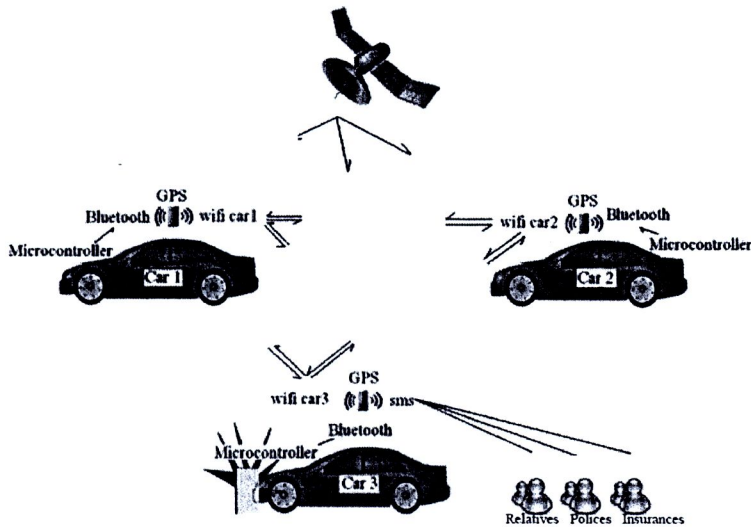


บทที่ 3

เทคนิคและวิธีการดำเนินงาน

3.1 ภาพรวมของระบบติดต่อสื่อสารและแจ้งเตือนการเกิดอุบัติเหตุระหว่างรถยนต์ผ่าน Wi-Fi เพื่อการขับขี่ที่ปลอดภัย



รูปที่ 3.1 ภาพรวมของระบบติดต่อสื่อสารและแจ้งเตือนการเกิดอุบัติเหตุระหว่างรถยนต์ผ่าน Wi-Fi เพื่อการขับขี่ที่ปลอดภัย

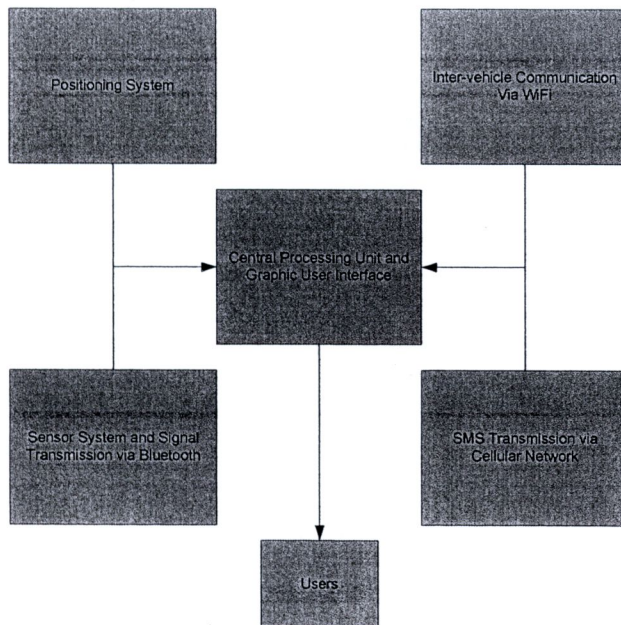
ระบบติดต่อสื่อสารและแจ้งเตือนการเกิดอุบัติเหตุระหว่างรถยนต์ผ่าน Wi-Fi เพื่อการขับขี่ที่ปลอดภัยจะทำงานเมื่อระบบได้รับสัญญาณเตือนความผิดปกติของรถยนต์จากระบบเซ็นเซอร์ ซึ่งติดตั้งอยู่รอบและภายในรถยนต์ อุบัติเหตุในงานวิจัยนี้คือ ความผิดปกติที่เกิดขึ้นกับรถยนต์ ซึ่งสามารถนำไปสู่ความไม่ปลอดภัยในการขับขี่ โดยระบบเซ็นเซอร์สามารถประกอบไปด้วย เซ็นเซอร์ตรวจจับการชน (Hit Sensor) เซ็นเซอร์ตรวจจับความสมดุล (Balance Sensor) และเซ็นเซอร์ตรวจจับอุณหภูมิ (Thermal Sensor) เป็นต้น เซ็นเซอร์เหล่านี้จะทำงานเมื่อเกิดการกระตุ้น เช่น รถชน รถยนต์เสียความสมดุล หรืออุณหภูมิในรถยนต์สูงผิดปกติ เซ็นเซอร์ที่ถูกกระตุ้นจะส่งข้อมูลไปยังอุปกรณ์ไมโครคอนโทรลเลอร์ เพื่อประเมินความผิดปกติของรถยนต์ ถ้าระบบประเมินว่าสถานะของรถยนต์ผิดปกติ ระบบจะส่งสัญญาณแจ้งเตือนไปยังโทรศัพท์เคลื่อนที่ผ่านการเชื่อมต่อทางบลูทูธ โทรศัพท์เคลื่อนที่ที่จะประมวลผลสัญญาณเตือน และ broadcast ข้อความแจ้งเตือนการเกิดอุบัติเหตุ พร้อมด้วยตำแหน่งของการเกิดอุบัติเหตุ ซึ่งได้จากระบบระบุตำแหน่ง (GPS) ไปยังรถยนต์ข้างเคียงผ่านการเชื่อมต่อทาง Wi-Fi เพื่อเตือนให้ผู้ขับขี่รอบข้างเพิ่มความระมัดระวังในการขับขี่ หรือเลี้ยวไปใช้เส้นทางอื่นๆที่มีความปลอดภัยสูงกว่า เป็นต้น นอกจากนี้ระบบยังสามารถส่งข้อความ SMS แจ้งเตือนไปยังบุคคลที่เกี่ยวข้องที่ไม่อยู่ในพื้นที่การเกิดอุบัติเหตุ เช่น ญาติ หมอหรือพยาบาล พนักงานประกันภัย และ

ตำรวจ เป็นต้น ผ่านทางเครือข่ายโทรศัพท์เคลื่อนที่ เพื่อช่วยให้ผู้ประสบภัยได้รับการช่วยเหลือได้ทันการ และในกรณีที่แย่ที่สุดผู้ประสบภัยอาจหมดสติก่อนที่จะสามารถแจ้งเหตุไปยังบุคคลเหล่านี้ ระบบนี้จึงเป็นอีกช่องทางหนึ่งที่จะช่วยให้การแจ้งเหตุมีประสิทธิภาพมากขึ้น ดังแสดงในรูปที่ 3.1

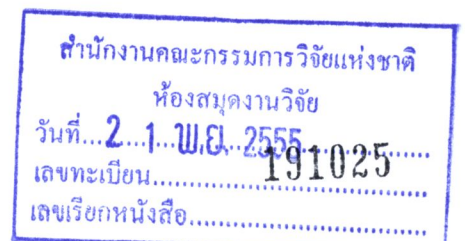
3.2 การออกแบบและพัฒนาระบบติดต่อสื่อสารและแจ้งเตือนการเกิดอุบัติเหตุระหว่างรถยนต์ผ่าน Wi-Fi เพื่อการขับขี่ที่ปลอดภัย

ในการพัฒนาระบบติดต่อสื่อสารและแจ้งเตือนการเกิดอุบัติเหตุระหว่างรถยนต์ผ่าน Wi-Fi เพื่อการขับขี่ที่ปลอดภัยนั้น สามารถแบ่งขั้นตอนการดำเนินงานได้ 5 ขั้นตอนหลัก แสดงในรูปที่ 3.2 คือ

1. การพัฒนาระบบเชื่อมต่อระหว่างโทรศัพท์เคลื่อนที่กับอุปกรณ์ไมโครคอนโทรลเลอร์ผ่านบลูทูธ เพื่อการรับส่งสัญญาณแจ้งเตือนการเกิดอุบัติเหตุระหว่างไมโครคอนโทรลเลอร์และโทรศัพท์เคลื่อนที่
2. การพัฒนาระบบระบุตำแหน่งของรถยนต์โดยใช้เทคโนโลยีจีพีเอส เพื่อใช้ระบุตำแหน่งของการเกิดอุบัติเหตุ ซึ่งเป็นส่วนหนึ่งของข้อความแจ้งเตือนซึ่งส่งไปยังรถยนต์คันอื่นๆ
3. การพัฒนาระบบติดต่อสื่อสารระหว่างรถยนต์ผ่าน Wi-Fi เพื่อเตือนรถยนต์รอบข้าง เพื่อใช้ส่งข้อความแจ้งเตือนจากรถยนต์คันที่ประสบอุบัติเหตุกับรถยนต์รอบข้าง
4. การพัฒนาระบบแจ้งเหตุอัตโนมัติในรูปแบบ SMS ผ่านเครือข่ายโทรศัพท์เคลื่อนที่ เพื่อแจ้งเหตุแก่บุคคลที่เกี่ยวข้อง ซึ่งไม่ได้อยู่ในพื้นที่การเกิดอุบัติเหตุ
5. ระบบประมวลผลและติดต่อกับผู้ใช้งาน



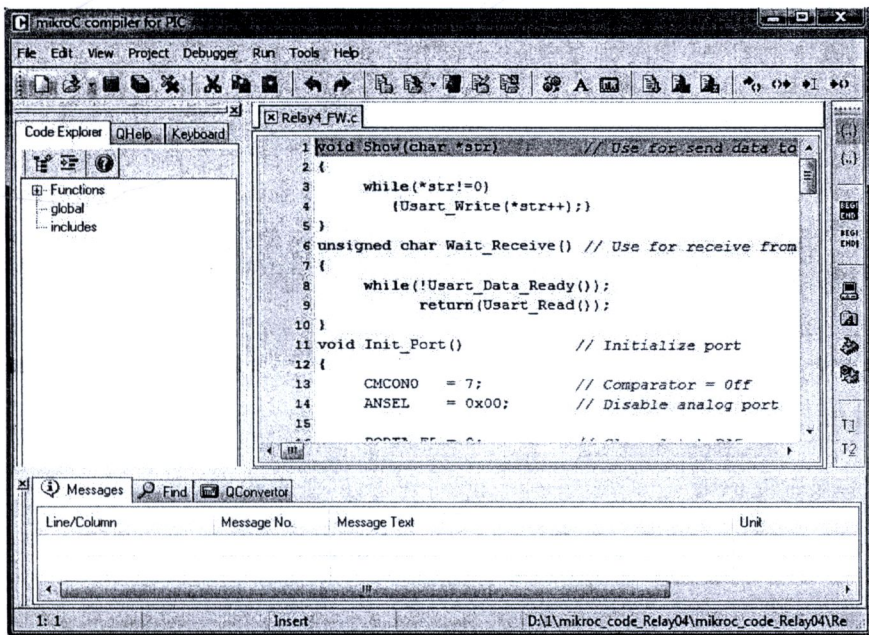
รูปที่ 3.2 องค์ประกอบของระบบติดต่อสื่อสารและแจ้งเตือนการเกิดอุบัติเหตุระหว่างรถยนต์ผ่าน Wi-Fi



3.2.1 การพัฒนาระบบเชื่อมต่อระหว่างเครื่องโทรศัพท์เคลื่อนที่กับอุปกรณ์ไมโครคอนโทรลเลอร์ผ่านบลูทูธ

เนื่องจากโทรศัพท์เคลื่อนที่ในปัจจุบันมีช่องทางเชื่อมต่อ (Port) กับอุปกรณ์ภายนอกจำกัด ทำให้การติดต่อสื่อสารระหว่างโทรศัพท์เคลื่อนที่กับอุปกรณ์ไมโครคอนโทรลเลอร์ทำได้ค่อนข้างยาก วิธีการหนึ่งที่เป็นไปได้คือการเชื่อมต่อผ่านบลูทูธ โดยมีขั้นตอนต่างๆ ดังนี้

- 1) พัฒนาโปรแกรมการรับส่งข้อมูลระหว่างโทรศัพท์เคลื่อนที่และอุปกรณ์ไมโครคอนโทรลเลอร์ผ่านบลูทูธ โดยใช้ภาษาซี (C Language) บนไมโครคอนโทรลเลอร์ (PIC 16F688) ผ่านโปรแกรม mikroC v8.1 ดังรูปที่ 3.3



รูปที่ 3.3 ตัวอย่างหน้าจอการพัฒนาโปรแกรม

รูปที่ 3.4 แสดงตัวอย่างของส่วนของโปรแกรมที่พัฒนาขึ้นเพื่อรับข้อมูลจากเซนเซอร์สัมผัสเพื่อตรวจสอบการชนและส่งข้อมูลผ่านบลูทูธไปยัง Terminal ของเครื่องคอมพิวเตอร์ โดยไมโครคอนโทรลเลอร์จะส่งค่า 1 ไปยังโทรศัพท์เคลื่อนที่ทุก 100 ms อย่างต่อเนื่องเมื่อระบบเซ็นเซอร์ทำงานโดยการถูกกดจากการชนกันของรถยนต์ (กรณีที่ 1 และ 2) และจะส่งค่า 0 ในสภาวะการปกติ (กรณีที่ 3)

ค่าช่องสัญญาณในส่วนของ Terminal เป็น COM40 ถ้าเชื่อมต่อสมบูร์ณไฟฟ้ที่อยู่ใน CTS (Clear-to-Send) กับ DSR (Data Set Ready) ที่อยู่ในส่วนของ Status จะเปลี่ยนจากสีแดงเป็นสีเขียว ซึ่งแสดงว่าระบบอยู่ในสถานะที่พร้อมที่จะรับและส่งข้อมูล ดังรูปที่ 3.5

```

void main()
{
    char cmd,val[5], *str= "Reset!";
    Init_Port();
    Usart_Init(9600);
    while(1)
    {
        if(!PORTA.F0)
        {
            Usart_Write('1');
            while(!PORTA.F0);
        }
        else if(!PORTA.F1)
        {
            Usart_Write('1');
            while(!PORTA.F1);
        }
        else
        {
            Usart_Write('0');
        }
        delay_ms(100);
    }
}

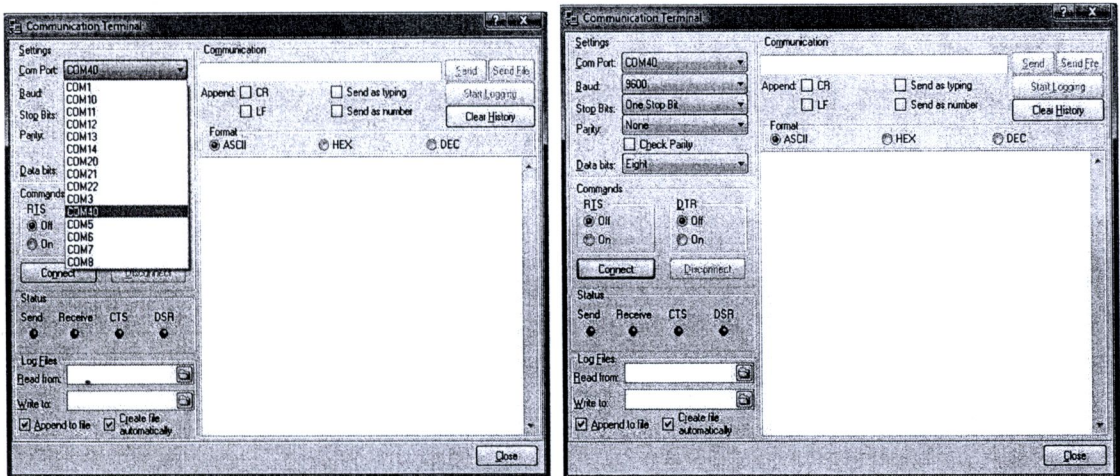
```

1

2

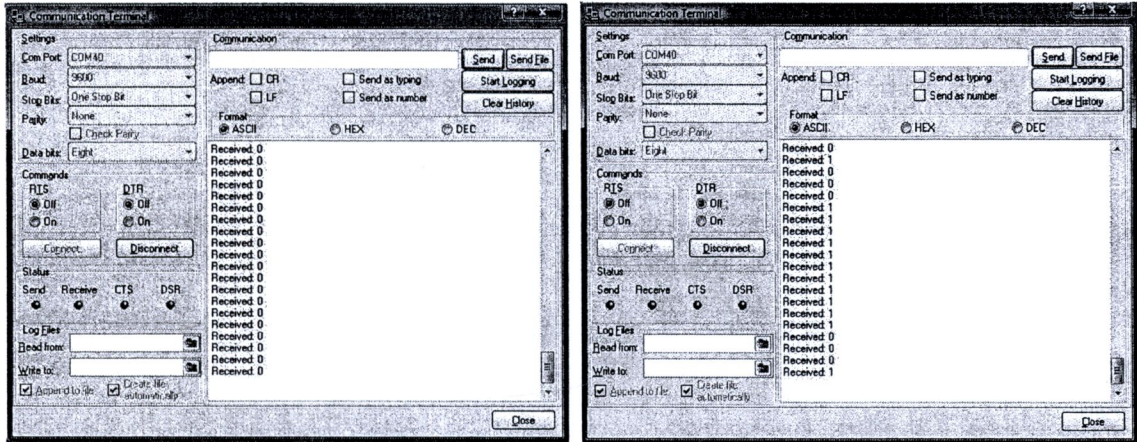
3

รูปที่ 3.4 ส่วนของโปรแกรมรับข้อมูลจากเซนเซอร์ เพื่อตรวจสอบการชนและส่งข้อมูลผ่าน Terminal ของ
เครื่องคอมพิวเตอร์



รูปที่ 3.5 การเชื่อมต่อระหว่างคอมพิวเตอร์ กับไมโครคอนโทรลเลอร์ผ่านบลูทูธ

จากนั้นไมโครคอนโทรลเลอร์จะส่งค่า 0 มาตลอดจนกว่าจะถูกชนโปรแกรมจึงจะส่งค่า 1 ดังรูปที่ 3.6



รูปที่ 3.6 การทดสอบการส่งข้อมูลผ่าน Terminal ของเครื่องคอมพิวเตอร์

- 2) พัฒนาโปรแกรมส่วนจัดการการเชื่อมต่อระหว่างโทรศัพท์เคลื่อนที่กับอุปกรณ์ไมโครคอนโทรลเลอร์ผ่านบลูทูธ โดยการเปิดพอร์ตสื่อสาร ในการเขียนโปรแกรมนั้นใช้ SerialPort ในการติดต่อสื่อสารกันระหว่างอุปกรณ์ รูปที่ 3.7 แสดงคำสั่งเปิดช่องสื่อสารของบลูทูธบนโทรศัพท์เคลื่อนที่ เพื่อรับค่าจากไมโครคอนโทรลเลอร์

```
private void SerialPort1_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
    string st = SerialPort1.ReadExisting();
    BeginInvoke(delegate { Show(st); });
}
```

รูปที่ 3.7 คำสั่งเปิดช่องสื่อสารของบลูทูธบนโทรศัพท์เคลื่อนที่

รูปที่ 3.8 แสดงตัวอย่างของการพัฒนาโปรแกรมการเชื่อมต่อบลูทูธระหว่างโทรศัพท์เคลื่อนที่และไมโครคอนโทรลเลอร์ โดยถ้าเชื่อมต่อสำเร็จ จะขึ้นคำว่า Connected เป็นสีเขียว หากการเชื่อมต่อผิดพลาด จะขึ้นคำว่า Not Connect เป็นสีแดง

```

private void btbluetooth_Click(object sender, EventArgs e)
{
    if (btbluetooth.Text == "Connect")
    {
        SerialPort1.PortName = CmbPort.Text;
        SerialPort1.Open();
        CmbPort.Enabled = false;

        btbluetooth.Text = "Disconnect";
        lblPortStatus.Text = "Connected";
        lblPortStatus.ForeColor = Color.Lime;

        SerialPort1.Write("R");
    }
    else
    {
        SerialPort1.Close();
        CmbPort.Enabled = true;

        btbluetooth.Text = "Connect";
        lblPortStatus.Text = "Not Connect";
        lblPortStatus.ForeColor = Color.Red;
    }
}

```

รูปที่ 3.8 ตัวอย่างคำสั่งในการเชื่อมต่อบลูทูธ พร้อมแสดงสถานะการเชื่อมต่อ

หน้าจอของโทรศัพท์เคลื่อนที่จะแสดงค่าที่ได้รับจาก ไมโครคอนโทรลเลอร์ซึ่งมีค่าเป็น 0 หรือ 1 ผ่านคำสั่งในรูปที่ 3.9

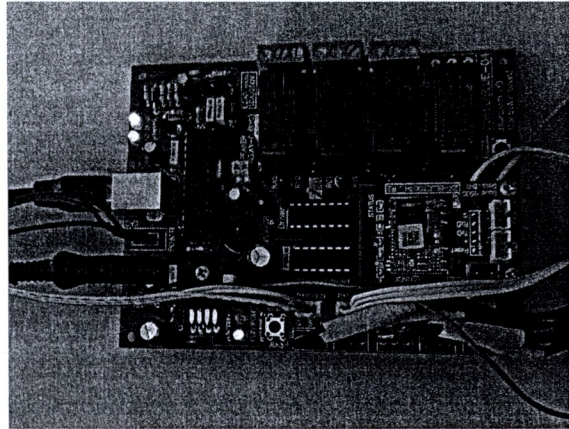
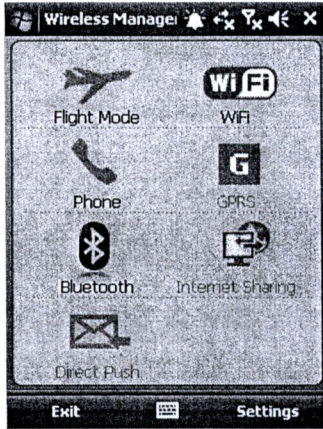
```

private void ShowReceive(string sr)
{
    lblResponse.Text = sr;
}

```

รูปที่ 3.9 ส่วนของโปรแกรมแสดงค่าที่รับได้จากไมโครคอนโทรลเลอร์

- 3) เปิดการใช้งานบลูทูธของโทรศัพท์เคลื่อนที่ (ในตัวอย่างครั้งนี้ใช้โทรศัพท์เคลื่อนที่ ASUS รุ่น P552w) กับอุปกรณ์ไมโครคอนโทรลเลอร์ ดังรูปที่ 3.10



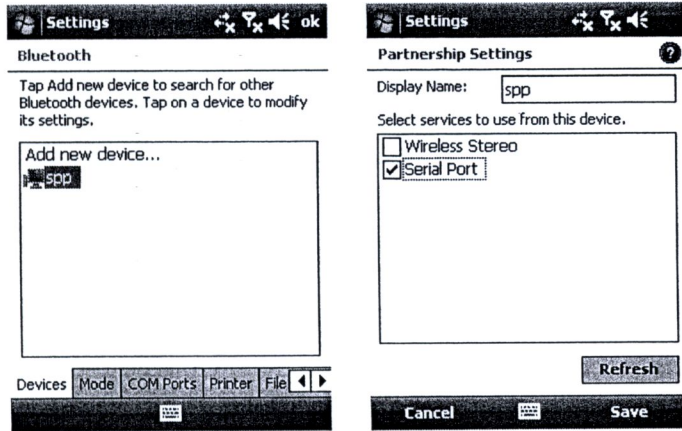
รูปที่ 3.10 การเปิดการใช้งานบลูทูธของโทรศัพท์เคลื่อนที่

- 4) เพิ่มรายชื่ออุปกรณ์ไมโครคอนโทรลเลอร์ที่จะเชื่อมต่อกับโทรศัพท์เคลื่อนที่ โดยเลือกที่ไอคอน Bluetooth จากนั้นเลือกแท็บ Devices ไปที่ Add new device ใส่ชื่ออุปกรณ์ไมโครคอนโทรลเลอร์ ในตัวอย่างนี้กำหนดให้ชื่อ spp จากนั้นกดปุ่ม Next เพื่อใส่ค่า Passcode ที่ใช้เป็นรหัสลับในการติดต่อสื่อสารระหว่างอุปกรณ์ทั้ง 2 จากนั้นกด Next ดังแสดงในรูปที่ 3.11



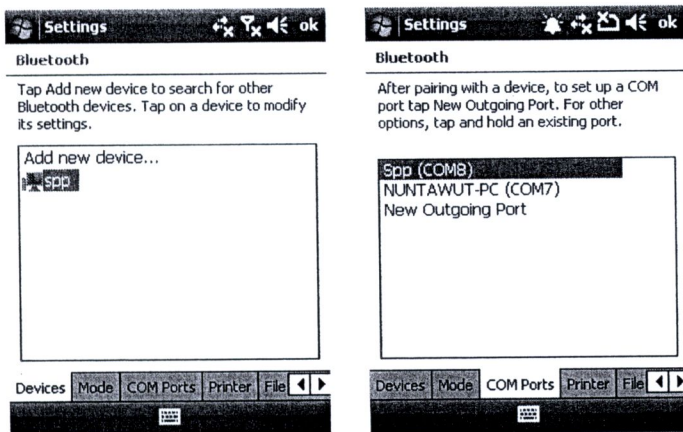
รูปที่ 3.11 การเพิ่มรายชื่ออุปกรณ์ไมโครคอนโทรลเลอร์

- 5) กำหนดค่าเบื้องต้นของการเชื่อมต่อเป็นแบบ Serial Port โดยการเลือกที่ไอคอน spp และ เลือกที่ Serial Port และกดปุ่ม Save ดังแสดงในรูปที่ 3.12



รูปที่ 3.12 การกำหนดค่าเบื้องต้นของการเชื่อมต่อเป็นแบบ Serial Port

- 6) ตรวจสอบค่าเบื้องต้นของเลขของช่องสัญญาณเชื่อมต่อ (Port Number) โดยการเลือกที่แท็บ COM Ports จากนั้นดูที่ spp ว่าเชื่อมต่ออยู่ที่ช่องสัญญาณใด (จากรูปที่ 3.13 เป็นค่าการเชื่อมต่อที่ช่องสัญญาณ COM8)



รูปที่ 3.13 การตรวจสอบค่าเบื้องต้นของเลขของช่องสัญญาณเชื่อมต่อ

- 7) เชื่อมต่อโทรศัพท์เคลื่อนที่กับอุปกรณ์ไมโครคอนโทรลเลอร์ โดยเปิดโปรแกรมระบบติดต่อสื่อสารและแจ้งเตือนการเกิดอุบัติเหตุระหว่างรถยนต์ผ่าน Wi-Fi ที่พัฒนาขึ้น โดยเลือกที่เมนู Set Bluetooth เลือกการเชื่อมต่อเป็น COM8 และกดปุ่ม Connect ถ้าการเชื่อมต่อเรียบร้อย โปรแกรมจะแสดงข้อความ Connected ดังรูปที่ 3.14



รูปที่ 3.14 การเชื่อมต่อโทรศัพท์เคลื่อนที่กับอุปกรณ์ไมโครคอนโทรลเลอร์ผ่านบลูทูธ

3.2.2 การพัฒนาระบบระบุตำแหน่งของรถยนต์โดยใช้เทคโนโลยีจีพีเอส

ส่วนของการพัฒนาโปรแกรมระบุตำแหน่งของรถยนต์โดยใช้เทคโนโลยีจีพีเอสนั้น ทางผู้พัฒนาได้ใช้โปรแกรม Open Source มาพัฒนาต่อยอดเพื่อรับค่าตำแหน่งรถยนต์ ณ เวลาต่างๆ อย่างต่อเนื่อง โดยรูปที่ 3.15 แสดงส่วนของโปรแกรมที่ใช้รับค่าตำแหน่งของรถยนต์ในรูปแบบของค่าละติจูด และค่าลองจิจูด

```
gpsPosition = new GpsPosition();
gpsPosition.dwVersion = 1;
gpsPosition.dwSize = Marshal.SizeOf(typeof(GpsPosition));

// Marshal our data to the native pointer we allocated.
Marshal.StructureToPtr(gpsPosition, ptr, false);

// call native method passing in our native buffer
int result = GPSGetPosition(gpsHandle, ptr, 500000, 0);
if (result == 0)
{
    // native call succeeded, marshal native data to our managed data
    gpsPosition = (GpsPosition)Marshal.PtrToStructure(ptr, typeof(GpsPosition));
}
```

รูปที่ 3.15 ส่วนของโปรแกรมที่ใช้รับค่าตำแหน่งของรถยนต์

3.2.3 การพัฒนาระบบติดต่อสื่อสารระหว่างรถยนต์ผ่าน Wi-Fi เพื่อเตือนรถยนต์รอบข้าง

การพัฒนาระบบติดต่อสื่อสารระหว่างรถยนต์ผ่าน Wi-Fi เพื่อเตือนรถยนต์รอบข้างนั้นสามารถแบ่งออกได้เป็น 2 ส่วน คือ การพัฒนาโปรแกรมฝั่งผู้ส่ง (รถยนต์คันที่เกิดอุบัติเหตุ) และการพัฒนาโปรแกรมฝั่งผู้รับ (รถยนต์รอบข้าง)

- 1) การพัฒนาโปรแกรมฝั่งผู้ส่ง จากรูป 3.16 เป็นการกำหนดเลขไอพี (IP Address) ปลายทางของการติดต่อสื่อสารเป็นค่า 169.254.255.255 ซึ่งเป็นเลขไอพีของการส่งข้อมูลแบบบรอดแคสต์ (Broadcast) ไปยังรยยนต์ทุกคันที่อยู่ในเครือข่ายวง 169.254.0.0 เดียวกัน ในการพัฒนาในอนาคตอาจปรับปรุงการติดต่อเป็นแบบจีโอแคสต์ (Geocast) เพื่อส่งข้อมูลไปยังรยยนต์ในพื้นที่ที่เกี่ยวข้องเท่านั้น

```
Socket sock;
Thread receiver;
private Button button1;
IPEndPoint multiep = new IPEndPoint(IPAddress.Parse("169.254.255.255"), 9050);
```

รูปที่ 3.16 ส่วนของโปรแกรมกำหนดเลขไอพีของผู้รับข้อมูล

เนื่องจากการติดต่อสื่อสารระหว่างรยยนต์นี้มุ่งเน้นด้านความเร็ว ประกอบกับข้อมูลที่ส่งนั้นมีขนาดเล็กมาก การส่งข้อมูลแบบ UDP (User Datagram Protocol) จึงเป็นวิธีการที่เหมาะสมที่สุด จากรูปที่ 3.17 แสดงการเลือกรูปแบบการส่งข้อมูลเป็นแบบ UDP

```
public FrmMain()
{
    InitializeComponent();
    sock = new Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);
    IPEndPoint iep = new IPEndPoint(IPAddress.Any, 9050);
    sock.Bind(iep);
    receiver = new Thread(new ThreadStart(packetReceive));
    receiver.IsBackground = true;
    receiver.Start();
}
```

รูปที่ 3.17 ส่วนของโปรแกรมกำหนดรูปแบบการส่งข้อมูลเป็นแบบ UDP

- 2) การพัฒนาโปรแกรมฝั่งผู้รับ เมื่อผู้รับได้รับข้อความจากรยยนต์คันที่เกิดอุบัติเหตุ โปรแกรมจะแยกค่าตำแหน่งละติจูดและลองจิจูดออกมาจากข้อความที่รับได้ เพื่อนำไปใช้คำนวณค่าระยะห่างจากจุดเกิดเหตุและแจ้งเตือนผู้ขับขี่ให้เพิ่มความระมัดระวังต่อไป

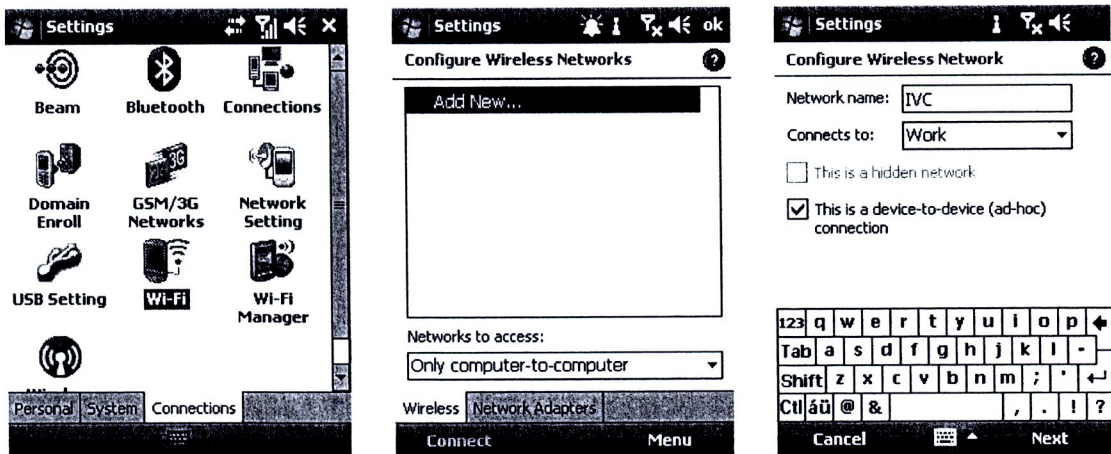
```

void packetReceive()
{
    EndPoint ep = (EndPoint)multiiep;
    byte[] data = new byte[1024];
    string stringData1;
    int recv;
    string path = @"MyDir";
    recv = sock.ReceiveFrom(data, ref ep);
    stringData1 = System.Text.Encoding.ASCII.GetString(data, 0, recv);
    if (File.Exists(@"MyDir\txt1.txt") == false)
    {
        Directory.CreateDirectory(path);
        StreamWriter sw2 = File.CreateText(@"MyDir\txt1.txt");
        sw2.WriteLine(stringData1);
        sw2.Close();
        sw2 = null;
    }
    else
    {
    }
}

```

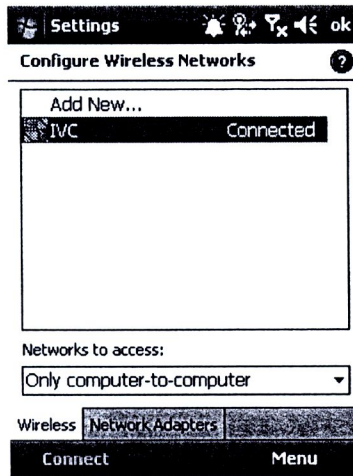
รูปที่ 3.18 ส่วนของโปรแกรมแยกค่าตำแหน่งละติจูดและลองจิจูดออกมาจากข้อความที่ฝั่งผู้รับ

โดยการเชื่อมต่อ Wi-Fi แบบ Ad-Hoc ซึ่งเป็นการเชื่อมต่อโดยไม่ใช้ Wireless Access Point มีขั้นตอนคือ เปิดการทำงานของ Wi-Fi ของโทรศัพท์เคลื่อนที่ หลังจากนั้นทำการเพิ่มชื่อเครือข่ายใหม่ โดยเลือกไอคอน Wi-Fi ทำการตั้งค่า Networks to access เป็น Only computer-to-computer จากนั้นคลิกที่ Add New ที่ช่อง Network name ตั้งชื่อเครือข่ายใหม่ ในตัวอย่างตั้งชื่อเครือข่ายใหม่เป็น “IVC” เลือกที่ This is a device-to-device (Ad-hoc) จากนั้นคลิกปุ่ม Next และปุ่ม Finish ตามลำดับ ดังรูปที่ 3.19



รูปที่ 3.19 การเพิ่มชื่อเครือข่ายใหม่

จากนั้นผู้ใช้งานสามารถเชื่อมต่อ Wi-Fi ได้โดยคลิกที่ IVC และเลือก Connect เมื่อเชื่อมต่อสำเร็จ หน้าจอจะแสดงคำว่า Connected ดังรูปที่ 3.20



รูปที่ 3.20 การเชื่อมต่อ Wi-Fi

3.2.4 การพัฒนาระบบแจ้งเหตุอัตโนมัติในรูปแบบ SMS ผ่านเครือข่ายโทรศัพท์เคลื่อนที่

นอกจากจะส่งข้อความไปยังรถยนต์บริเวณใกล้เคียงจุดการเกิดอุบัติเหตุ ระบบยังมีความสามารถในการส่งข้อความสั้นหรือ SMS ผ่านเครือข่ายโทรศัพท์เคลื่อนที่ไปยังบุคคลอื่นๆ เช่น ญาติของผู้ประสบอุบัติเหตุ ตำรวจ หรือบริษัทประกันภัย เพื่อให้การช่วยเหลือผู้ประสบภัยได้ทันต่อเหตุการณ์ ดังรูปที่ 3.21

```
public void smssend()
{
    for (i1 = 0; i1 < txtsms.Items.Count; i1++)
    {
        Interceptor = null;
        Interceptor = new MessageInterceptor(InterceptionAction.NotifyAndDelete, false);
        Interceptor.MessageReceived += new MessageInterceptorEventHandler(Interceptor_MessageReceived);
        if (Label9.Text == "ตำแหน่ง")
        {
            SmsMessage OutgoingMessage = new SmsMessage(txtsms.Items[i1].ToString(), txtCarID.Text +
                " เกิดอุบัติเหตุ ที่ตำแหน่ง ละติจูด : " + TextLat1.Text + " Long : " + TextLong1.Text);
            OutgoingMessage.Send();
        }
        else if (Label9.Text == "Position")
        {
            SmsMessage OutgoingMessage = new SmsMessage(txtsms.Items[i1].ToString(), txtCarID.Text +
                " has an accident at lat : " + TextLat1.Text + " Long : " + TextLong1.Text);
            OutgoingMessage.Send();
        }
        Interceptor.MessageReceived -= Interceptor_MessageReceived;
        Interceptor.Dispose();
    }
}
```

รูปที่ 3.21 ส่วนของโปรแกรมที่ทำหน้าที่ส่ง SMS

3.2.5 ระบบประมวลผลและติดต่อกับผู้ใช้งาน

ประมวลผลและติดต่อกับผู้ใช้งานสามารถแบ่งออกได้เป็นส่วนย่อยๆ ดังนี้

- 1) ส่วนการคำนวณค่าระยะห่างจากจุดเกิดอุบัติเหตุ รูปที่ 3.22 แสดงส่วนของโปรแกรมที่ใช้คำนวณระยะห่างจากจุดเกิดอุบัติเหตุ โดยอ้างอิงจากตำแหน่งของรถยนต์ที่เกิดอุบัติเหตุ กับตำแหน่ง (ค่าละติจูด และค่าลองจิจูด) ของรถยนต์ตนเอง

```
private double calkilo(double lat1, double long1, double lat2, double long2)
{
    const double RadiusEarth = 6371.10;
    return (RadiusEarth * ((2 * Math.Asin(Math.Sqrt(Math.Pow(Math.Sin((RADIANS(lat1) - RADIANS(lat2)) / 2), 2)
    + Math.Cos(RADIANS(lat1)) * Math.Cos(RADIANS(lat2))
    * Math.Pow(Math.Sin((RADIANS(long1) - RADIANS(long2)) / 2), 2))))));
}

public static double RADIANS(double Degrees)
{
    return Degrees * (Math.PI / 180);
}
```

รูปที่ 3.22 ส่วนของโปรแกรมที่ทำหน้าที่คำนวณค่าระยะห่างจากจุดเกิดอุบัติเหตุ

- 2) ส่วนการเก็บหมายเลขทะเบียนรถ จะทำหน้าที่เก็บข้อมูลหมายเลขทะเบียนรถไว้ในไฟล์ txtCarID.ini ซึ่งไฟล์นี้ถูกสร้างขึ้นในฟังก์ชัน caridlist() และสามารถแก้ไขได้โดยผู้ใช้งาน แสดงในรูปที่ 3.23

```
private void CarIDSave_Click(object sender, EventArgs e)
{
    Directory.Delete(@"MyCarID", true);
    string path = @"MyCarID";
    if (File.Exists(@"MyCarID\txtCarID.ini") == false)
    {
        Directory.CreateDirectory(path);
        StreamWriter sw2 = File.CreateText(@"MyCarID\txtCarID.ini");
        sw2.WriteLine(txtCarID.Text);
        sw2.Close();
        sw2 = null;
    }
    caridlist();
    panel2.Visible = false;
}
```

รูปที่ 3.23 ส่วนของโปรแกรมที่ทำหน้าที่บันทึกหมายเลขทะเบียนรถ

- 3) ส่วนการบันทึกชื่อและหมายเลขโทรศัพท์ เป็นการบันทึกหมายเลขโทรศัพท์ เพื่อใช้ในการส่งข้อความสั้น หรือ SMS โดยผู้ใช้งานสามารถเพิ่ม ลบ หรือ แก้ไขหมายเลขโทรศัพท์ได้ตามต้องการ โดยหมายเลขโทรศัพท์จะถูกบันทึกในไฟล์ txtPhonebook .ini ดังรูปที่ 3.23

```
private void btsetpbook_Click(object sender, EventArgs e)
{
    Directory.Delete(@"MyPhonebook", true);
    string path = @"MyPhonebook";
    if (File.Exists(@"MyPhonebook\txtphonebook.ini") == false)
    {
        Directory.CreateDirectory(path);
        StreamWriter sw2 = File.CreateText(@"MyPhonebook\txtphonebook.ini");
        for (i1 = 0; i1 < lstnumber.Items.Count; i1++)
        {
            sw2.WriteLine(lstnumber.Items[i1].ToString());
        }
        sw2.Close();
        sw2 = null;
    }
    phonebooklist();
    panel1.Visible = false;
}
```

รูปที่ 3.24 ส่วนของโปรแกรมที่ทำหน้าที่บันทึกชื่อและหมายเลขโทรศัพท์

- 4) ส่วนการเลือกภาษาไทย/อังกฤษ โปรแกรมสามารถทำงานได้ 2 ภาษา คือ ภาษาไทยและภาษาอังกฤษ เพื่อให้ทั้งชาวต่างชาติและชาวไทย สามารถใช้โปรแกรมนี้ได้ รูปที่ 3.25 แสดงส่วนของโปรแกรมที่ใช้เลือกการทำงานในโหมดภาษาต่างๆ

```
private void PicFthai_Click(object sender, EventArgs e)
{
    Label19.Text = "ตำแหน่ง";
    Label17.Text = "ระยะทาง";
    Label13.Text = "เบอร์โทรศัพท์";
    Label11.Text = "พอร์ต";
    Label12.Text = "ทะเบียนรถ";
    Label14.Text = "เพิ่มเบอร์โทร";
    Label16.Text = "รายชื่อ";
    Label15.Text = "กดในรายการเพื่อลบเบอร์ออก";
    startGpsMenuItem.Text = "จีพีเอสเริ่มทำงาน";
    stopGpsMenuItem.Text = "หยุดการทำงาน";
    phonebookMenuItem.Text = "สมุดโทรศัพท์";
    CarIdMenuItem.Text = "ทะเบียนรถ";
    HideShowMenuItem.Text = "แสดง/ไม่แสดง";
    BluetoothMenuItem.Text = "ตั้งค่าบลูทูธ";
}

private void picFeng_Click(object sender, EventArgs e)
{
    Label19.Text = "Position";
    Label17.Text = "Range";
    Label13.Text = "Mobile No.";
    Label11.Text = "Port.";
    Label12.Text = "Car ID";
    Label14.Text = "Add Number";
    Label16.Text = "Add Name";
    Label15.Text = "Click list to Delete Number";
    startGpsMenuItem.Text = "Start GPS";
    stopGpsMenuItem.Text = "Stop GPS";
    phonebookMenuItem.Text = "Phone Book";
    CarIdMenuItem.Text = "Car ID";
    HideShowMenuItem.Text = "Hind/Show";
    BluetoothMenuItem.Text = "Set Bluetooth";
}
```

รูปที่ 3.25 ส่วนของโปรแกรมที่ใช้เลือกการทำงานในโหมดภาษาไทย/อังกฤษ