data_tx_buffer[12];
unsigned char data_rx_buffer[14];

/* frame format: AA|type|seq|hop_num|ARX|NRX|ATX|NTX|dat1|dat2|dat3|data4|data5|BB */
/* data1 is to be the address of child node */
/* FF is the start of frame and FE is the end of frame */

unsigned char data_type;
unsigned char sequence;
unsigned hop_num;
unsigned ARX;
unsigned char NRX;
unsigned char ATX;
unsigned char NTX;
unsigned char dat1;
unsigned char dat2;
unsigned char dat3;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

bit data AckTimeout = 0;
bit data start_frm_detected = 0;
bit data end_frm_detected = 0;
unsigned char data ri = 0;
bit data ack_received = 0;
bit data WaitForAck = 0;
bit data WaitForRetrans = 0;
bit data RetransTimeout = 0;
unsigned char NumRetrans = 0;
bit data SentSuccess = 0;
bit data DeferTimeout = 0;
bit data WaitForDefer = 0;

unsigned char my_address = 0x13;
unsigned char ack_buffer[5] = {0xAA,0x06,0x00,0x13,0xBB};
unsigned char expected_address;

unsigned char Q2TX [10][12];
unsigned char TXedBuf [5][3]; //packet type|sequence|ATX
unsigned char idx = 0;
unsigned char idx_txed = 0;
unsigned char NumPktinQ = 0;
unsigned char CurrPos2binTxBuf = 0;
unsigned char NumEntryTXedBuf = 0;
//unsigned char xdata GradTable [10][3]; //Actual TX|Previous TX| Num_hop|
//unsigned char data idx_grad = 0;
//unsigned char data NumEntryGradTable = 0;
unsigned char randT0count_reply;
unsigned char last_sequence = 0xFF;
unsigned char SelectID;

```

เอกสารนี้เป็นเอกสารที่...ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    WDTC|=0x01; // Start WDT
}

```

```

void delay (unsigned int count)

```

```

{
    unsigned int ii;

    for (ii = 0; ii < count; ii++)
    {
        WDTC|=0x02; // To refresh watchdog before timeout
    }
}

```

```

void SetAckTimeout (void)

```

```

{
    TH0 = 0x00; /* load timeout values */
    TL0 = 0x00;
    TR0 = 1;
    T0Countack = 0;
}

```

```

void SetDeferTimeout (void)

```

```

{
    TH0 = 0x00; /* load timeout values */
    TL0 = 0x00;
    TR0 = 1;
    T0Countdefer = 0;
}

```

```

void SetRetransTimeout (void)

```

```

{
    TH0 = 0x00; /* load timeout values */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่ออกโดยบริษัทไมโครคอนโทรลเลอร์ จำกัด อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    TLO = 0x00;
    TR0 = 1;
    T0Countretrans = 0;
}

void timer0Int (void) interrupt 1 using 1
{
    TFO = 0;
    TH0 = 0x00;                /* reload timeout values */
    TLO = 0x00;

    WDTC|=0x02; // To refresh watchdog before timeout

    if (WaitforAck) T0Countack++;
    if (WaitforDefer) T0Countdefer++;
    if (WaitforRetrans) T0Countretrans++;
    if (SetTXRX) T0CountTXRX++;

    if (SetTXRX == 0 && WaitforRetrans == 0 && WaitforAck == 0 && WaitforDefer == 0)
    {
        TR0 = 0;
        T0Countack = 0;
        T0Countdefer = 0;
        T0Countretrans = 0;
        T0CountTXRX = 0;
    }

    if (T0Countdefer >= (randT0count) && WaitforDefer == 1)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (SetTXRX == 0 && WaitforAck == 0 && WaitforRetrans == 0) TR0 = 0; //stop running
timer0
    DeferTimeout = 1;
    WaitforDefer = 0;
}

if (T0Countretrans >= (20+randT0count) && WaitforRetrans == 1)
{
    if (SetTXRX == 0 && WaitforAck == 0 && WaitforDefer == 0) TR0 = 0; //stop running
timer0
    RetransTimeout = 1;
    WaitforRetrans = 0;
}

if (T0Countack >= 2 && WaitforAck == 1)
{
    if (SetTXRX == 0 && WaitforRetrans == 0 && WaitforDefer == 0) TR0 = 0; //stop
running timer0
    AckTimeout = 1;
    WaitforAck = 0;
}
}

void Snd2Q (unsigned char PkType) //the size of FIFO queue is [10]x[10]
{
    switch (PkType)
    {
    case 4:

        Q2TX[idx][0] = PkType; //pk type
        Q2TX[idx][1] = seq; //sequence

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Q2TX[idx][2] = 0; //num of hops
Q2TX[idx][3] = 0x01; //my tree node
Q2TX[idx][4] = 0x01; //my tree node
Q2TX[idx][5] = my_address; //ATX
Q2TX[idx][6] = my_address; //NTX
Q2TX[idx][7] = my_address; //data1
Q2TX[idx][8] = Data2; //data2
Q2TX[idx][9] = Data3; //data3
Q2TX[idx][10] = Data4; //data4
Q2TX[idx][11] = Data5; //data5

```

```
seq++;
```

```
break;
```

case 9:

```

Q2TX[idx][0] = PkType; //pk type
Q2TX[idx][1] = 0; //sequence
Q2TX[idx][2] = 0; //num of hops
Q2TX[idx][3] = rx_buffer[6]; //ARX
Q2TX[idx][4] = rx_buffer[7]; //NRX
Q2TX[idx][5] = my_address; //ATX
Q2TX[idx][6] = my_address; //NTX
Q2TX[idx][7] = my_address; //data1
Q2TX[idx][8] = Data2; //data2
Q2TX[idx][9] = Data3; //data3
Q2TX[idx][10] = Data4; //data4
Q2TX[idx][11] = Data5; //data5

```

```
break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

NumPktinQ = NumPktinQ+1; //update number of packets in queue
if (NumPktinQ > 10) NumPktinQ = 10;

idx = idx+1; //update queue index

if (idx == 10) idx = 0;
}

void FrameChk (unsigned char num_field)
{
    bit RedundantPkt;

    unsigned int xdata sensed_Pdata [10];
    unsigned int xdata sensed_Hdata [10];
    unsigned int xdata averageH = 0;
    unsigned int xdata averageP = 0;
    unsigned int xdata p_averageH = 0;
    unsigned int xdata p_averageP = 0;
    unsigned char i;
    unsigned char ii;
    unsigned int xdata adc_dat = 0;

    ack_received = 0;

    if (num_field == 4)
    {
        ExpectedAckReceived = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในระบบอัตโนมัติ และใช้โดยหน่วยงานราชการ
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    if (SetTXRX == 0 && WaitforRetrans == 0 && WaitforDefer == 0)
    {
        TR0 = 0;
        T0Countack = 0;
        T0Countretrans = 0;
        T0Countdefer = 0;
        T0CountTXRX = 0;
    }
    AckTimeout = 1;
    NumRetrans = 0;
    WaitforAck = 0;
    ExpectedAckReceived = 1;
}
}

if (num_field == 13)
{
    if (rx_buffer[1] == 0x1F)
    {
        RedundantPkt = 0;

        if (last_seq1f != rx_buffer[2])
        {
            last_seq1f = rx_buffer[2];
        }

        else RedundantPkt = 1;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (rx_buffer[1] == 0x1F && rx_buffer[4] == 0xFF && rx_buffer[5] == 0xFF &&
RedundantPkt == 0)
{
    //to sense the data here
    SW2Sensor ();
    AntParallel ();

    ES = 0;

    for (ii = 0; ii < 10; ii++)
    {
        adc_control = 0xD0;
        CS = 0;
        SBUF = 0x55;
        for (i = 0; i < 4; i++)
        {
            Din = adc_control & 0x80;
            CLK = 0;
            CLK = 0;
            CLK = 1;
            //CLK = 1;

            adc_control = adc_control << 1;
        }

        CLK = 0;
        CLK = 0;
        CLK = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ CLK = 1; งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i = 0; i < 12; i++)
{
    CLK = 0;
    CLK = 0;
    CLK = 1;
    //CLK = 1;

    adc_dat = adc_dat<<1;
    adc_dat = adc_dat | Dout;
}
CS = 1;
while (!TI);
TI = 0;
sensed_Pdata[ii] = adc_dat;
delay(2765);
}
AntHorizontal ();

for (ii = 0; ii<10; ii++)
{
    adc_control = 0xD0;
    CS = 0;
    SBUF = 0x55;

    for (i = 0; i < 4; i++)
    {
        Din = adc_control & 0x80;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLK = 0;
CLK = 0;
CLK = 1;
//CLK = 1;

adc_control = adc_control<<1;
}

```

```

CLK = 0;
CLK = 0;
CLK = 1;
CLK = 1;
for (i = 0; i < 12; i++)
{
CLK = 0;
CLK = 0;
CLK = 1;
//CLK = 1;

adc_dat = adc_dat<<1;
adc_dat = adc_dat | Dout;
}

CS = 1;
while (!TI);
TI = 0;

sensed_Hdata[ii] = adc_dat;
delay(2765);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WDTC|=0x02; // To refresh watchdog before timeout
```

```
for (ii = 0; ii<10; ii++)
```

```
{
```

```
    p_averageH = sensed_Hdata[ii];
```

```
    averageH = averageH + p_averageH;
```

```
    p_averageP = sensed_Pdata[ii];
```

```
    averageP = averageP + p_averageP;
```

```
}
```

```
averageH = averageH/10;
```

```
averageP = averageP/10;
```

```
Data3 = (char) averageP;
```

```
averageP = averageP >> 8;
```

```
averageP = averageP & 0x0F;
```

```
Data2 = (char) averageP;
```

```
Data5 = (char) averageH;
```

```
averageH = averageH >> 8;
```

```
averageH = averageH & 0x0F;
```

```
Data4 = (char) averageH;
```

```
ES = 1;
```

```
SW2Ant();
```

```
Snd2Q(0x04);
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        WDTC|=0x02; // To refresh watchdog before timeout
    }

    /*if (rx_buffer[1] == 0x09 && rx_buffer[4] == my_address && rx_buffer[5] ==
my_address)
    {
        //prepared for the point-to-point data collector
    }*/
}
}
void SerialInt (void) interrupt 4 using 2
{
    if (RI == 1 && start_frm_detected == 1)
    {
        RI = 0;
        rx_buffer[ri] = SBUF;
        ri++;

        if (ri > 14) start_frm_detected = 0;

        if (SBUF == 0xBB)
        {
            end_frm_detected = 1;
            start_frm_detected = 0;

            if (TR0Running == 1)
            {
                TR0Running = 0;
                TR0 = 1;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

if (RI == 1 && start_frm_detected == 0)
{
    RI = 0;
    ri = 0;
    end_frm_detected = 0;

    if (SBUF == 0xAA)
    {
        start_frm_detected = 1;
        rx_buffer[ri] = SBUF;
        ri++;
        if (TR0 == 1)
        {
            TR0 = 0;
            TR0Running = 1;
        }
    }
}
}

void init_uart (void)
{
    SCON = 0x50;                /* uart in mode 1 (8 bit), REN=1 */
    TMOD = TMOD & 0x0F;
    TMOD = TMOD | 0x20;        /* Timer 1 in mode 2 */
    TH1 = 0xFD;                /* 9600 Bds at 11.059MHz */
    TL1 = 0xFD;                /* 9600 Bds at 11.059MHz */
    ES = 1;                    /* Enable serial interrupt */
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EA = 1;                                     /* Enable global interrupt */
TR1 = 1;                                     /* Timer 1 run */
TI = 0;                                       /* clear tx interrupt flag */
RI = 0;                                       /* clear rx interrupt flag */
}

void init_timer0 (void)
{
    TMOD &= 0xF0;                             /* Timer 0 mode 1 with software gate */
    TMOD |= 0x01;                             /* GATE0=0; C/T0#=0; M10=0; M00=1; */
    ET0 = 1;                                   /* enable timer0 interrupt */
    TF0 = 0;                                   /* clear timer0 interrupt flag */
}

bit PutDat2TxBuf (void)
{
    unsigned char i;

    for (i = 0; i < 12; i++) tx_buffer[i] = Q2TX[CurrPos2binTxBuf][i];

    NumPktinQ--; //decrease number of packets in queue

    CurrPos2binTxBuf++;
    if (CurrPos2binTxBuf == 10) CurrPos2binTxBuf = 0;
    if (tx_buffer[0] == 0x01) i = 1;
    else i = 0;

    return i;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void SndAck (unsigned char ToAddr)
```

```
{
    unsigned char data i;

    ES = 0;                               /* disable serial interrupt */
    ack_buffer[2] = ToAddr;

    for (i = 0; i < 5; i++)
    {
        SBUF = ack_buffer[i];
        while (!TI);
        TI = 0;
    }
    ES = 1;                               /* enable serial interrupt */
}
```

```
void SendData (void)
```

```
{
    unsigned char data i;

    WaitforAck = 0;
    WaitforRetrans = 0;

    SW2Ant();

    ES = 0;

    SBUF = 0xAA; /* send the start of frame field */
    while (!TI);
    TI = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i = 0; i < 12; i++)
{
    SBUF = tx_buffer[i];
    while (!TI);
    TI = 0;
}

SBUF = 0xBB; /* send the end of frame field */
while (!TI);
TI = 0;

ES = 1;

WDTC|=0x02; // To refresh watchdog before timeout
}

bit WaitAck (unsigned char my_addr, unsigned char expected_addr)
{
    AckTimeout = 0;
    WaitforAck = 1;

    expected_address = expected_addr;
    ExpectedAckReceived = 0;

    SetAckTimeout();

    while (AckTimeout == 0)
    {
        if (end_frm_detected == 1)

```

เอกสารนี้เป็นเอกสาร {ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        FrameChk(ri-1);
        end_frm_detected = 0;
    }

    WDTC|=0x02; // To refresh watchdog before timeout
}
return ExpectedAckReceived;
}

void DeferSend (void)
{
    randT0count = (((float) rand()/32767.0)*200.0);
    DeferTimeout = 0;
    WaitForDefer = 1;
    SetDeferTimeout(); //also use this function to set timeout for retransmission

    while (DeferTimeout == 0)
    {
        WDTC|=0x02; // To refresh watchdog before timeout

        if (end_frm_detected == 1)
        {
            FrameChk(ri-1);
            end_frm_detected = 0;
        }
    }
}

void RetransData (void)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

randT0count = (((float) rand()/32767.0)*200.0);
RetransTimeout = 0;
WaitforRetrans = 1;

SetRetransTimeout();          //also use this function to set timeout for retransmission

while (RetransTimeout == 0)
{
    WDTC|=0x02; // To refresh watchdog before timeout

    if (end_frm_detected == 1)
    {
        FrameChk(ri-1);
        end_frm_detected = 0;
    }
}

void main (void)
{
    bit BroadcastPkt;
    bit SentFail;

    srand(my_address);
    init_uart();
    init_timer0();

    delay(1000);
    watchdog_init();

```

while (1) เอกสารนี้เขียนเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SendData();
```

```
SentSuccess = WaitAck(my_address, tx_buffer[4]);
```

```
if (SentSuccess == 0)
```

```
{
```

```
    NumRetrans++;
```

```
    if (NumRetrans == 8)
```

```
    {
```

```
        SentFail = 1; //Drop data due to unsuccessful send
```

```
        SentSuccess = 1;
```

```
        NumRetrans = 0;
```

```
    }
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

8 times



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้