



Chapter 6: Experiment on VANET

- Collision Warning System

Prof. **Yuh-Shyan Chen** (陳裕賢)

Department of Computer Science and
Information Engineering
National Taipei University

National Taipei University



Outline



- 實驗室
- 實驗目的
- 實驗方法
- 實驗成果
- 三個實驗場景

車載資通訊網路與安全實驗室



車載資通訊網路與安全 研究室

Telematics Network and
Security Research Group

工作實驗室



實驗目的

- 藉由 VANET (Vehicular Ad-Hoc Network 車用行動通訊網路) 之特性，設計出一套車間駕駛人行車安全警示系統。
- 藉由以下判斷決定是否發送警告訊息給周圍車輛，以達到警示效果。
 - 駕駛人頭部姿勢。
 - 油門、煞車的操作狀況。
 - 方向盤是否急轉與否。

實驗目的

- 此實驗系統具有：
 - 精確的預警判斷機制。
 - 高效的車間通訊。
 - 即時的地圖資訊。
 - 人性化的介面設計。

實驗方法

- 系統介面結構：
 - 以「人性化操作」、「資訊醒目呈現」、「樣式美觀」三大原則為基礎設計出符合該需求的程式介面。

GPS定位點

利用GPS接收器可告知駕駛本車與其他車輛位置。

地圖圖資

使用Google map api 提供駕駛圖資資訊。

警告訊息

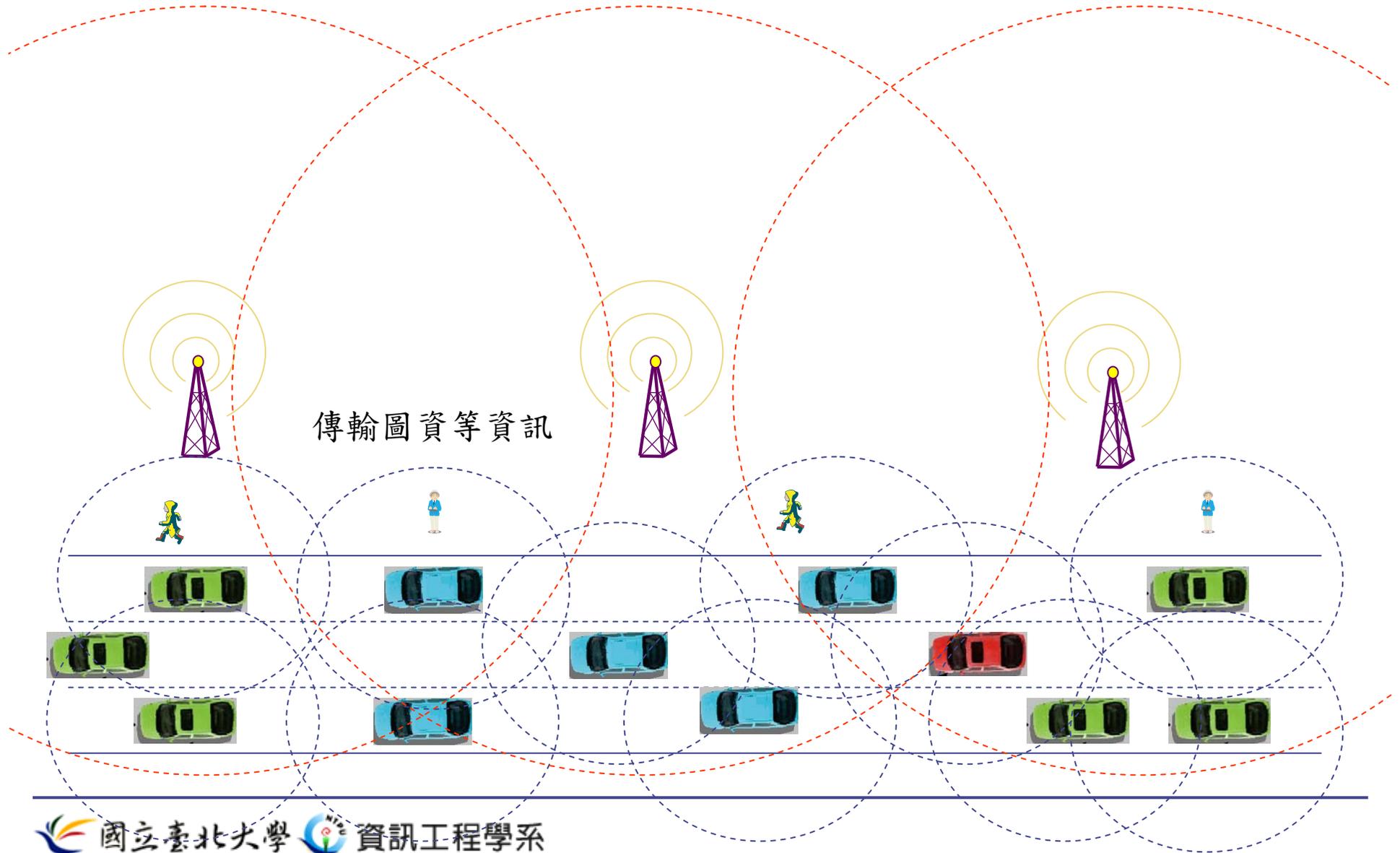
實驗成果

- 駕駛人可以透過行車安全警示系統清楚的得知周邊車輛的狀態，在對方急轉、急煞時可以更快做出及時的反應。
- 因為系統的警告，可以保護駕駛人不在打瞌睡等注意力不集中的狀態下駕駛，降低車禍事故的發生。
- 結合GPS衛星定位系統，駕駛人會很清楚的知道鄰近的車輛狀況，像是在高速公路上，就能避免因高速追撞而造成的連環車禍。
- 行車安全警示系統將在未來帶給人們一個更安全、順暢的交通狀態。

系統運作場景

- 本系統運作場景假設為在高速公路上
- 裝置及配備
 - For 車輛
 - 顯示警告訊息的UMPC一台
 - 感測器四個（耳部、方向盤、方向桿以及煞車踏板）
 - GPS接收器一顆
 - For 道路
 - AP數台（每隔一段距離放置一座，用以發送地圖資訊、重要公告等相關訊息）

系統場景



系統介面



Form1

地圖資訊

地圖 衛星 混合地圖

福爾摩沙高速公路 隆恩路

重新定位

POWERED BY Google

↑ ↓ ← → + - 使用條款

綜合狀態訊息
開始接收資料...!!

請小心駕駛

Edit1
Set
Send

離開

✓ 狀態解除

各類警告圖示



三個實驗場景

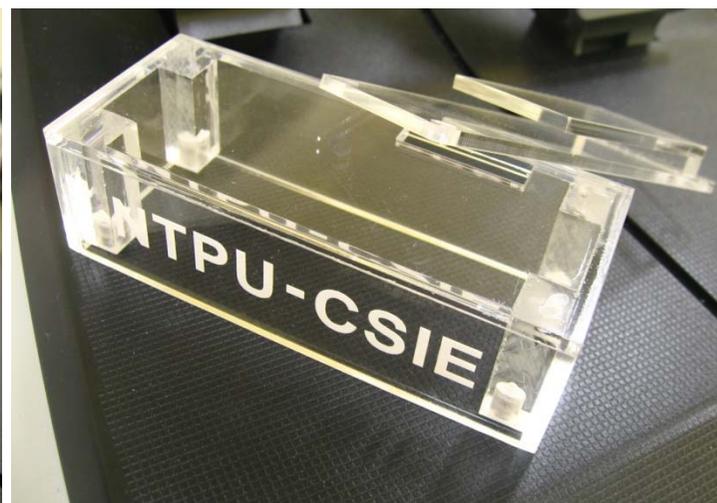
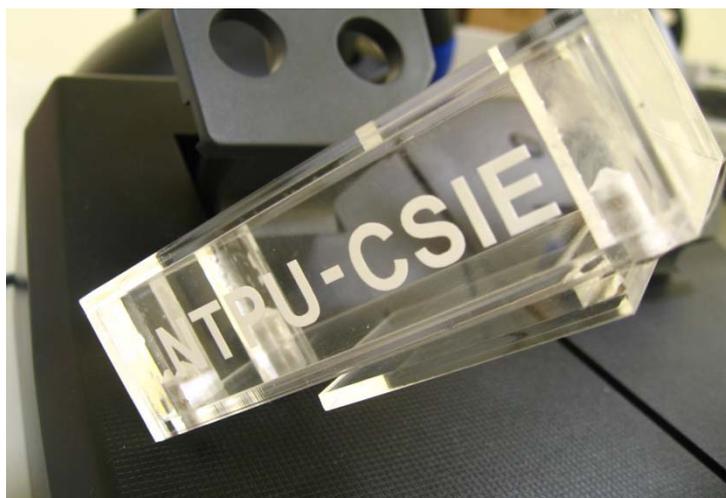
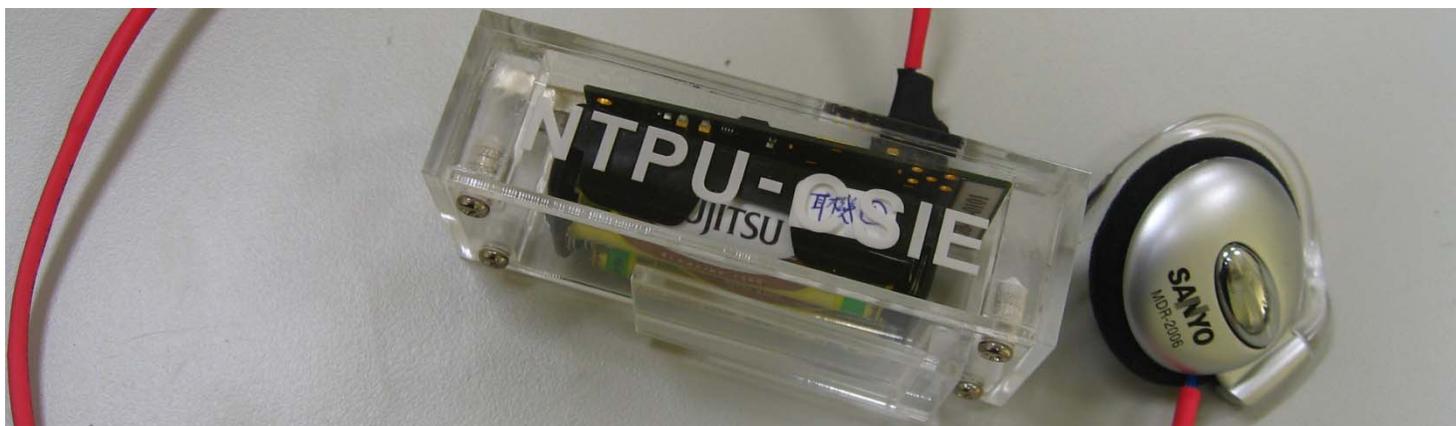
1. 駕駛人姿勢不良 (Driver-Status Warning)
2. 車輛改道 (Lane-Change Warning)
3. 緊急煞車 (Emergency-Braking Warning)

場景一：Driver-Status Warning

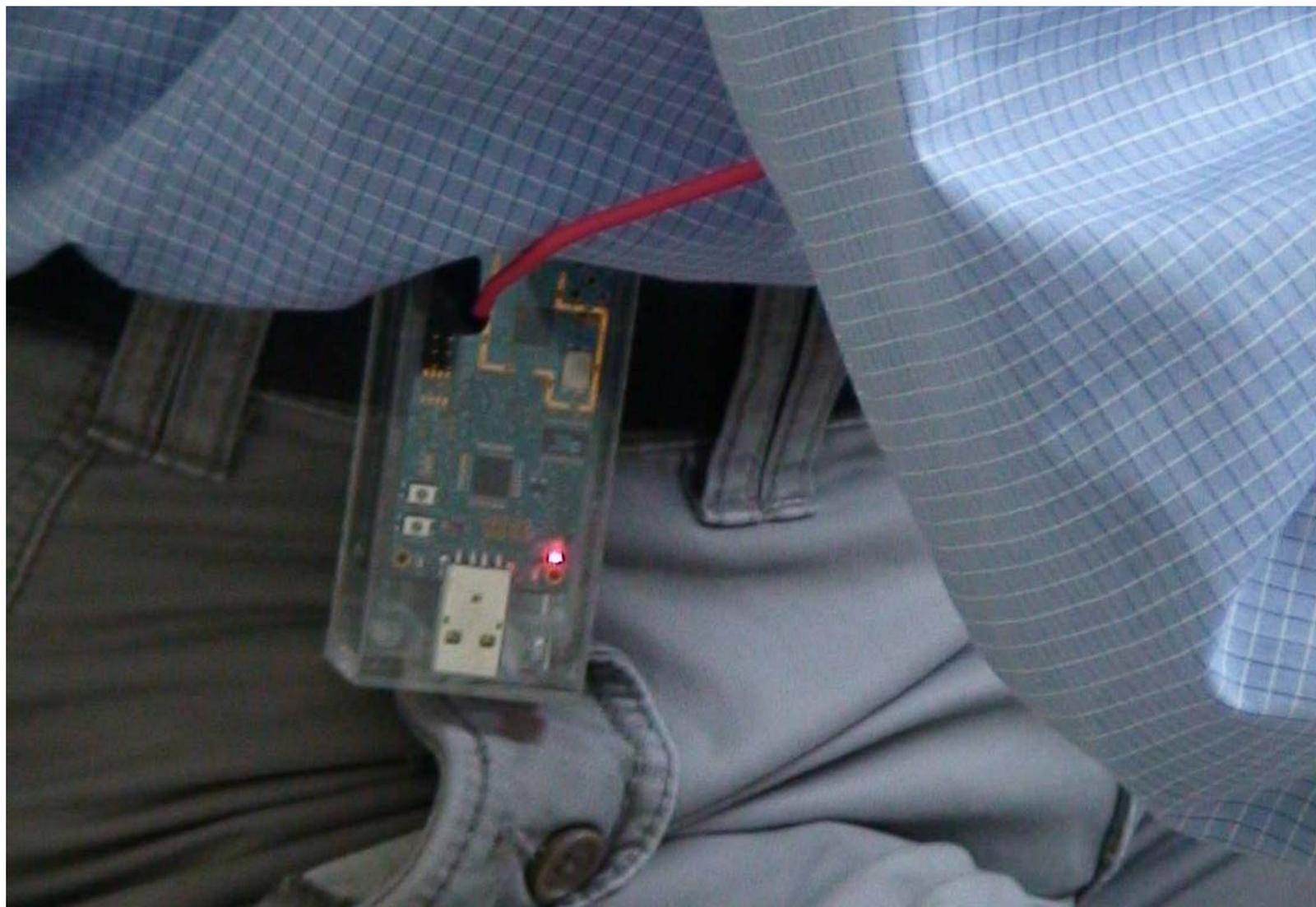
- 警告方式
 - 語音及UI改變顯示
 - 對象為本車以及他車使用者
- 警告範圍：本車以及本車外圍3hop之車輛
- 用掛置於使用者耳部的感測器，根據感測器的讀取數值作判斷
- 偵測使用者姿勢不良(打瞌睡)，判斷應發警告時，除車內以警示音、UI圖示變換，在透過車間傳輸，將警告訊息廣播出去

陀螺儀(感測器)- 耳機 包裝

- 所使用之感測器



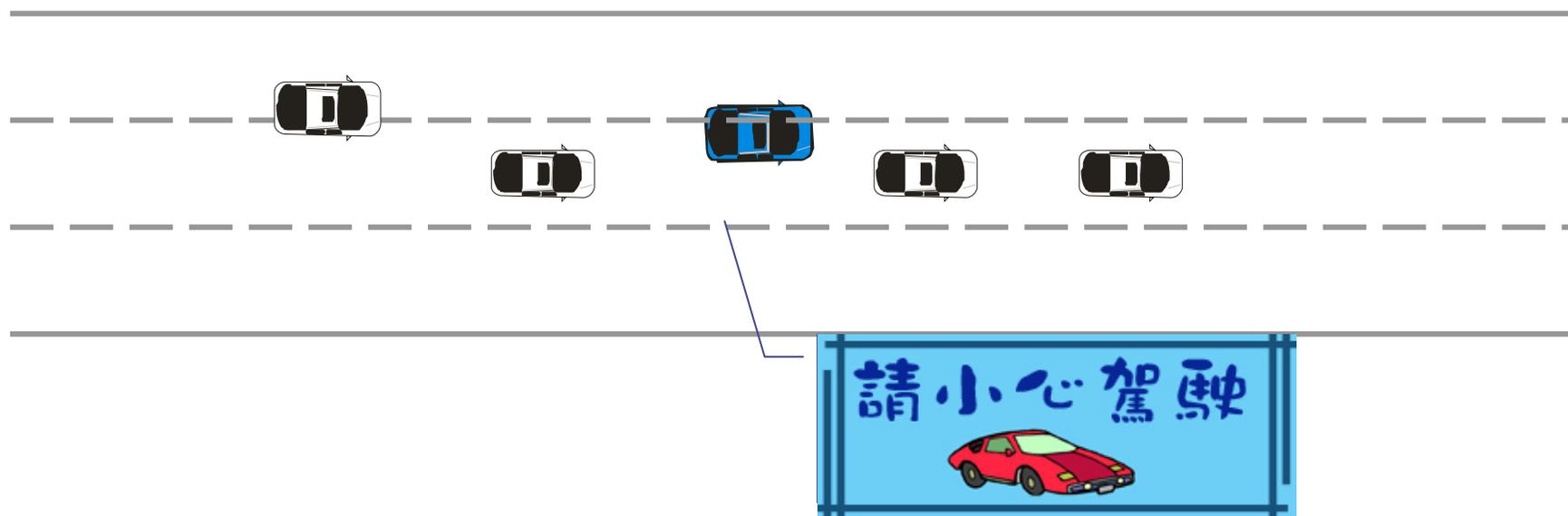
陀螺儀耳機 懸掛在腰間



駕駛人正常駕駛



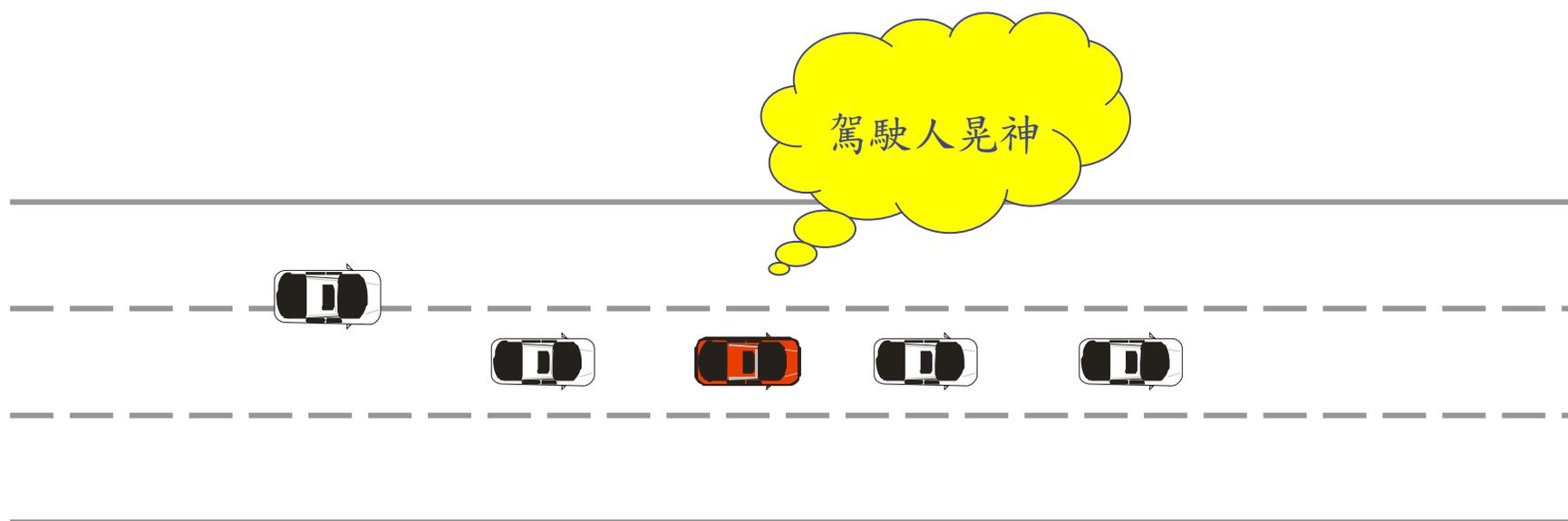
場景一：駕駛人姿勢不良



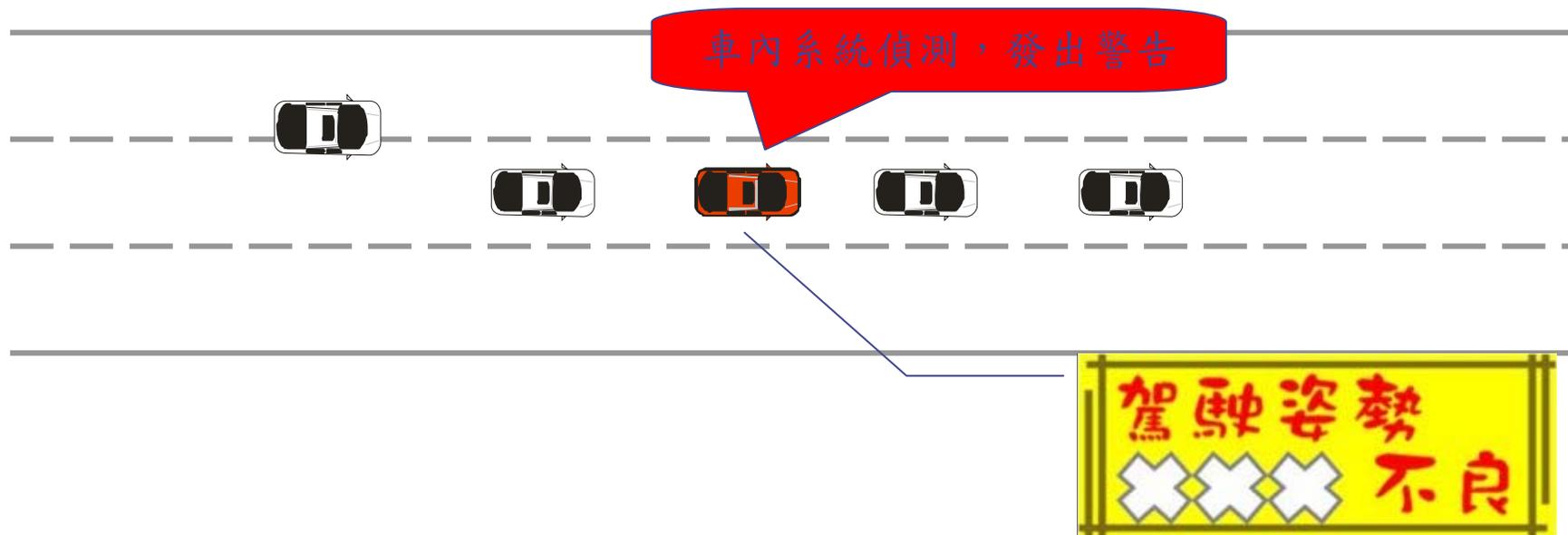
駕駛人姿勢不良 (打瞌睡)



場景一：駕駛人姿勢不良

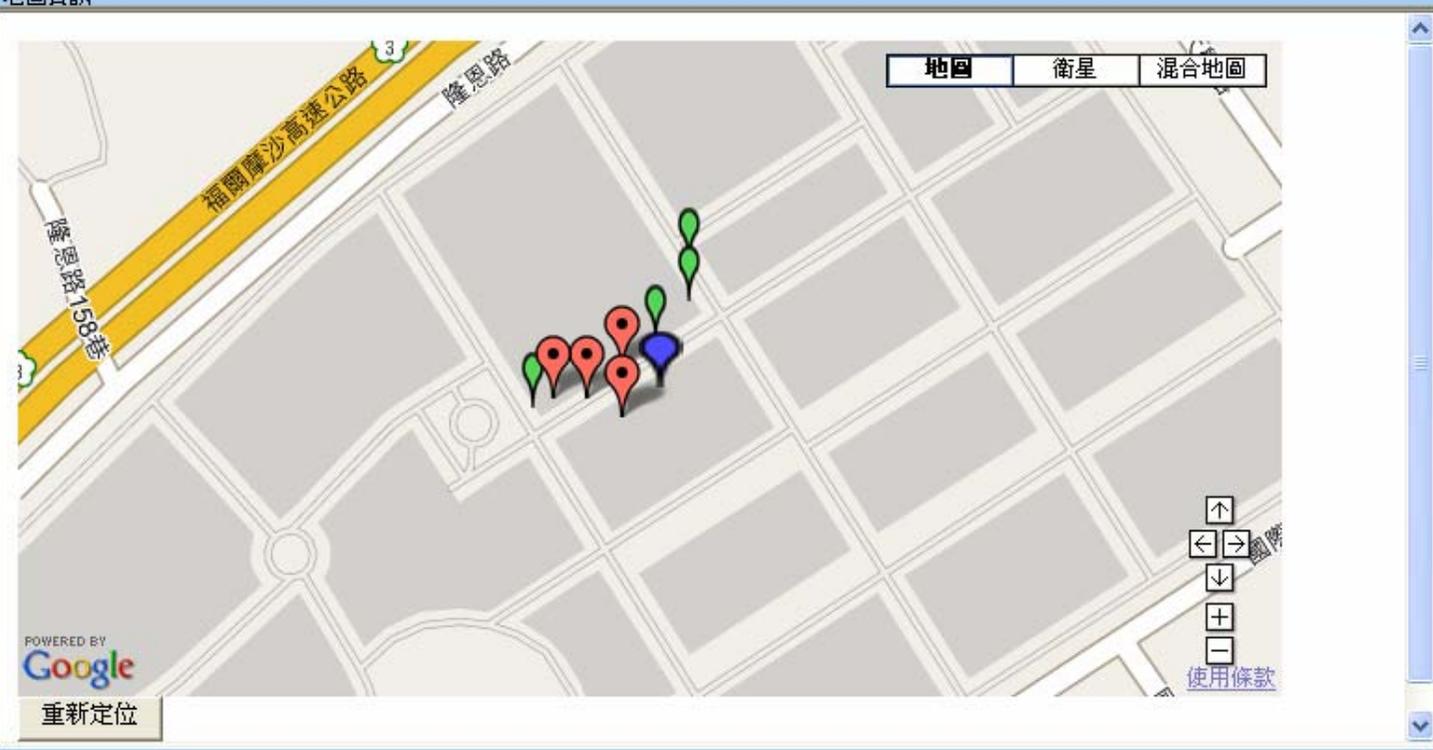


場景一：駕駛人姿勢不良



Form1

地圖資訊



地圖 衛星 混合地圖

POWERED BY Google

重新定位

使用條款

綜合狀態訊息

開始接收資料...!!



Edit

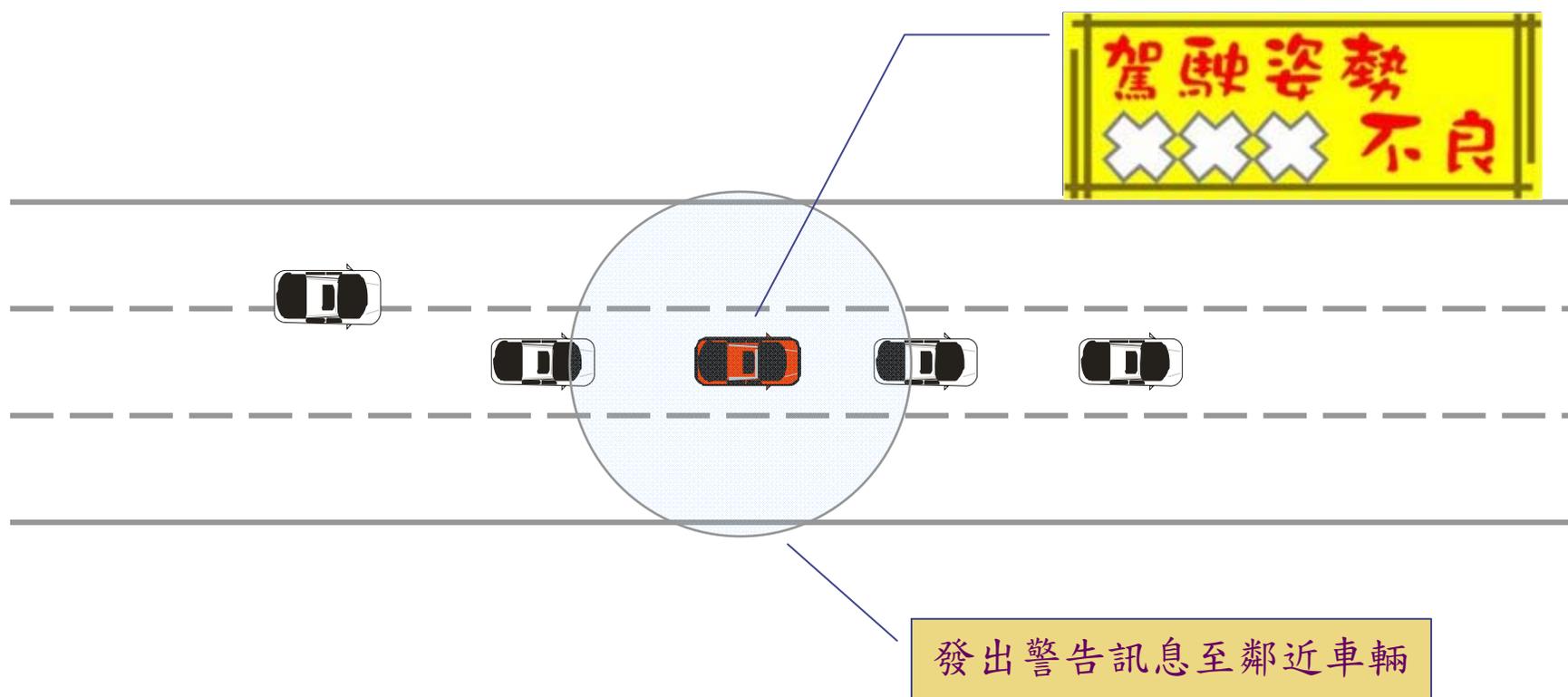
Set

Send

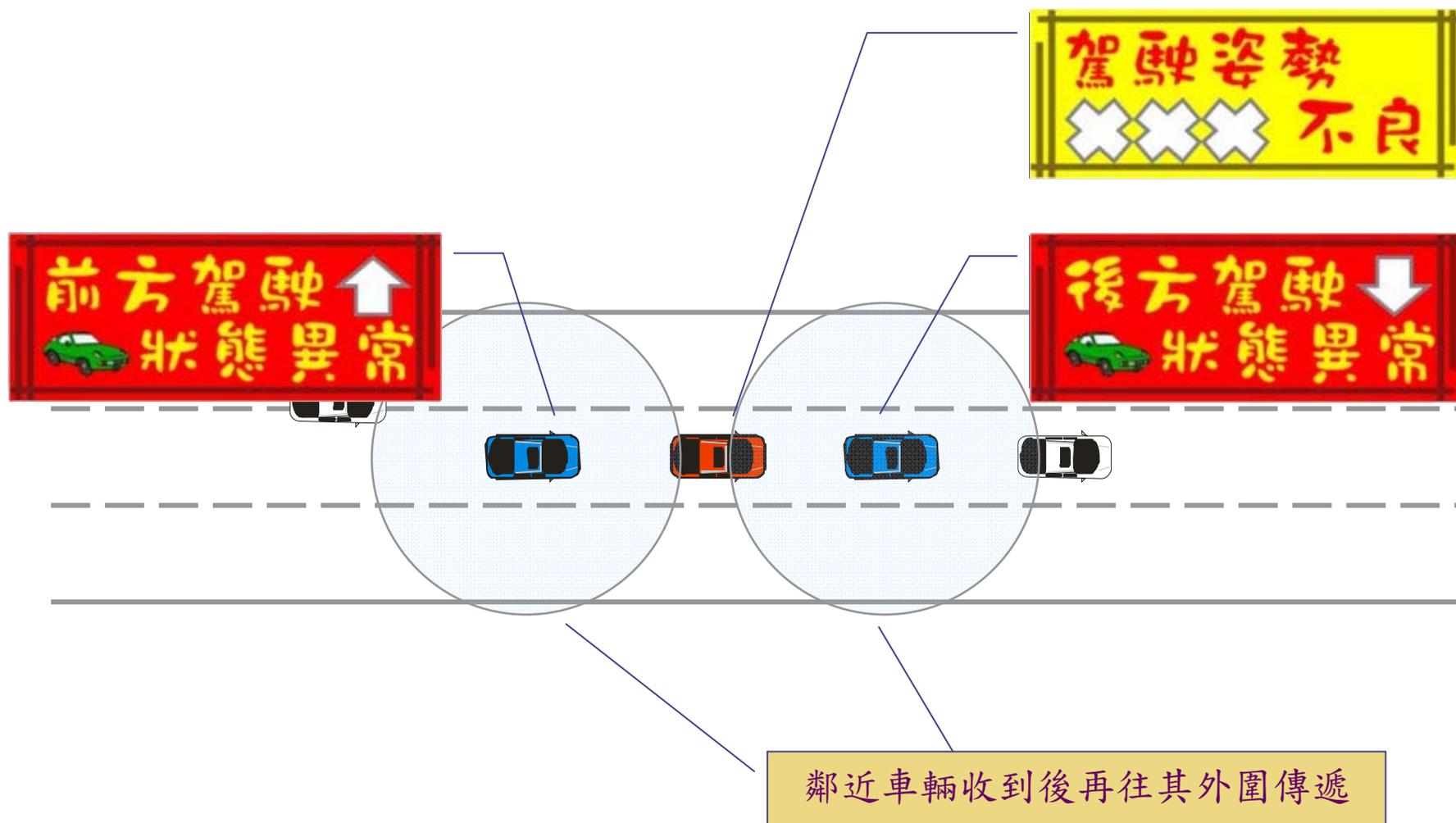
離開

✓ 狀態解除

場景一：駕駛人姿勢不良



場景一：駕駛人姿勢不良



Vehicular Ad Hoc Networks



警告訊息



Form1

地圖資訊

地圖 衛星 混合地圖

POWERED BY Google

重新定位

使用條款

綜合狀態訊息

開始接收資料...!!

Edit

Set

Send

離開

✓ 狀態解除

後方駕駛
↓
狀態異常

警告訊息



Form1

地圖資訊

地圖 衛星 混合地圖

重新定位

POWERED BY Google

使用條款

綜合狀態訊息

開始接收資料...!!

Edit

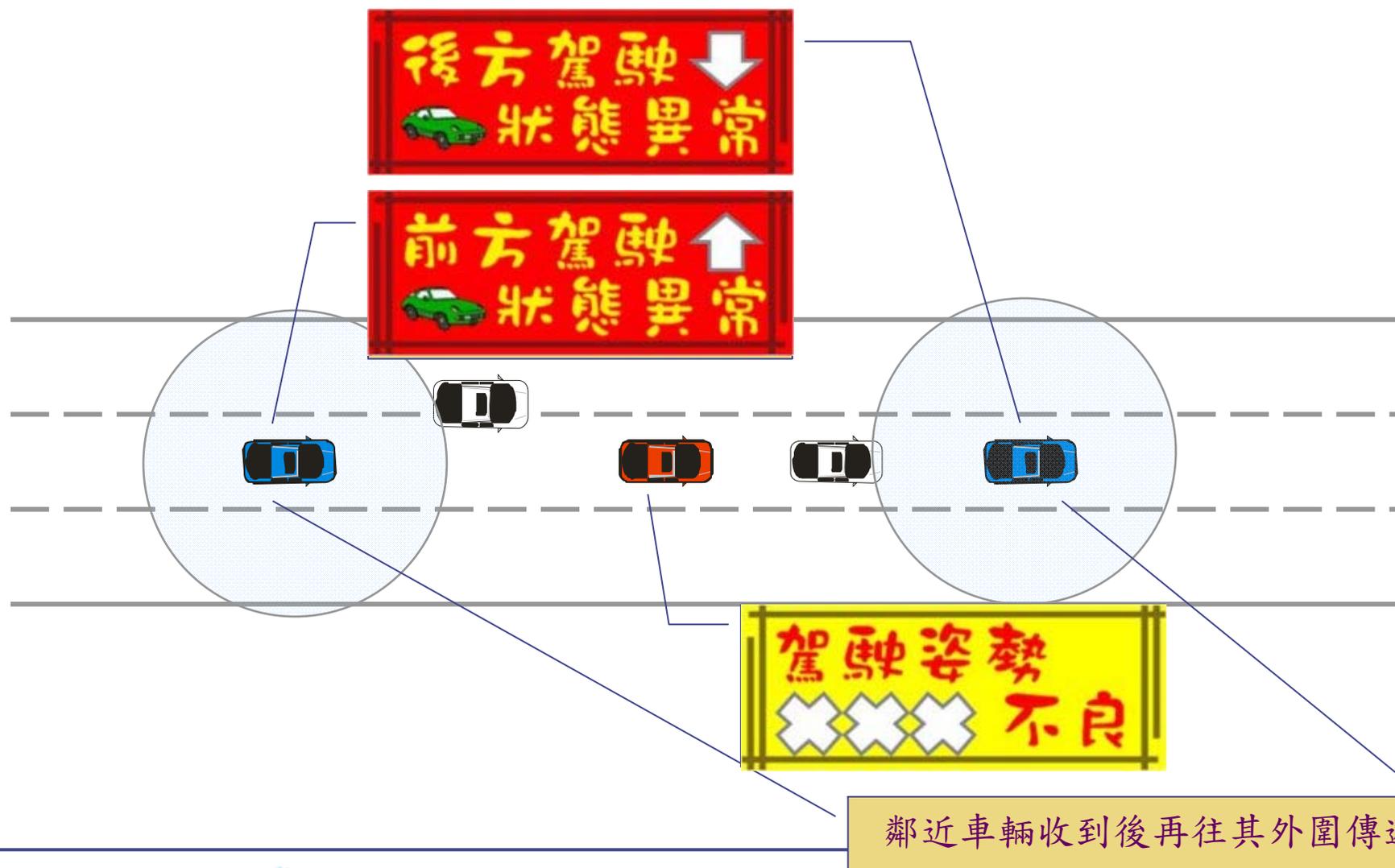
Set

Send

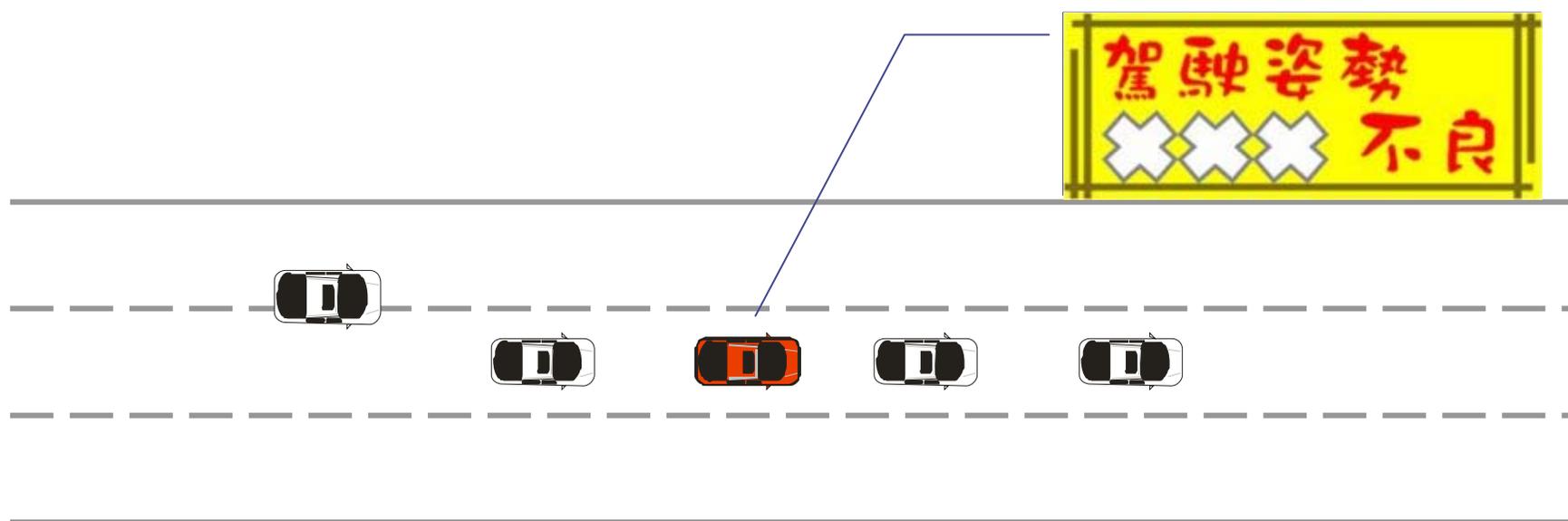
離開

✓ 狀態解除

場景一：駕駛人姿勢不良

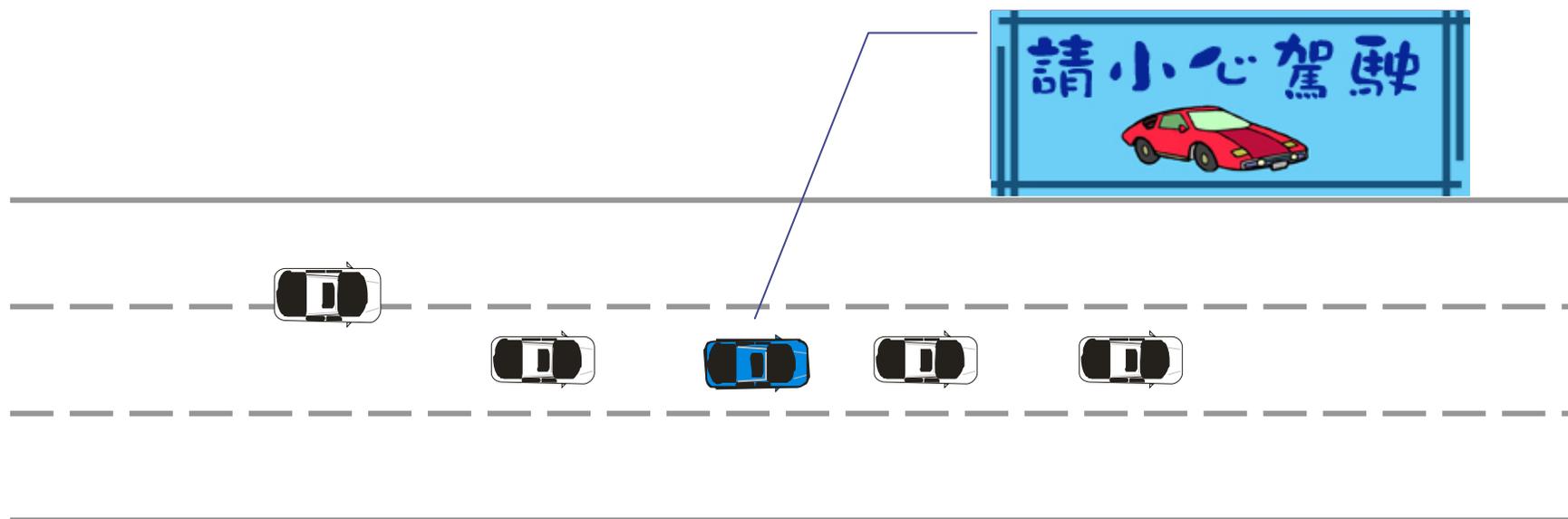


場景一：駕駛人姿勢不良



使用者可因系統警告恢復正確姿勢，或他車根據傳輸的警告作應變

場景一：駕駛人姿勢不良



直到本車駕駛恢復正常姿勢，才予以解除警告

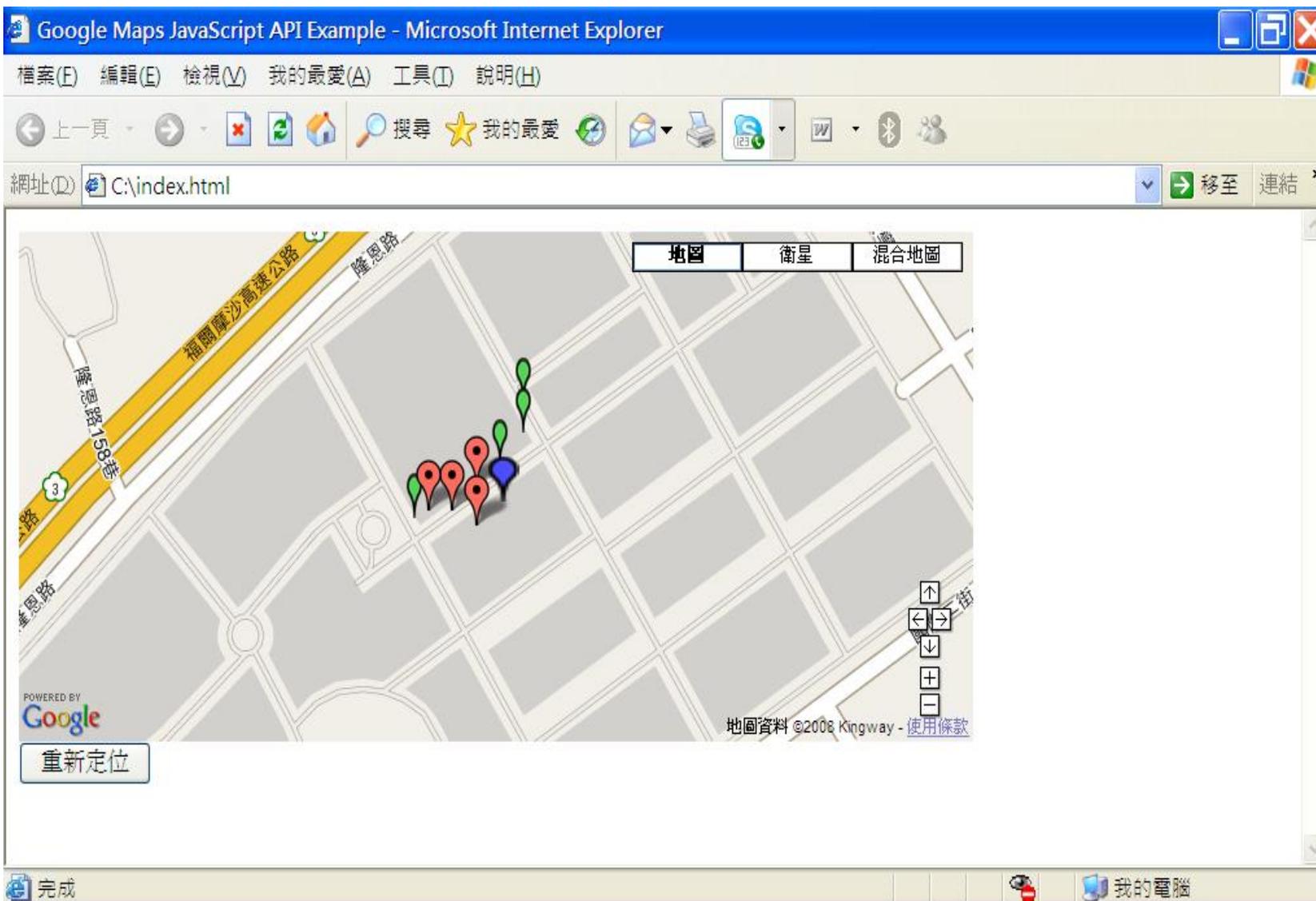
系統介面



The screenshot shows a web browser window titled "Form1" displaying a Google Maps interface. The map shows a street grid with labels for "福爾摩沙高速公路" (Foshan Expressway) and "隆恩路" (Lung-en Road). Several colored pins (red, green, blue) are placed on the map. The interface includes a "地圖資訊" (Map Information) header, a "地圖" (Map) tab, and a "衛星" (Satellite) tab. A "混合地圖" (Hybrid Map) tab is also visible. The map is powered by Google, as indicated by the "POWERED BY Google" logo. A "重新定位" (Reposition) button is located at the bottom left of the map area. On the right side of the map, there are navigation controls (up, down, left, right arrows) and a "使用條款" (Terms of Use) link.

Below the map, there is a "綜合狀態訊息" (General Status Information) section with the text "開始接收資料....!!". To the right of this section is a blue banner with the text "請小心駕駛" (Please Drive Carefully) and an illustration of a red car. Further right, there is a control panel with a "Edit1" field, "Set" and "Send" buttons, and a "離開" (Exit) button. At the bottom right, there is a "狀態解除" (Status Release) button with a green checkmark.

系統介面與 Google Map 結合



新元件介紹

- 藍色氣球為車主,氣球大小較大
- 綠色氣球為正常車輛,氣球大小較小
- 紅色氣球為發出警告的車輛,氣球大小正常

GPS 訊號格式: GGA

- 手動轉碼
- 得到訊號格式如下
- \$GPGGA,<1> ,<2> ,<3> ,<4> ,<5> ,<6> ,<7> ,<8> ,<9> ,<10>
,<11> ,<12> ,<13> , <14> ,<15>
- 確認GPS格式後由程式轉碼

擷取封包重要資訊

- <2> **緯度**表示格式:
度度分分.分分分分(ddmm.mmmm)
- <3> **緯度區域**表示:
北半球為N;南半球為S
- <4> **經度**表示格式:
度度度分分.分分分分(dddmm.mmmm)
- <5> **經度區域**表示:
東半球為E;西半球為W

封包內容顯示

- 36 71 80 71 71 65 44 49 51 51 57 52 57 46 48 48 48 44 50 52 53 54 46 54 53
50 49 44 78 44 49 50 49 50 50 46 49 54 51 56 44 69 44 48 44 48 48 44 44 48
46 48 44 77 44 48 46 48 44 77 44 44 48 48 48 48 42 52 68 13 10 36 71 80 82
77 67 44 48 57 51 52 48 49 46 50 51 48 44 86 44 48 48 48 48 46 48 48 48 48
44 78 44 48 48 48 48 48 46 48 48 48 48 44 69 44 44 44 50 48 48 52 48 54 44
44 44 78 42 55 56 13 10 36 71 80 86 84 71 44 44 84 44 44 77 44 44 78 44 44
75 44 78 42 50 67 13 10

擷取封包

- 以BCB將封包分段
- 擷取封包為
Str1:Buffer[18]~Buffer[26] //緯度資訊
Str2:Buffer[28] //南、北半球標示資訊
Str3:Buffer[30]~Buffer[39] //經度資訊
Str4:Buffer[41] //東、西
半球標示資訊

ASCII code 對照表

ASCII 碼	鍵盤	ASCII 碼	鍵盤	ASCII 碼	鍵盤	ASCII 碼	鍵盤
27	ESC	32	SPACE	33	!	34	"
35	#	36	\$	37	%	38	&
39	'	40	(41)	42	*
43	+	44	,	45	-	46	.
47	/	48	0	49	1	50	2
51	3	52	4	53	5	54	6
55	7	56	8	57	9	58	:
59	;	60	<	61	=	62	>
63	?	64	@	65	A	66	B
67	C	68	D	69	E	70	F
71	G	72	H	73	I	74	J
75	K	76	L	77	M	78	N

ASCII 碼	鍵盤						
79	O	80	P	81	Q	82	R
83	S	84	T	85	U	86	V
87	W	88	X	89	Y	90	Z
91	[92	\	93]	94	^
95	_	96	`	97	a	98	b
99	c	100	d	101	e	102	f
103	g	104	h	105	i	106	j
107	k	108	l	109	m	110	n
111	o	112	p	113	q	114	r
115	s	116	t	117	u	118	v
119	w	120	x	121	y	122	z
123	{	124		125	}	126	~

ASCII Code To Integer

- 數字部份ASCII CODE-48=Integer
- 標點部分已SWITCH轉換
- ex: 50 52 53 54 46 54 53 50 49字串各自-48得 2 4 5 6 . 6
5 2 1
- 大於等於0且小於10的整數直接帶入Integer其他則以switch轉換48在ASCII CODE中代表的是''
- 之後封包將再分割處理,''被忽略

Google map 經緯度格式

- Google Map 經緯度值為 (x,y)
- X 為緯度值, 當北緯時, $x > 0$; 南緯時, $x < 0$
- Y 為經度值, 當東經時, $x > 0$; 西經時, $x < 0$
- (x,y) 的精準值必須到小數點第六位, 多餘的捨去

利用BCB處理GPS訊號值1:緯度

```
s1d= (Buffer[18]-48)*10+(Buffer[19]-48); //讀取緯度'度'的部份
s1f= (Buffer[20]-48)*10+(Buffer[21]-48) //讀取緯度'分'的部份
s1m=(Buffer[23]-48)*1000+(Buffer[24]-48)*100+(Buffer[25]-48)*10+(Buffer[26]-48); // 讀取
緯度'秒'的部份
s1=s1d+(s1f*60+(s1m/10000)*60)/3600; //獲得緯度
s2=Buffer[28];
if(s2 == 78)
{
s1 = 0+s1; //北緯時值>0
}
if(s2 == 83)
{
s1 =0-s1; //南緯時值<0
}
Memo1->Lines->Add(s1); //印出緯度
```

利用BCB處理GPS訊號值1:經度

```
s3d= (Buffer[30]-48)*100+(Buffer[31]-48)*10+(Buffer[32]-48); //讀取經度'度'的  
    部分  
3f= (Buffer[33]-48)*10+(Buffer[34]-48) //讀取經度'分'的部份  
s3m=(Buffer[36]-48)*1000+(Buffer[37]- 48)*100+ (Buffer[38]- 48)*10  
    +(Buffer[39]-48); /讀取經度'秒'的部份  
s3=s3d+(s3f*60+(s3m/10000)*60)/3600; //獲得經度  
    s4= Buffer[41];  
    if(s4 == 69)  
    {  
        s3=0+s3; //東經時值>0  
    }  
    if(s4 == 87)  
    {  
        s3=0-s3; //西經時值<0  
    }  
    Memo1->Lines->Add(s3); //印出經度
```

暫存資料

利用BCB將擷取字串存取下來存取在C槽檔名為data1.txt與data2.txt

data1 存取緯度;data2存取經度

程式碼如下: 記得先加標頭檔#include"stdio.h"

```
FILE *outfile1; //宣告緯度輸出檔案
```

```
if((outfile1=fopen("C:\\lat.txt","w"))==NULL)return;
```

```
//輸出值成為lat.txt存於C槽
```

```
fprintf(outfile1,"%f",s1); //經準度取到小數點第6位
```

```
fclose(outfile1); //輸出結束
```

```
FILE *outfile2; //宣告經度輸出檔案
```

```
if((outfile2=fopen("C:\\lon.txt","w"))==NULL)return;
```

```
//輸出值成為lon.txt存於C槽
```

```
fprintf(outfile2,"%f",s3); //經準度取到小數點第6位
```

```
fclose(outfile2); //輸出結束
```

讀取暫存檔

- 讀取暫存資料lat.txt與lon.txt
- lat.txt為緯度資料
- lon.txt為經度資料

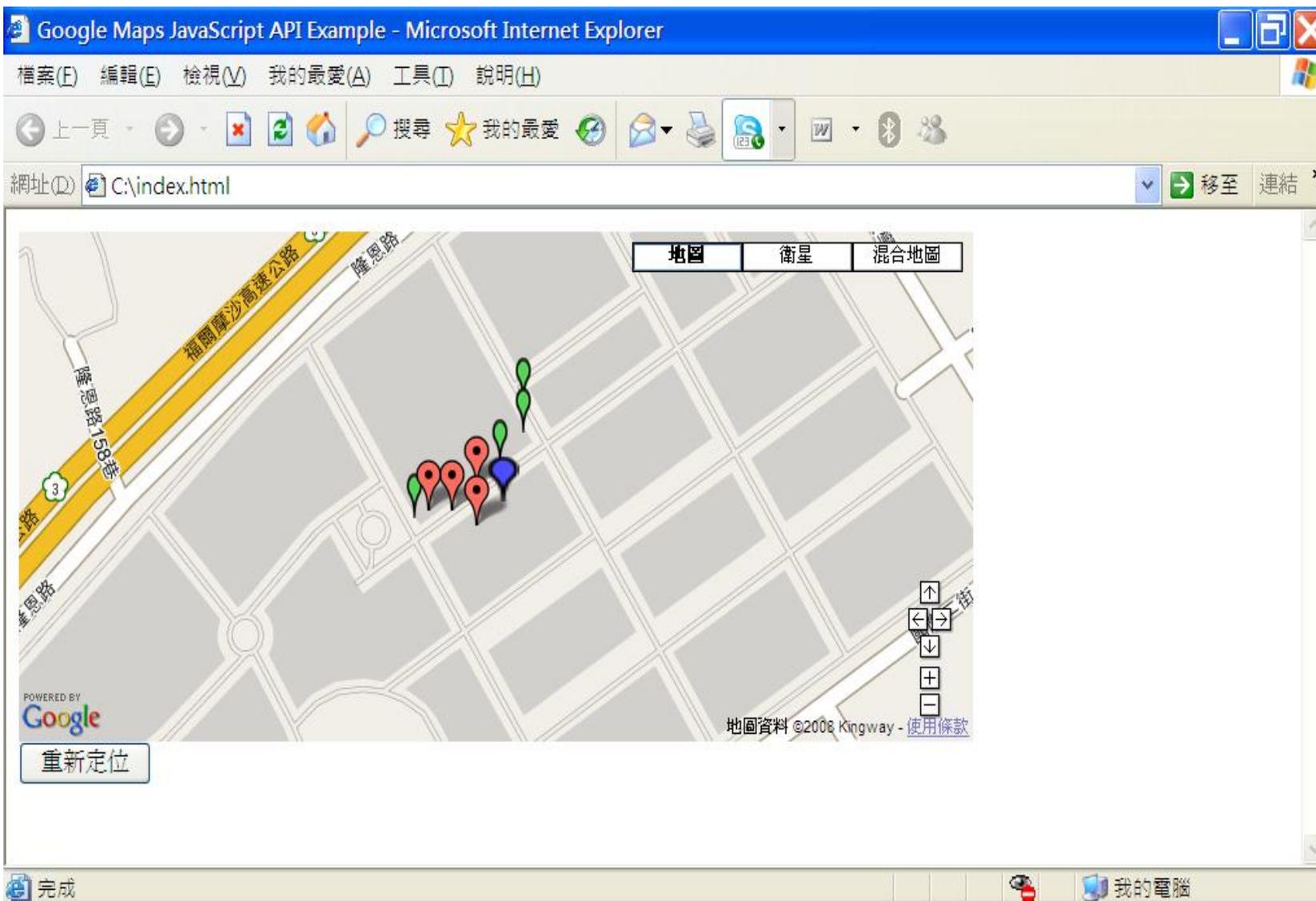
與 Google Map 結合

- 將所收到的訊號透過 **Javascript** 讀取標示於Google Map地圖
- 以下為讀檔部份程式碼:
- `var fso1, ts1, s1 ,o1 ; //宣告變數`
- `fso1 = new ActiveXObject("Scripting.FileSystemObject"); //宣告新的activeX物件`
- `ts1 = fso1.OpenTextFile("c:\\lat.txt", 1); //打開lat.txt`
- `s1 = ts1.ReadLine(); //讀取緯度資料`
-
- `var fso2, ts2, s2 ,o2 ; //宣告變數`
- `fso2 = neActiveXObject("Scripting.FileSystemObject"); //宣告新的activeX物件`
- `ts2 = fso2.OpenTextFile("c:\\lon.txt", 1); //打開lon.txt`
- `s2 = ts2.ReadLine(); //讀取經度資料`

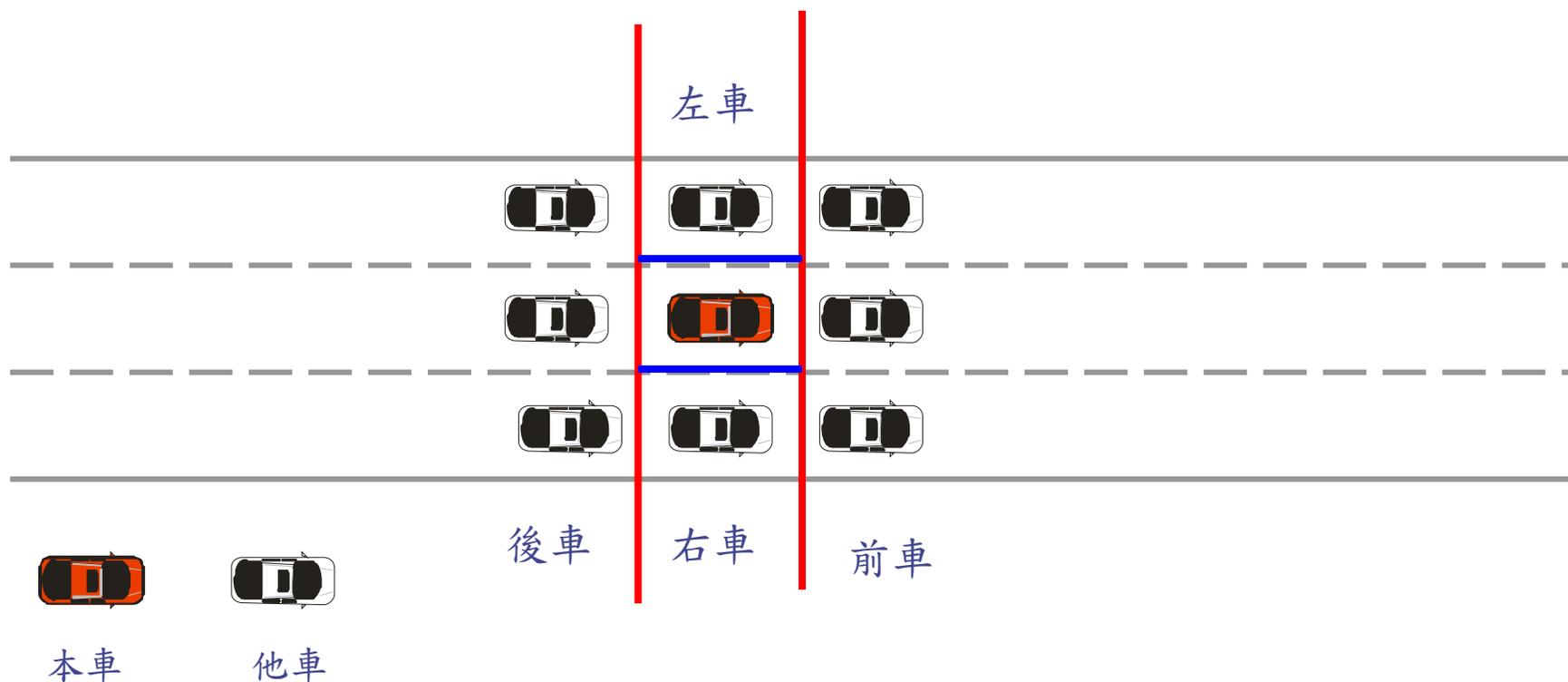
GPS讀取至Google Map

- 一顆的資料收取完畢後
- 嘗試搜尋2顆及多顆GPS訊號
- 試著讀取同group以及不同group所收到的GPS訊息封包直至可同時顯示至Google Map

Google Map

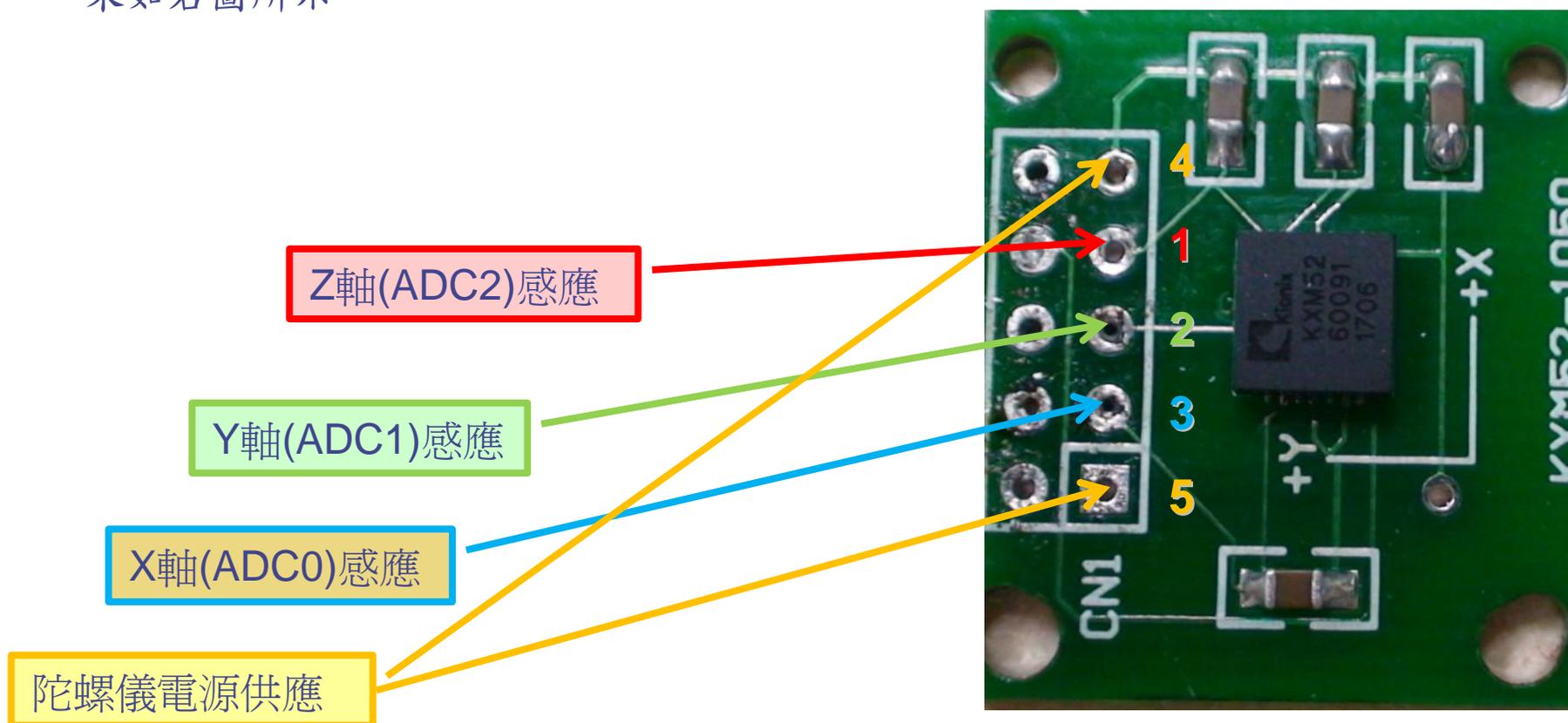


車輛的相對位置之判別



陀螺儀結構

陀螺儀可感應測量X、Y、Z三軸的加速度變化，本次實驗所需接線的腳未如右圖所示

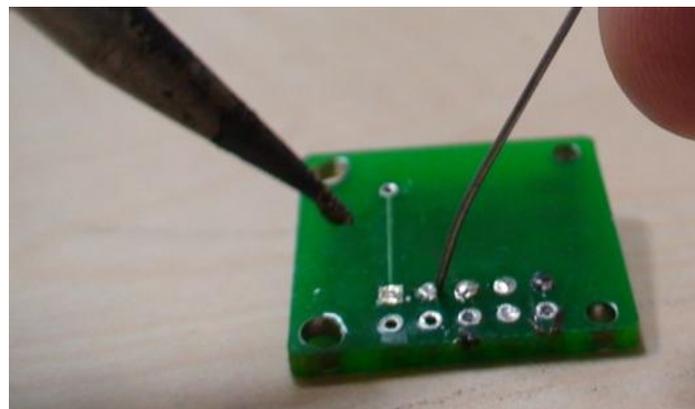


陀螺儀接線

Step1：利用透膠將陀螺儀背面貼上保護層。



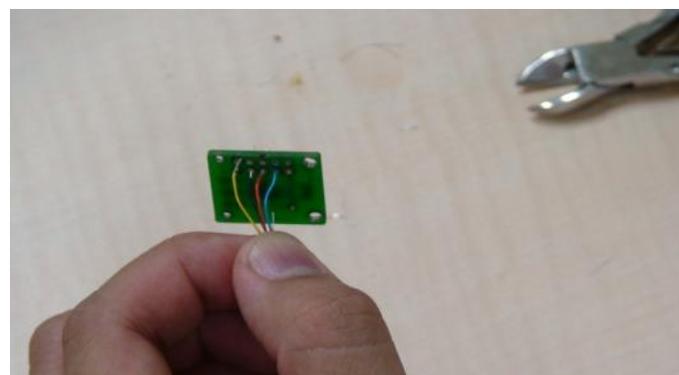
Step2：用焊錫將針腳填滿備用



Step3：取五條單芯線備用，請先編號以免搞混



Step4：依序焊接完成



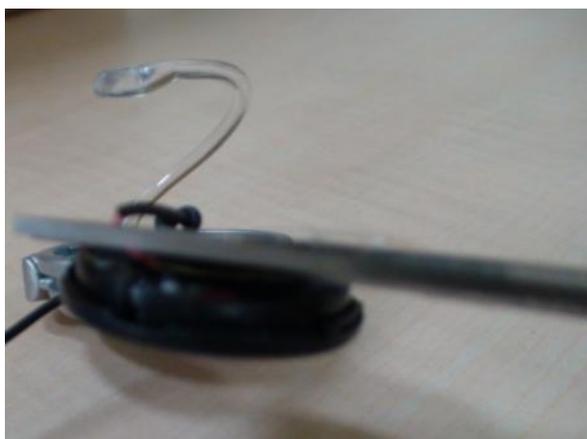
拆解耳機

Step1：取右耳耳機，將其外耳海綿取下



拆解耳機

Step2：利用剪刀或尖嘴鉗，將耳機外殼敲開，並將內部音源線及擴音單元拆下



組合陀螺儀及耳機

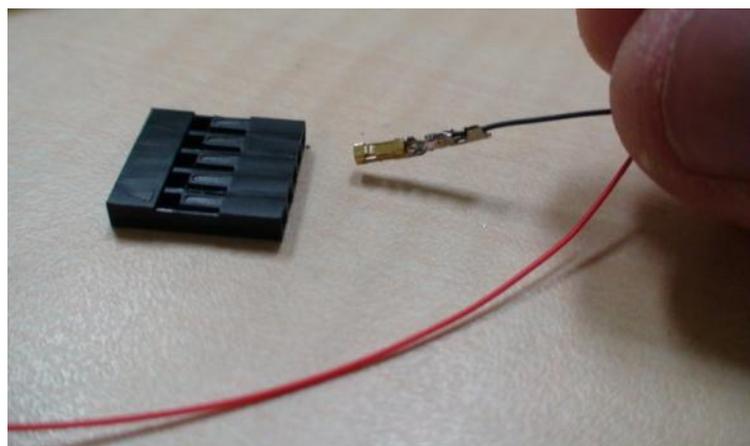


Step1：將焊好之單芯線束穿過
耳機外殼，並利用保麗龍膠將陀
螺儀固定



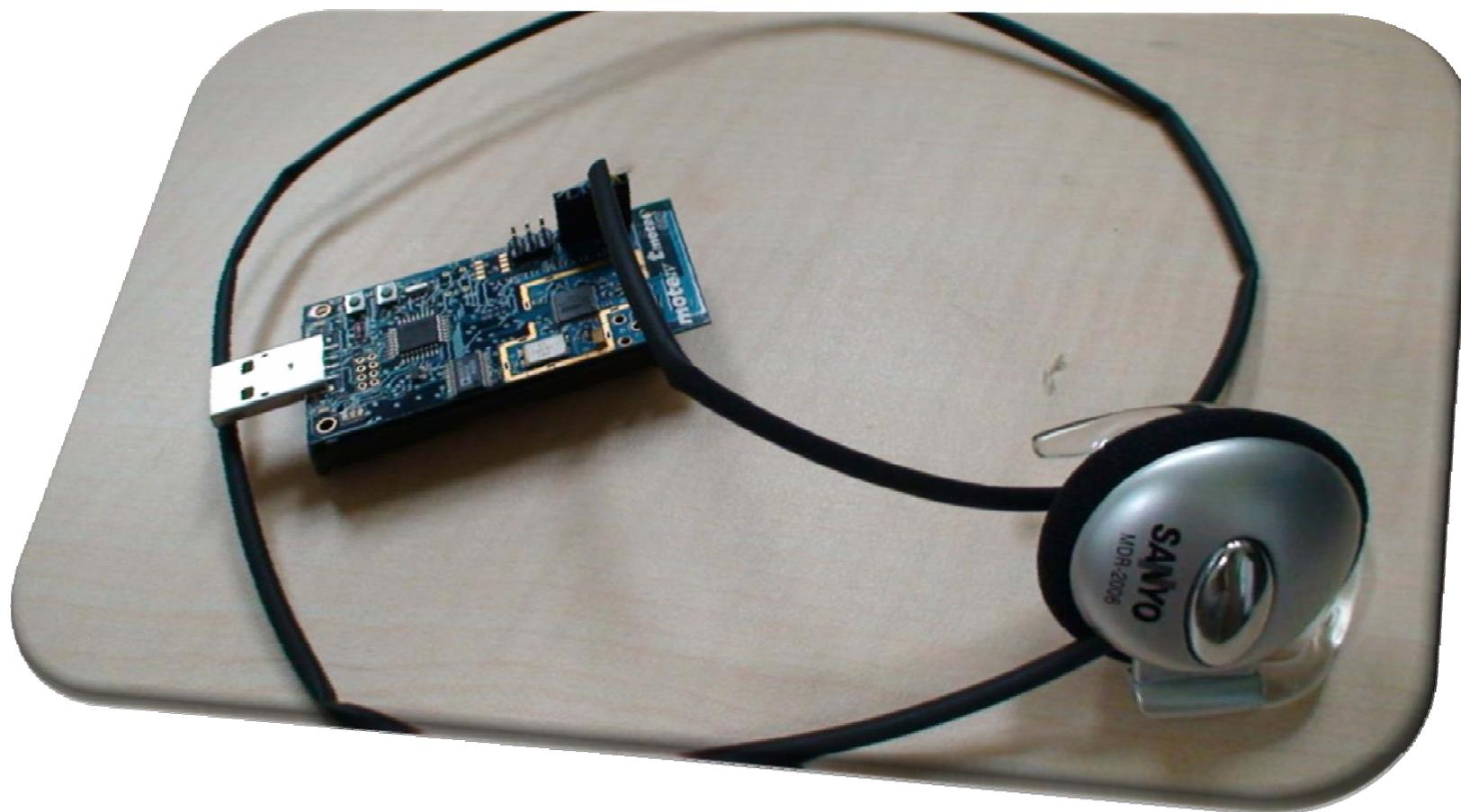
Step2：將外殼及外
耳海綿裝回

收線及接上連接器



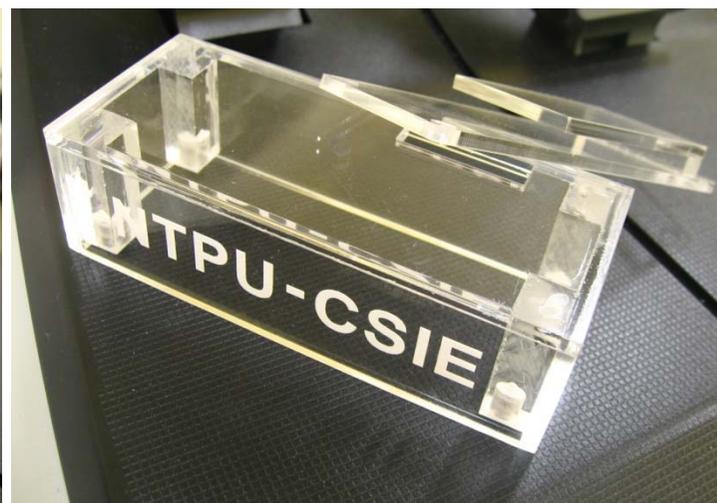
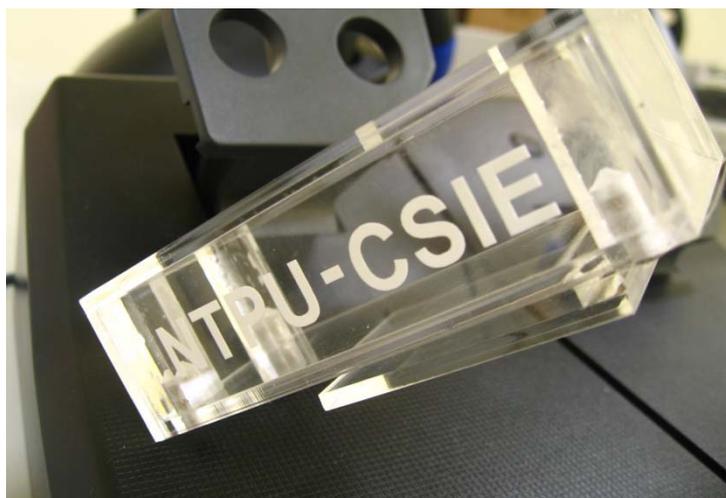
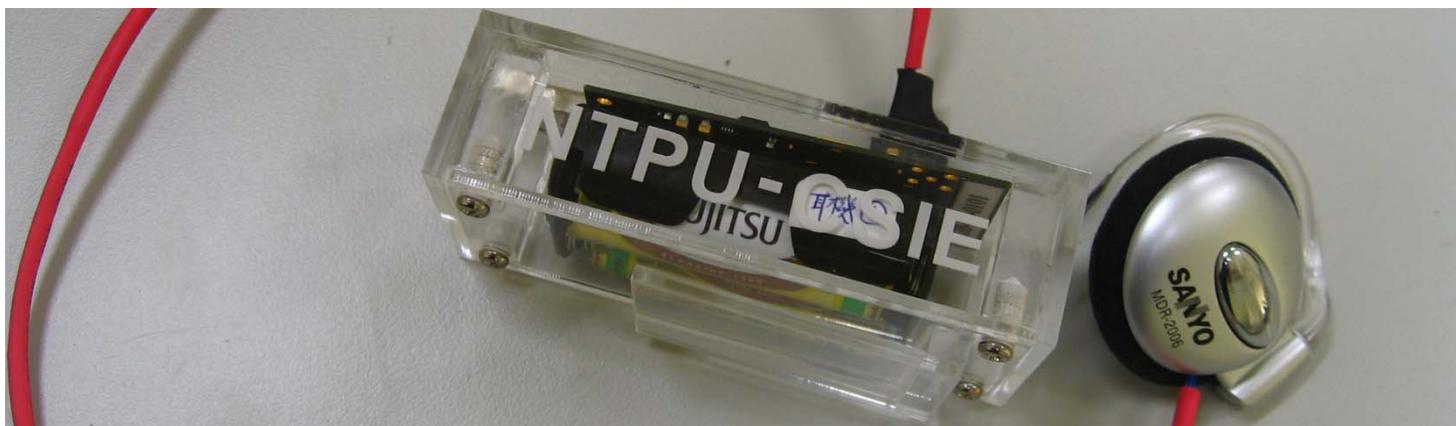
取下金屬接頭，與單芯線焊接好，最後再
嵌入塑膠連接器，請注意嵌入順序

陀螺儀(感測器)-耳機 成品

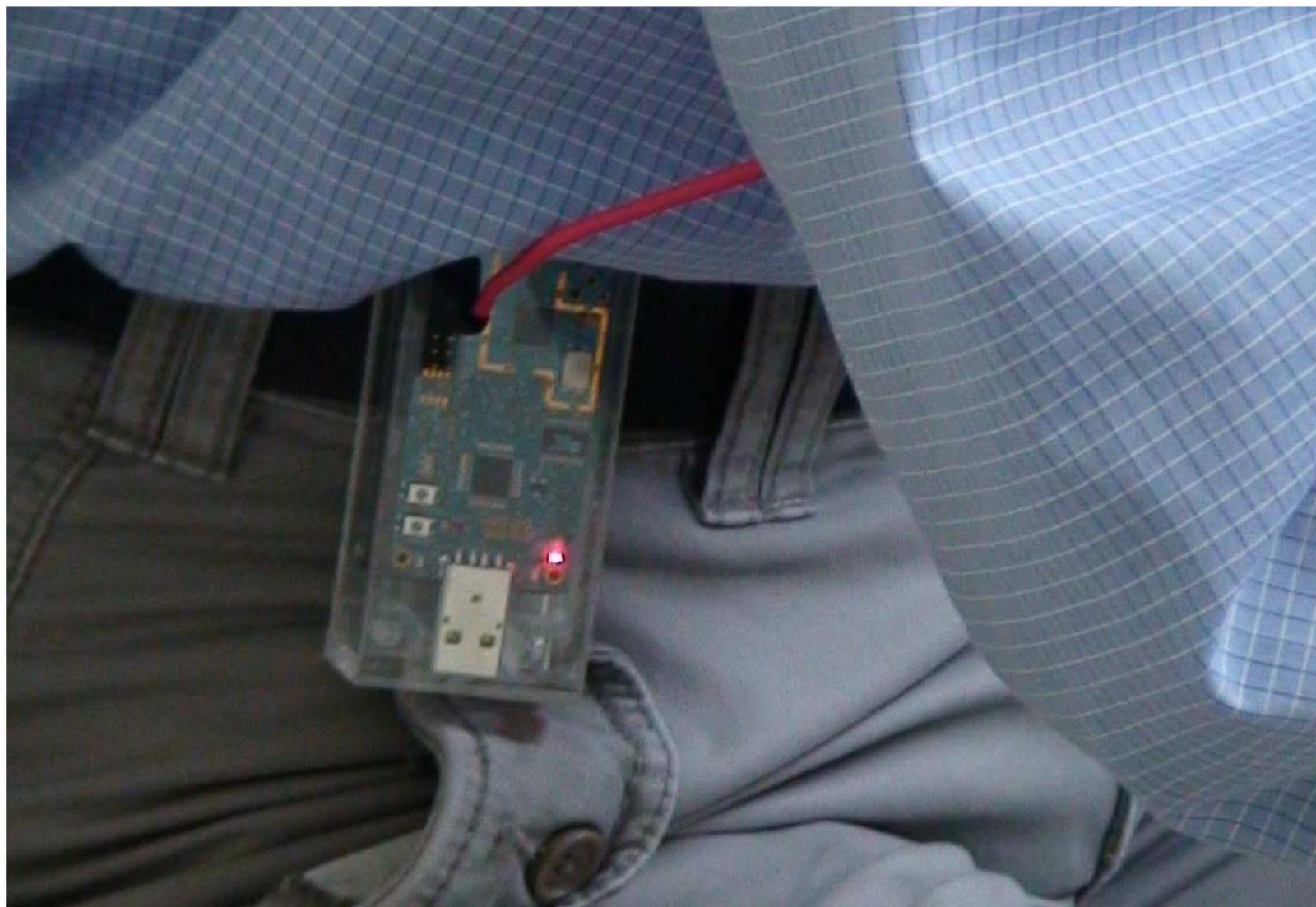


陀螺儀(感測器)- 耳機 包裝

- 所使用之感測器

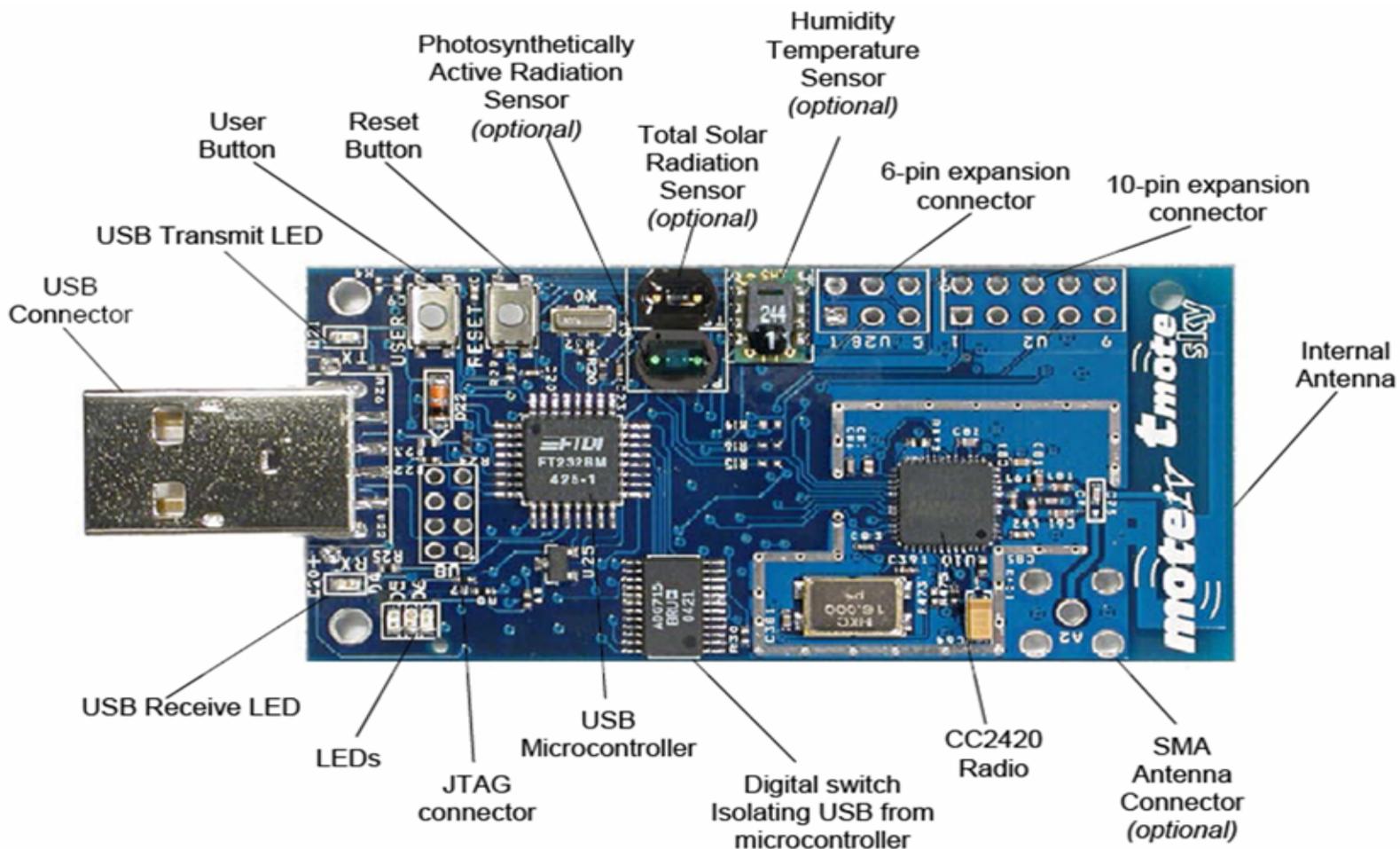


懸掛在腰間

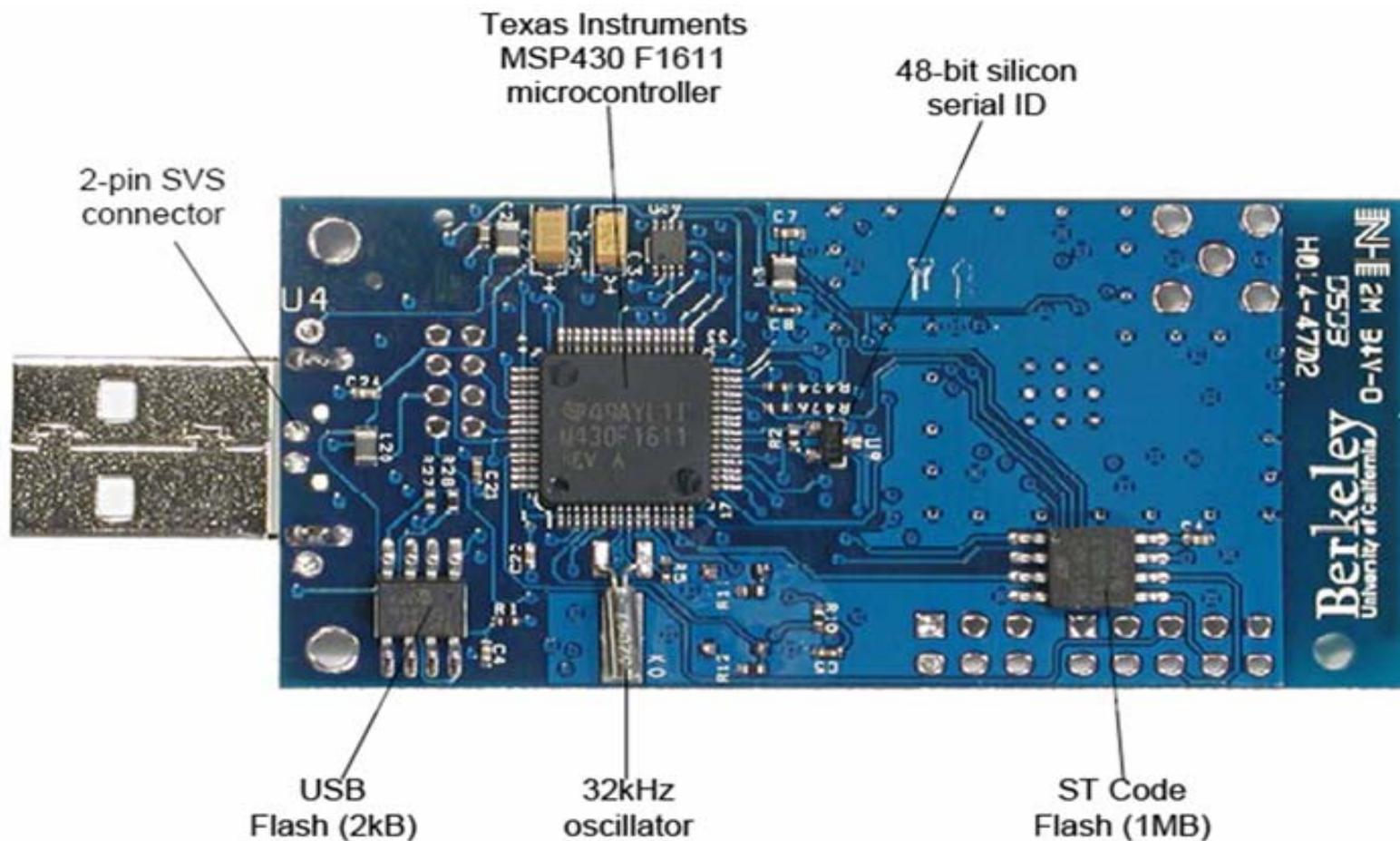


Tmote Sky Hardware (1/2)

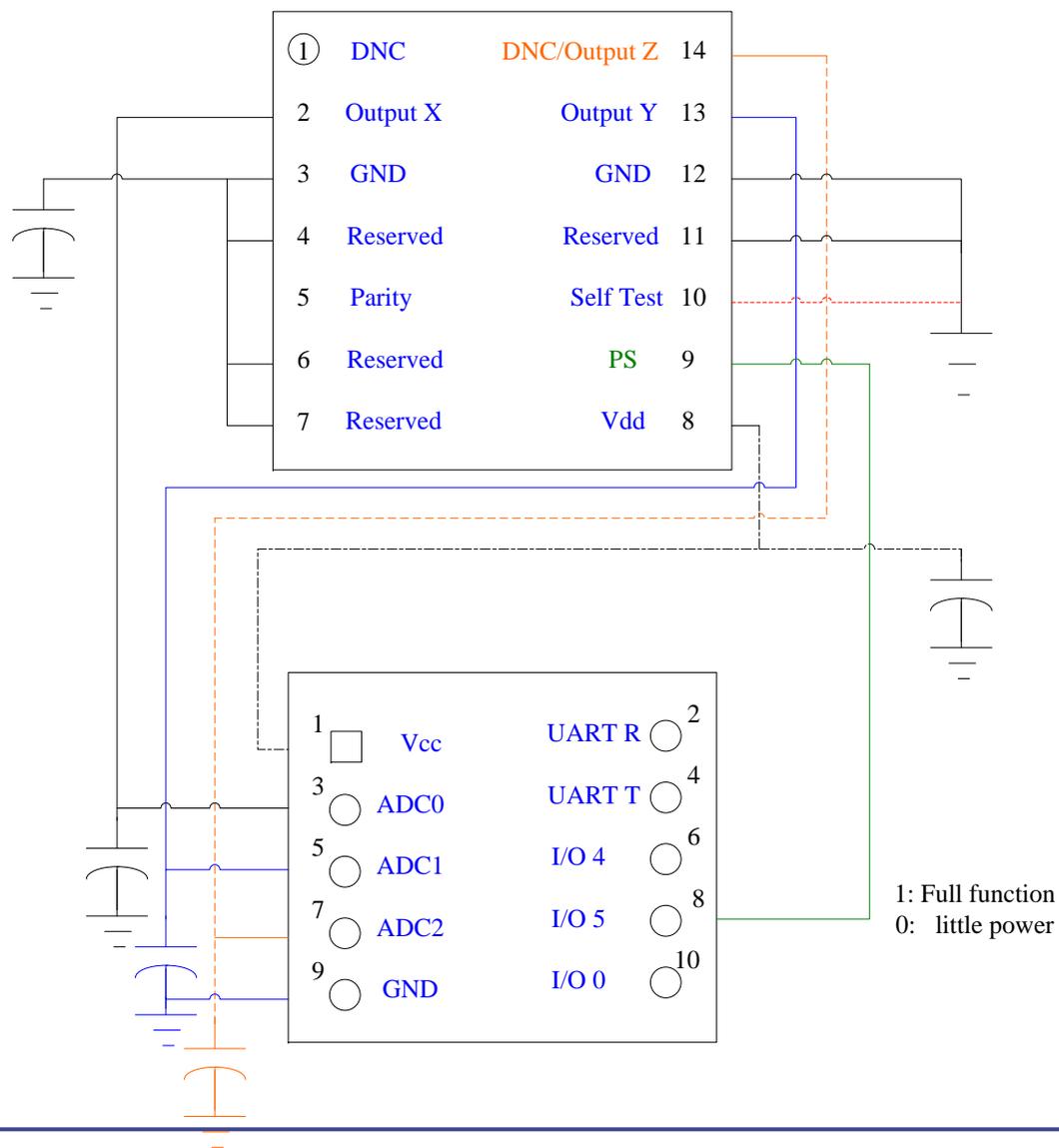
Figure : Front of the Tmote Sky module



Tmote Sky Hardware (1/2) Figure : Back of the Tmote Sky module



Sensor Node 與陀螺儀模組之腳位圖



TinyOS Basic Command

- Command : **Motelist**
 - Motelist is a command-line utility that displays which serial ports have Tmote motes attached
 - Run it with "**motelist**"; try "**motelist -h**" for additional options

```
Administrator@mychat-937530e0 ~  
$ motelist  
SerialNum  PortName  Description  
-----  
M4ACW9GU  COM7      tmote sky
```

TinyOS Basic Command - Displaying Data on a PC

- After programming your mote with the code, type
 - `export MOTECOM=serial@serialport:baudrate`
 - Serialport is the serial port that you have connected the programming board to
 - baudrate is the specific baudrate of the mote.
- For the mica and mica2dot motes, the baud rate is 19200
- for the mica2 it is 57600 baud

TinyOS Basic Command - Displaying Data on a PC

- For example:
 - `export MOTECOM=serial@COM1:19200 # mica baud rate`
 - `export MOTECOM=serial@COM3:57600 # explicit mica2 baud rate`
- Set MOTECOM appropriately, then run
 - `java net.tinyos.tools.Listen`
- You should see some output resembling the following:
 - `07 01 08 C2 FF FF FF FF 07 27 01 FF FF 09 F2 0F 56 07 01 08 C2 0C
FF FF FF FF 07 27 01 0F FF 09 F2 0F 56`

DEMO - VANET Application

- We need to burn
 - TOSbase in one tmote.
 - TelosADC in 4 tmotes.
- Set the node id for different tmote.
- **Node id 0 is for TOSBase**
- **Node id 1 is for Ear tmote sensor**
- Node id 2 is for wheel tmote sensor
- Node id 3 is for stick tmote sensor
- Node id 4 is for brake tmote sensor.

DEMO - VANET Application

- 1. Set the directory to : /opt/moteiv/apps/TOSBase

```
Administrator@mychat-937530e0 ~  
$ cd /opt/moteiv/apps/TOSBase  
Administrator@mychat-937530e0 /opt/moteiv/apps/TOSBase  
$
```

- 2. make tmote TOSBase

```
Administrator@mychat-937530e0 /opt/moteiv/apps/TOSBase  
$ make tmote  
mkdir -p build/tmote  
compiling TOSBase to a tmote binary  
ncc -o build/tmote/main.exe -O -Wall -Wshadow -DDEF_TOS_AM_GROUP  
-target=tmote -fnesc-cfile=build/tmote/app.c -board= -DTOSH_MAX  
TOSH_DATA_LENGTH=TOSH_MAX_DATA_LENGTH -I/opt/moteiv/tos/platform  
teiv/tos/platform/tmote/util/uartdetect -I/opt/moteiv/tos/platfo  
/opt/moteiv/tos/platform/msp430/dac -I/opt/moteiv/tos/platform/m  
/moteiv/tos/platform/msp430/resource -I/opt/moteiv/tos/platform/  
opt/moteiv/tos/platform/msp430 -I/opt/moteiv/tos/lib/util/pool
```

DEMO - VANET Application

- 3. make tmote install,0 bsl,8
 - 0 for node id,
 - Bsl,8 is for comport 9 (bsl,8 = COM9)

```
Administrator@mychat-937530e0 /opt/moteiv/apps/TOSBase
$ motelist
SerialNum  PortName      Description
-----
M4AC68U5   COM9          tmote sky

Administrator@mychat-937530e0 /opt/moteiv/apps/TOSBase
$ make tmote install,0 bsl,8
mkdir -p build/tmote
    compiling TOSBase to a tmote binary
ncc -o build/tmote/main.exe -O -Wall -Wshadow -DDEF_TOS_A
    -target=tmote -fnesc-cfile=build/tmote/app.c -board= -DT
TOSH_DATA_LENGTH=TOSH_MAX_DATA_LENGTH -I/opt/moteiv/tos/p
```

DEMO - VANET Application

- 4. Set the directory to :opt/tinyos-1.x/apps/TelosADc

```
Administrator@mychat-937530e0 /opt/moteiv/apps/TOSBase
$ cd /opt/tinyos-1.x/apps/TelosADc
Administrator@mychat-937530e0 /opt/tinyos-1.x/apps/TelosADc
$
```

- 5. make tmote

```
Administrator@mychat-937530e0 /opt/tinyos-1.x/apps/TelosADc
$ make tmote
mkdir -p build/tmote
    compiling TelosADCC to a tmote binary
ncc -o build/tmote/main.exe -O -Wall -Wshadow -DDEF_TOS_AM_GROU
  -target=tmote -fnesc-cfile=build/tmote/app.c -board= -I/opt/mo
/tmote -I/opt/moteiv/tos/platform/tmote/util/uartdetect -I/opt/
```

DEMO - VANET Application

- 6. make tmote install,1 bsl,8
 - 1 for is node id
 - Bsl,8 is COM9

```
Administrator@mychat-937530e0 /opt/tinyos-1.x/apps/TelosADc
$ make tmote install,1 bsl,8
mkdir -p build/tmote
    compiling TelosADCC to a tmote binary
ncc -o build/tmote/main.exe -O -Wall -Wshadow -DDEF_TOS_AM_GROUP=0x27 -Wnesc-all
-target=tmote -fnesc-cfile=build/tmote/app.c -board= -I/opt/moteiv/tos/platform
/tmote -I/opt/moteiv/tos/platform/tmote/util/uartdetect -I/opt/moteiv/tos/platfo
rm/msp430/adc -I/opt/moteiv/tos/platform/msp430/dac -I/opt/moteiv/tos/platform/m
sp430/dma -I/opt/moteiv/tos/platform/msp430/resource -I/opt/moteiv/tos/platform/
msp430/timer -I/opt/moteiv/tos/platform/msp430 -I/opt/moteiv/tos/lib/util/pool -
I/opt/moteiv/tos/lib/util/button -I/opt/moteiv/tos/lib/util/null -I/opt/moteiv/t
os/lib/util -I/opt/moteiv/tos/lib/MultiHopLQI -I/opt/moteiv/tos/lib/netsync -I/o
pt/moteiv/tos/lib/sp -I/opt/moteiv/tos/lib/sp/cc2420 -I/opt/moteiv/tos/lib/timer
-I/opt/moteiv/tos/lib/resource -I/opt/moteiv/tos/lib/sched -I/opt/moteiv/tos/li
```

DEMO - VANET Application

- Do the same instruction for the rest of the three tmotes from the step 4 to 6 by using the **TelosADC**
- But notice that changes the command line to
 - make tmote install,**2** bsl,**COMPORT**
 - make tmote install,**3** bsl,**COMPORT**
 - make tmote install,**4** bsl,**COMPORT**

DEMO - VANET Application

- Display the data on the PC
- 1. check the **motelist**

```
Administrator@mychat-937530e0 /opt/tinyos-1.x/apps/TelosADc
$ motelist
SerialNum  PortName      Description
-----
M4ACW9GU   COM7          tmote sky

Administrator@mychat-937530e0 /opt/tinyos-1.x/apps/TelosADc
$
```

- 2. **export MOTECOM=serial@COM7:57600**
- 3. **java net/tinyos.tools.Listen**

DEMO - VANET Application

```
Administrator@mychat-937530e0 /opt/tinyos-1.x/apps/TelosADc
$ export MOTECOM=serial@COM7:57600

Administrator@mychat-937530e0 /opt/tinyos-1.x/apps/TelosADc
$ java net/tinyos.tools.Listen
serial@COM7:57600: resynchronising
TOS_Msg length is invalid: header_length=260,real_length=17 ... modifying msg to
fit
Received message:07 01 08 31 FF FF FF FF 07 27 01 0F FF 0F FF 09 12
07 01 08 31 0C FF FF FF 07 27 01 0F FF 0F FF 09 12
TOS_Msg length is invalid: header_length=260,real_length=17 ... modifying msg to
fit
Received message:07 01 08 32 FF FF FF FF 07 27 01 0F FF 0F FF 09 13
07 01 08 32 0C FF FF FF 07 27 01 0F FF 0F FF 09 13
TOS_Msg length is invalid: header_length=260,real_length=17 ... modifying msg to
```

- 01 is node id
- 0F FF 0F FF 09 13 is the tmote sensor data.

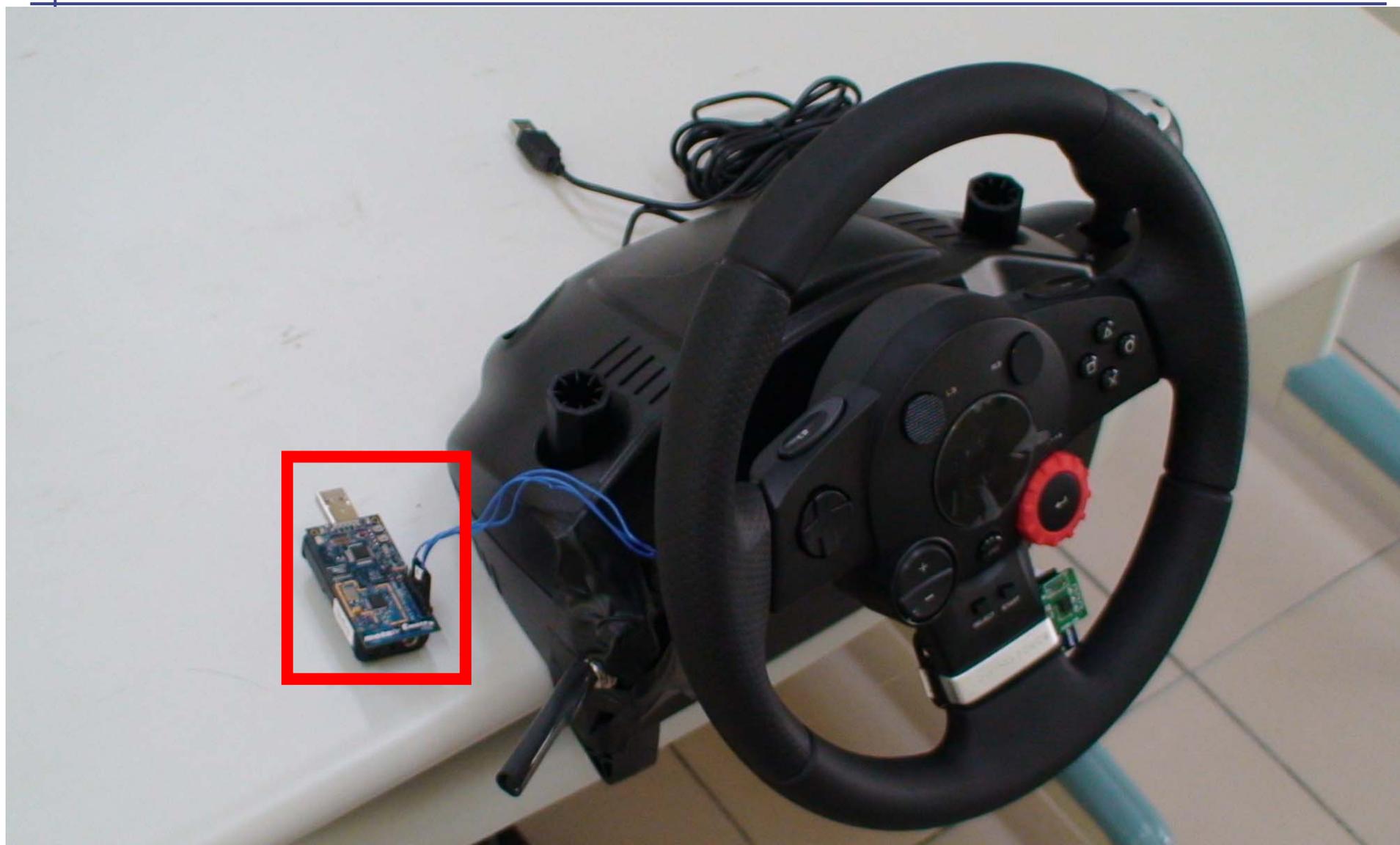
場景二：Lane-Change Warning

- 警告方式
 - 自己車內不提示
 - 語音及UI改變顯示
 - 對象為他車使用者3
- 警告範圍：本車後方、欲轉向之方向1hop範圍內之車輛
- 本場景用到的偵測器有下列兩項
 - 裝置於方向燈桿
 - 裝置於方向盤
- 使用者本身不會被告知，UI上也不會改變圖示
- 若方向盤與方向桿不一致時（如方向桿打左轉，但實際上駕駛人方向盤打右轉），則以方向盤為判斷標準

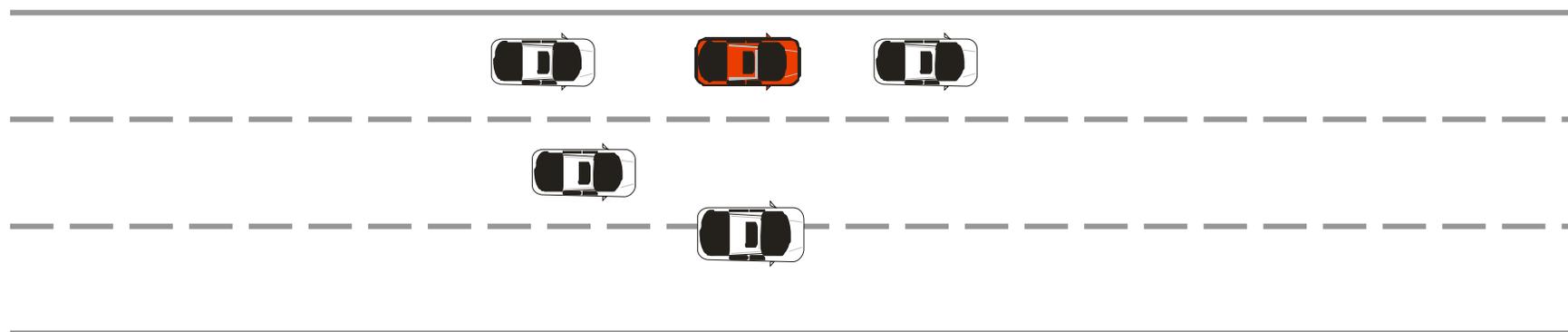
場景二：車輛改道



模擬場景之環境

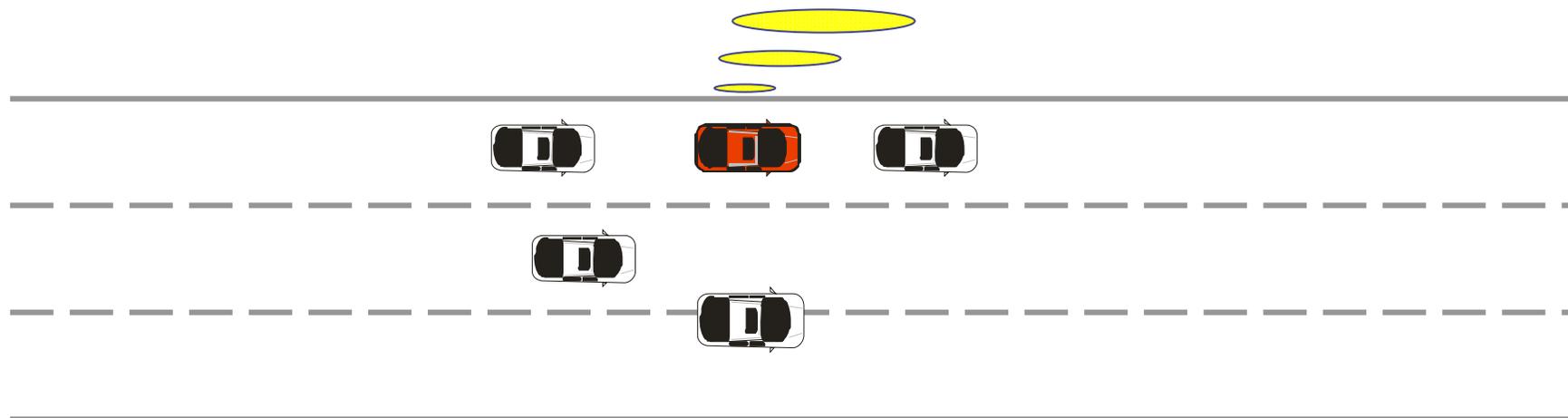


場景二：車輛改道



場景二：車輛改道

使用者想切換車道，並打右方向燈

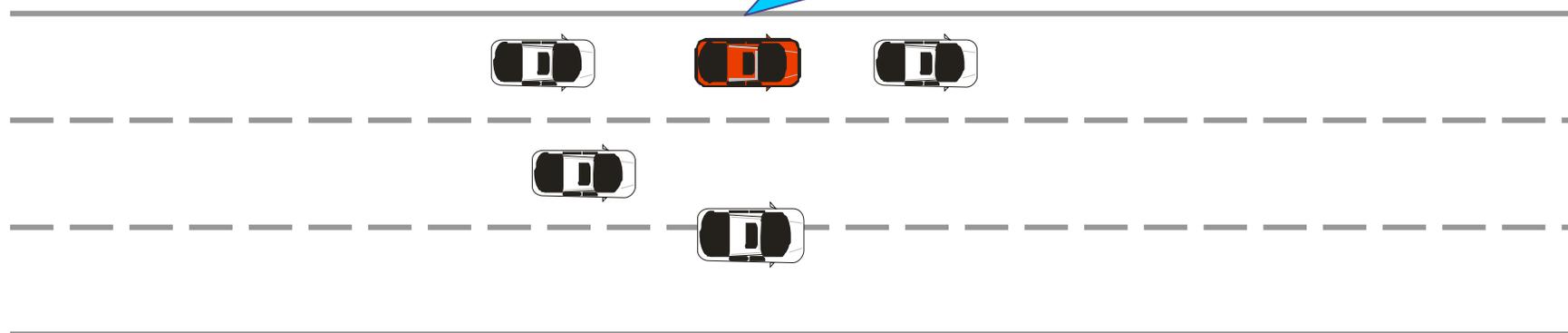


使用者想切換車道

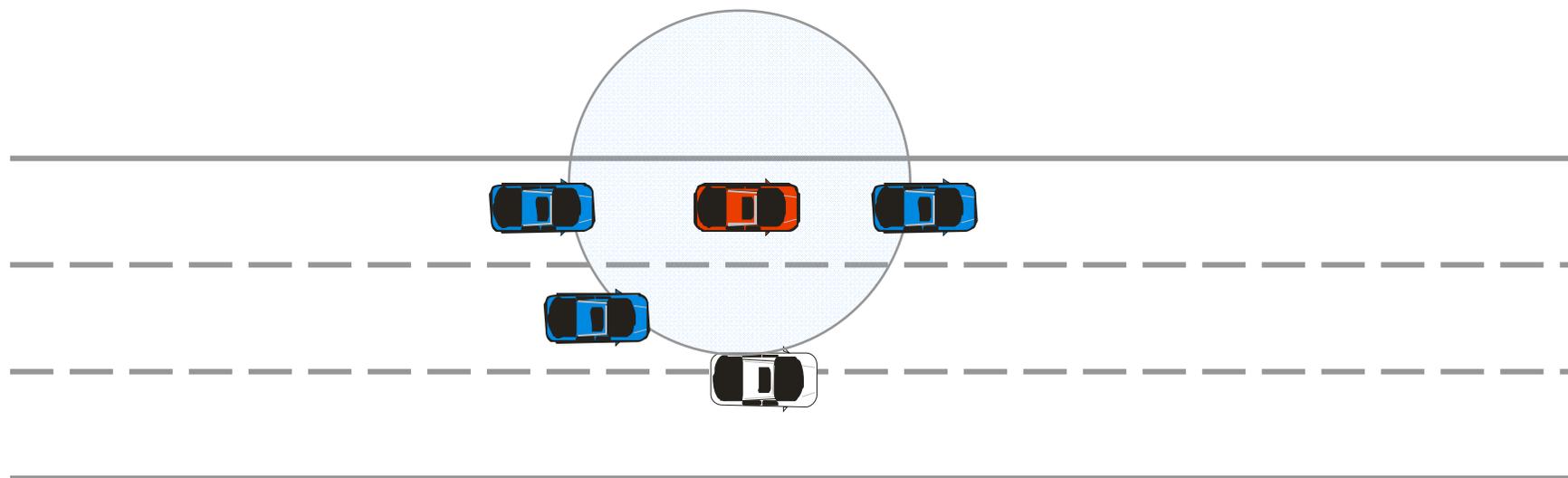


場景二：車輛改道

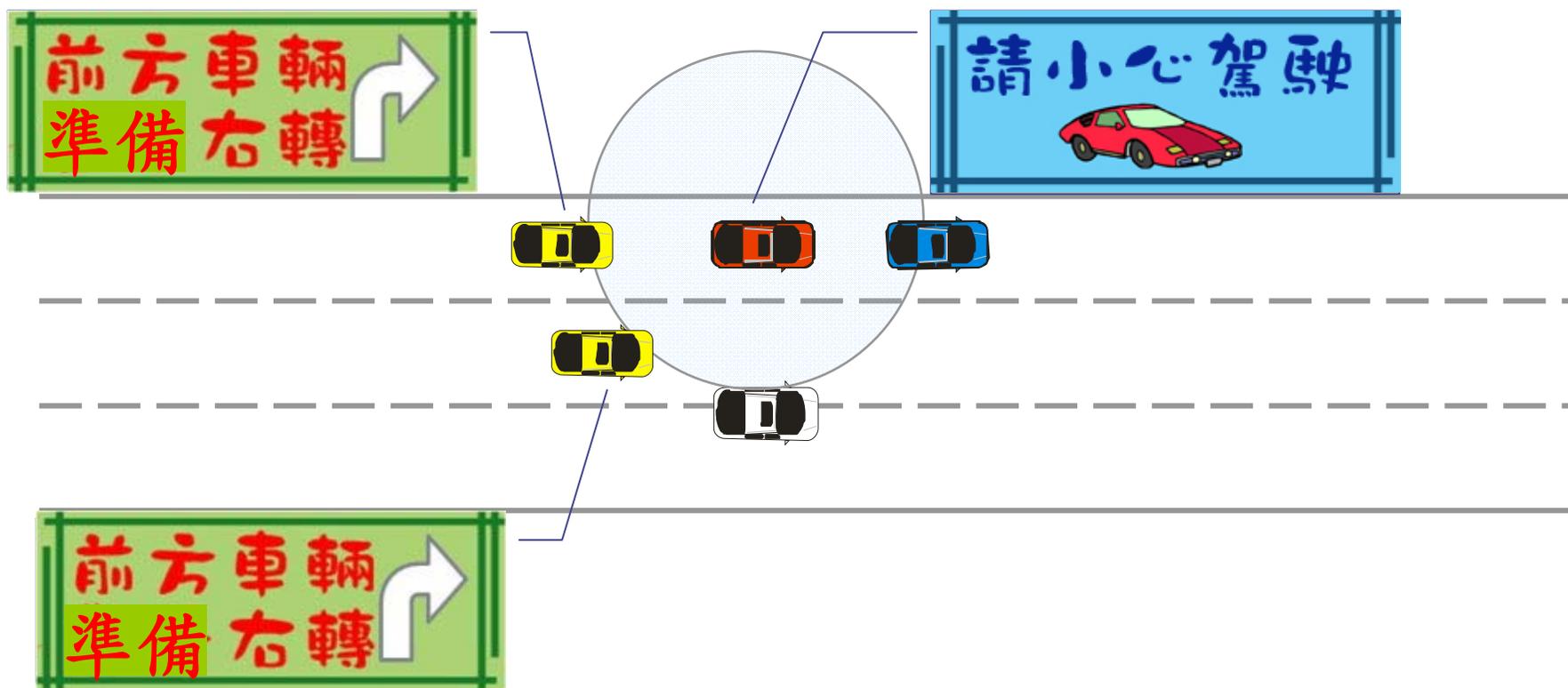
系統同時尋找右方以及後方的車輛，並通報該訊息



場景二：車輛改道



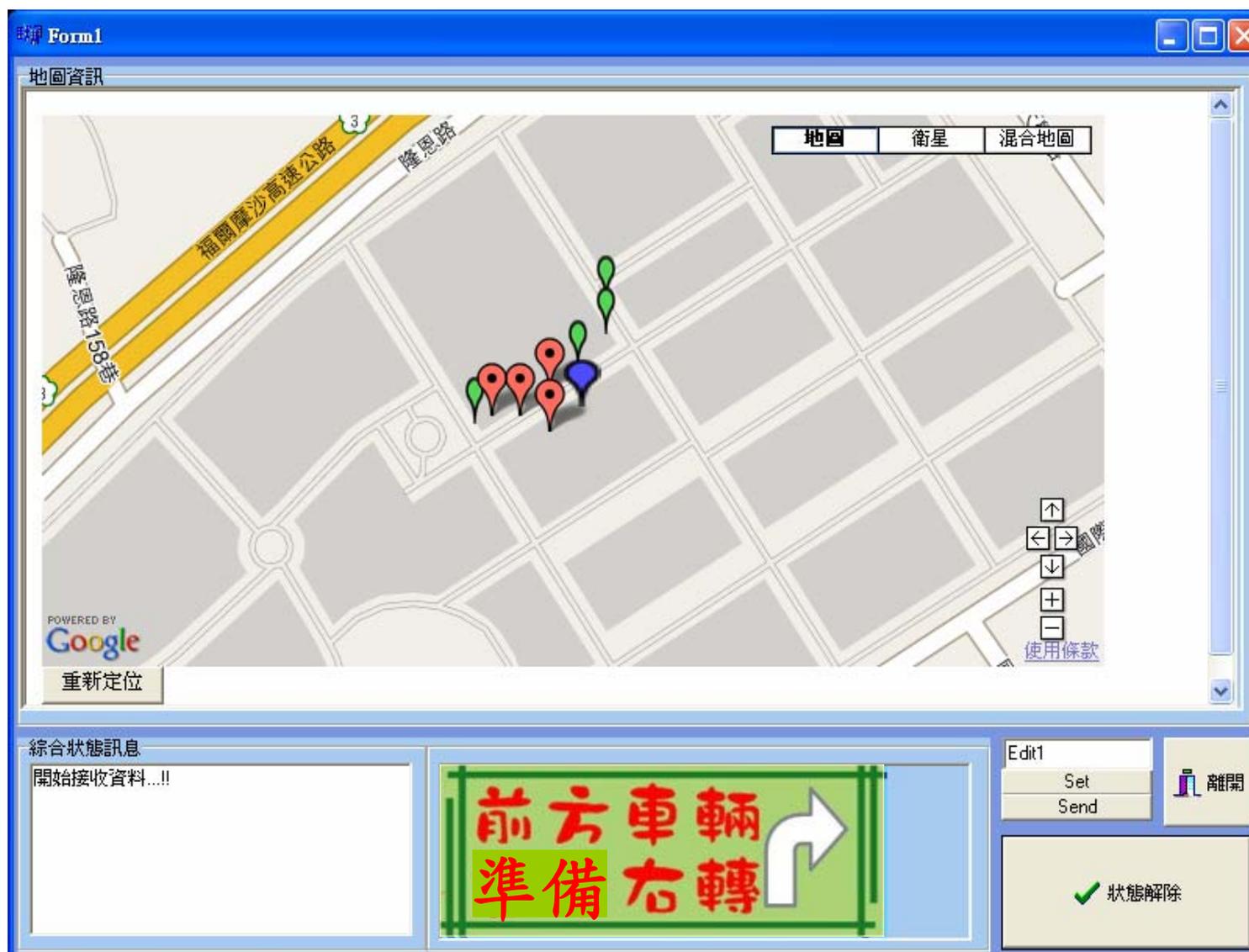
場景二：車輛改道



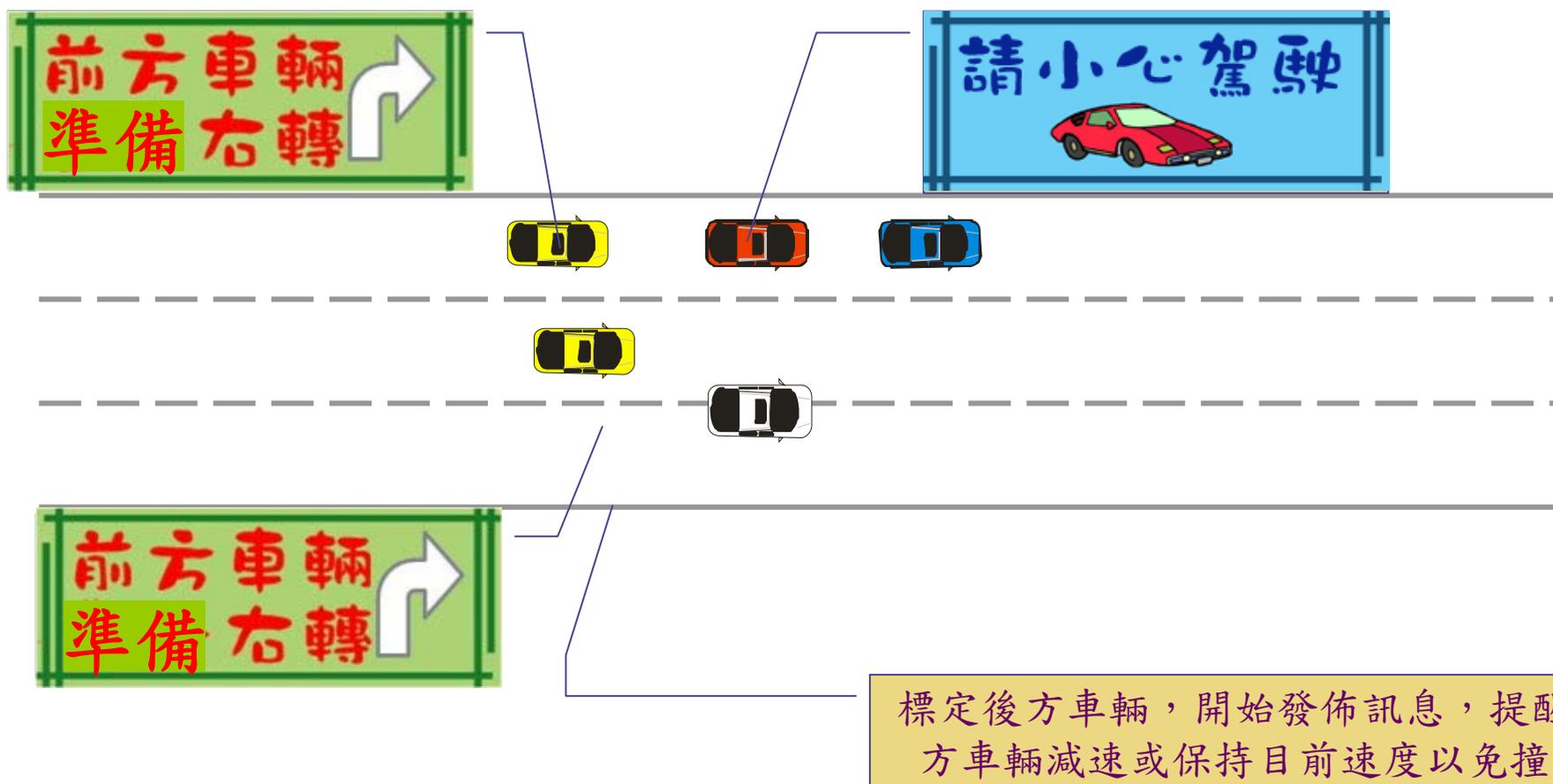
本車介面示意圖



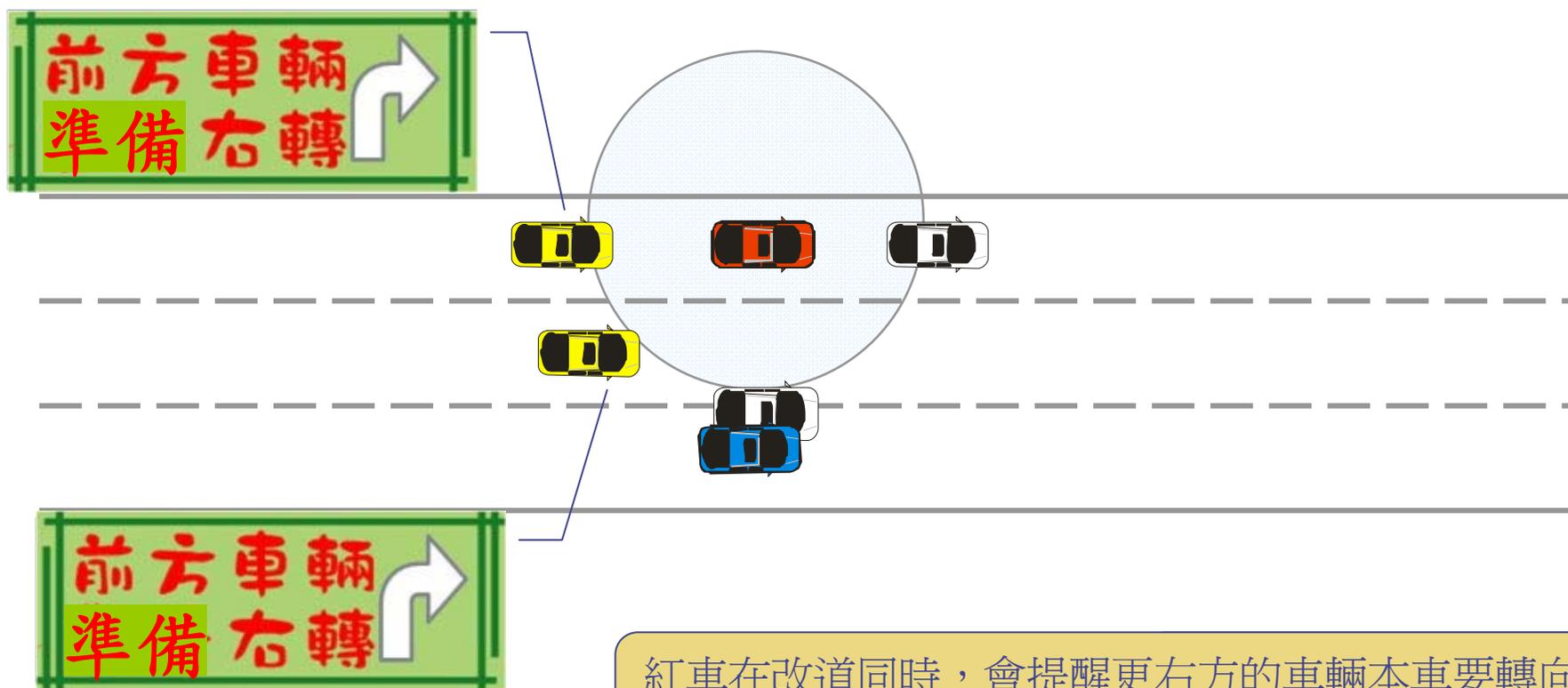
後方車輛介面示意圖



場景二：車輛改道



場景二：車輛改道



紅車在改道同時，會提醒更右方的車輛本車要轉向的動作，以免右方車輛突然改道而發生事故

急改道 (陀螺儀感測到異常急轉方向盤)

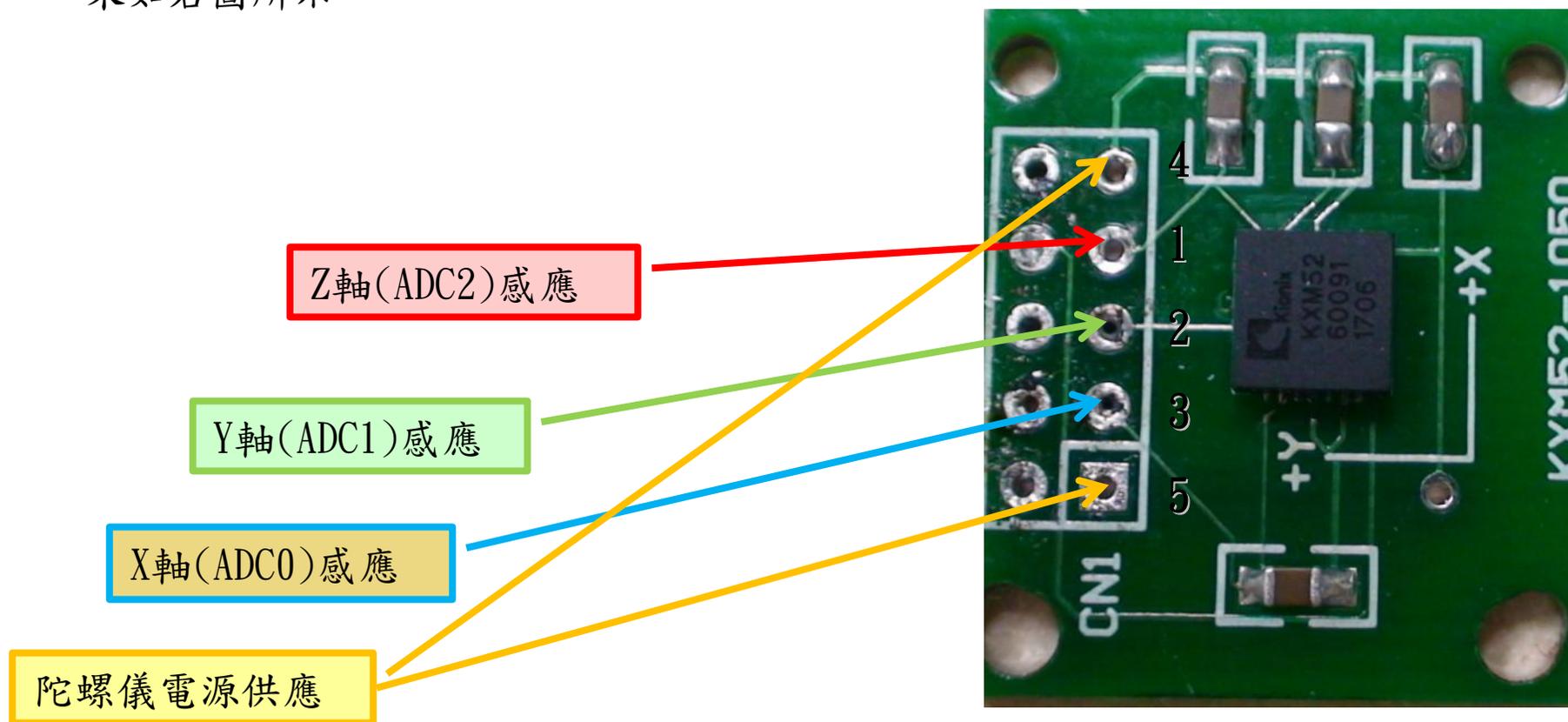


後方車輛介面變化示意圖



陀螺儀結構

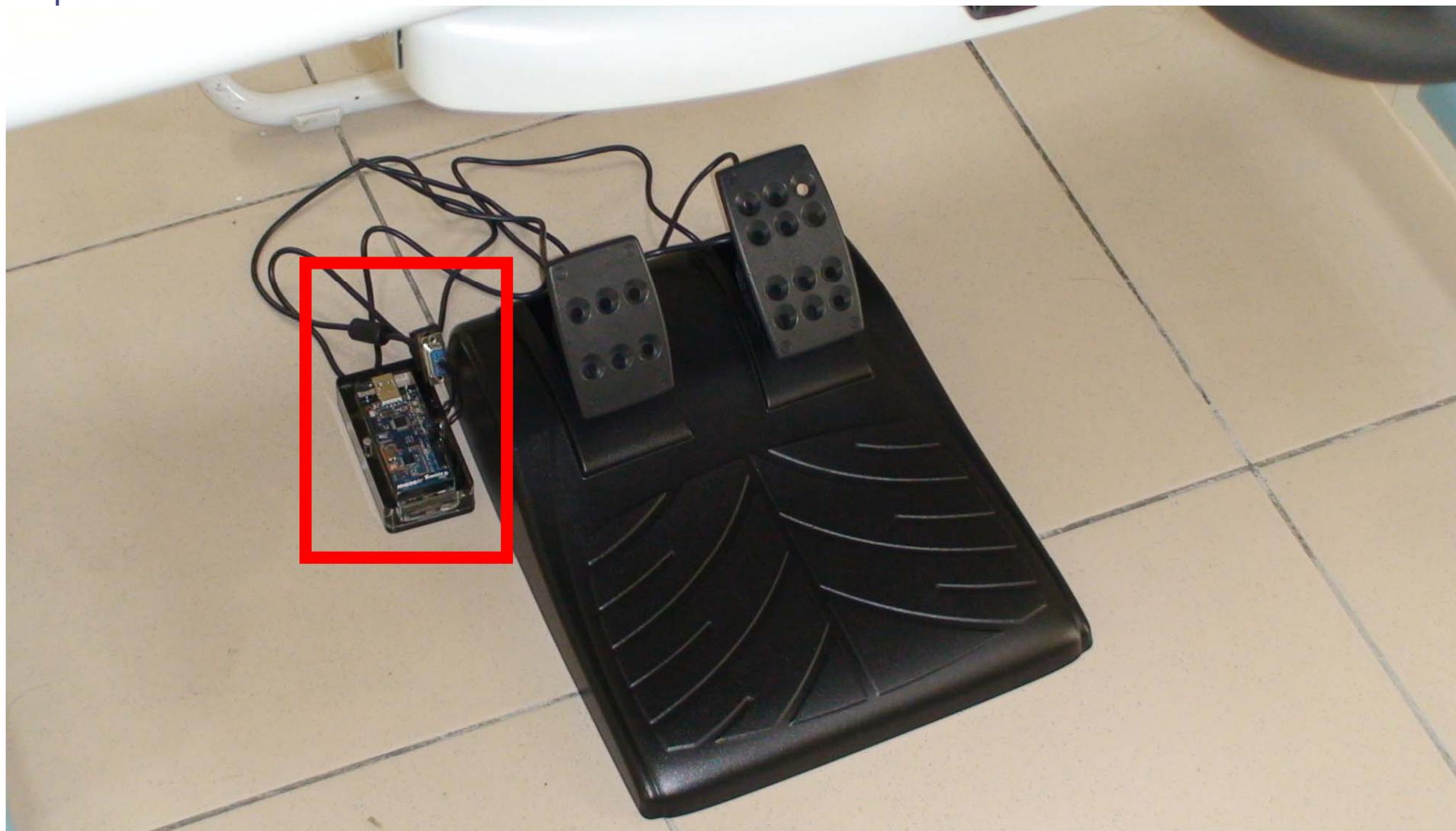
陀螺儀可感應測量X、Y、Z三軸的加速度變化，本次實驗所需接線的腳未如右圖所示



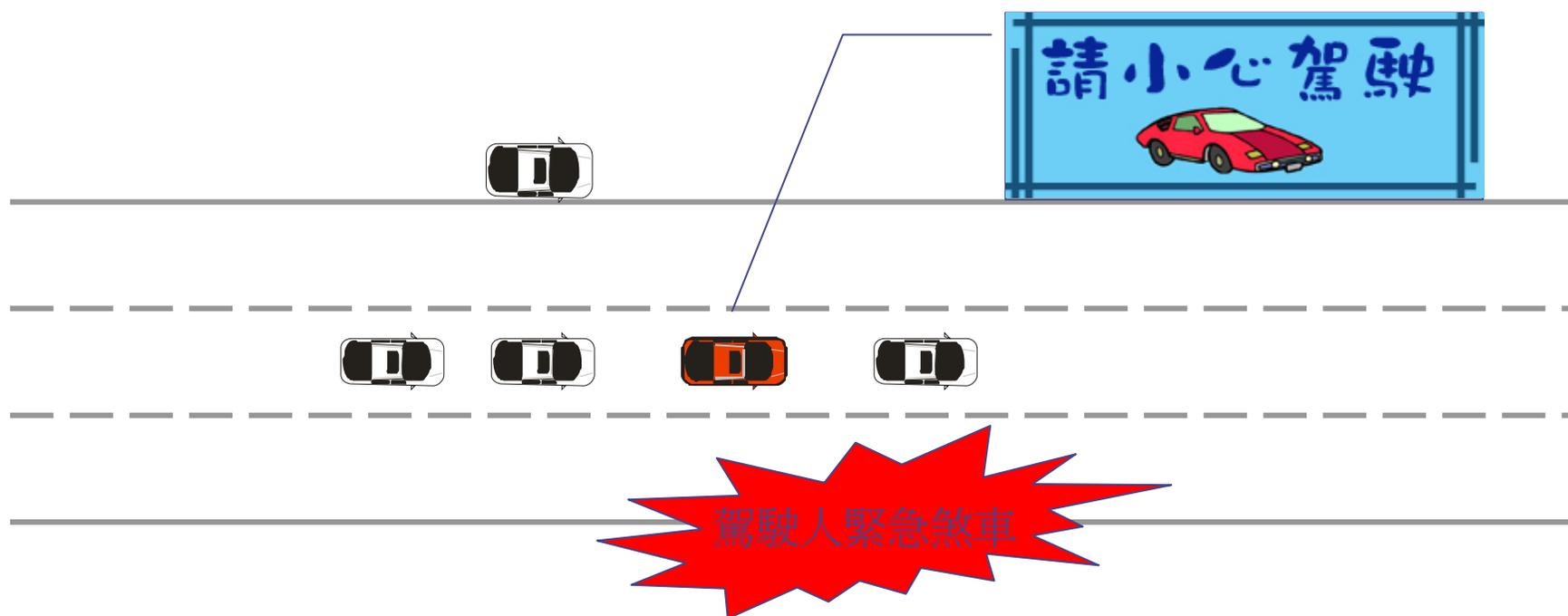
場景三：Emergency-Braking Warning

- 警告方式
 - 自己車內不提示
 - 語音、警示音效及UI改變顯示
 - 對象為他車使用者
- 警告範圍：後方 2-hops 之內的所有車輛
- 本場景為透過車上煞車踏板的感測器所讀到的數據，作為是否發出警告的依據
- 當系統偵測到駕駛人急踩，將會針對本車後方兩層之內的所有車輛發出警告

感測器與煞車結合



場景三：緊急煞車



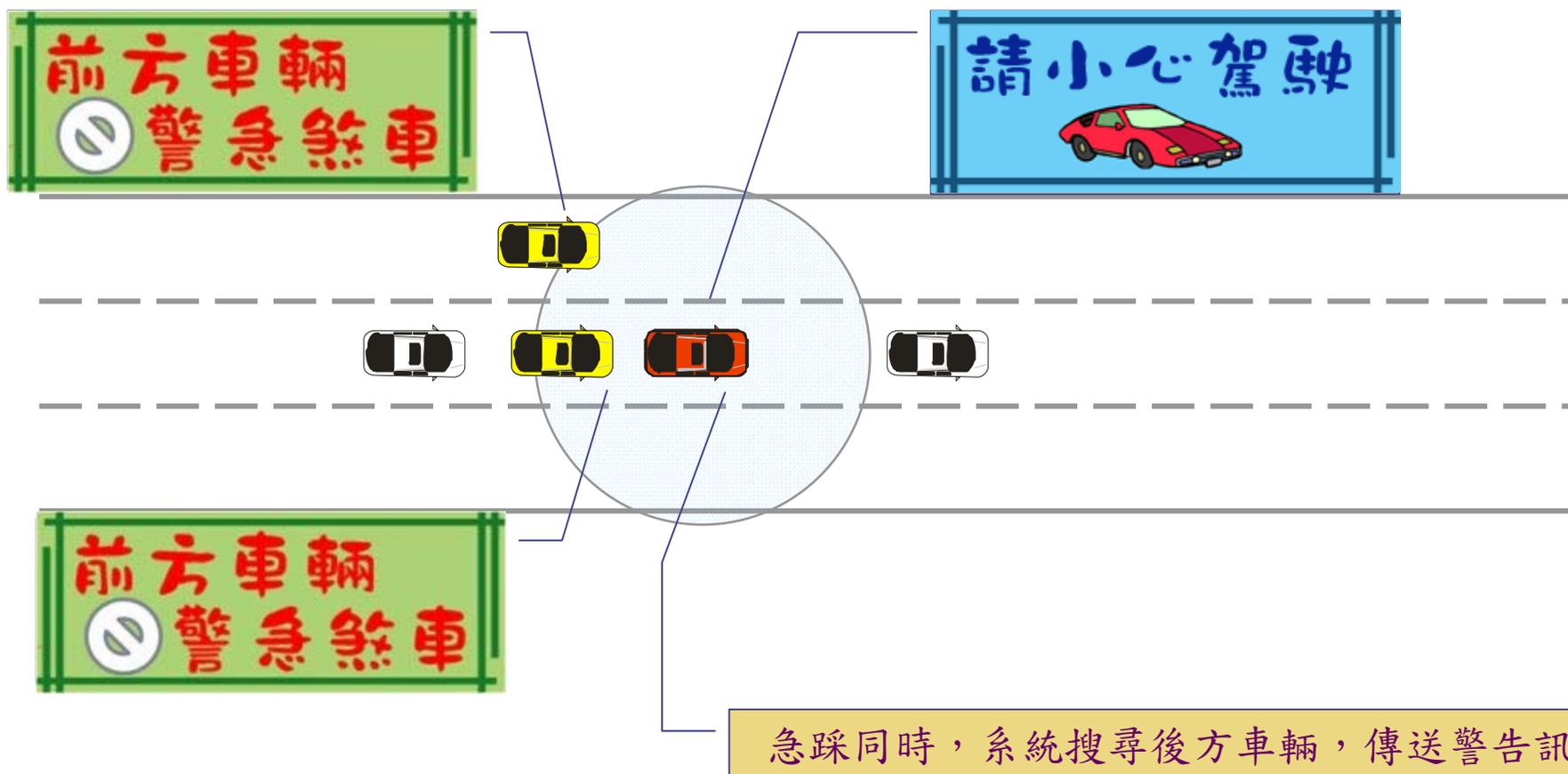
急踩煞車



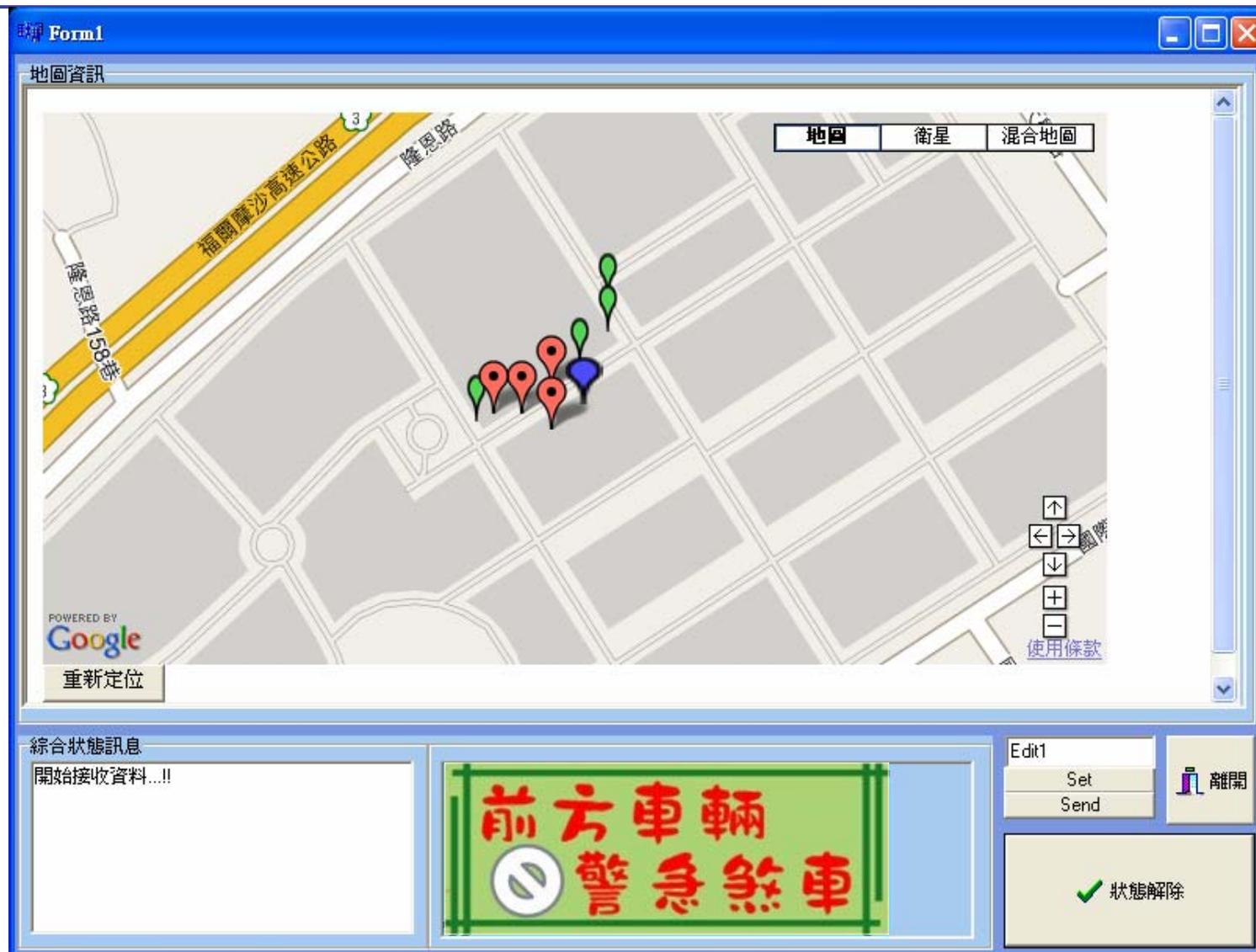
急踩煞車



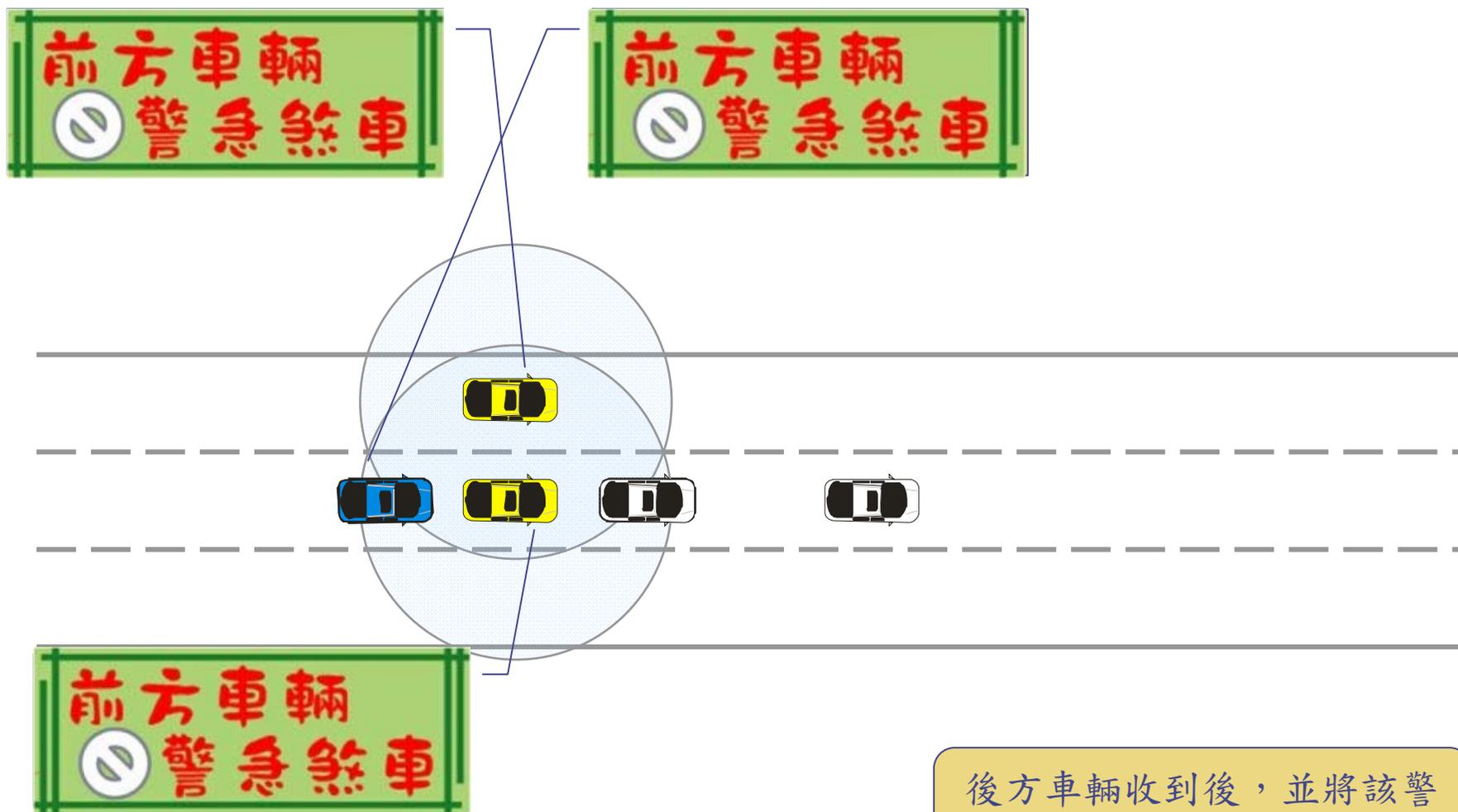
場景三：緊急煞車



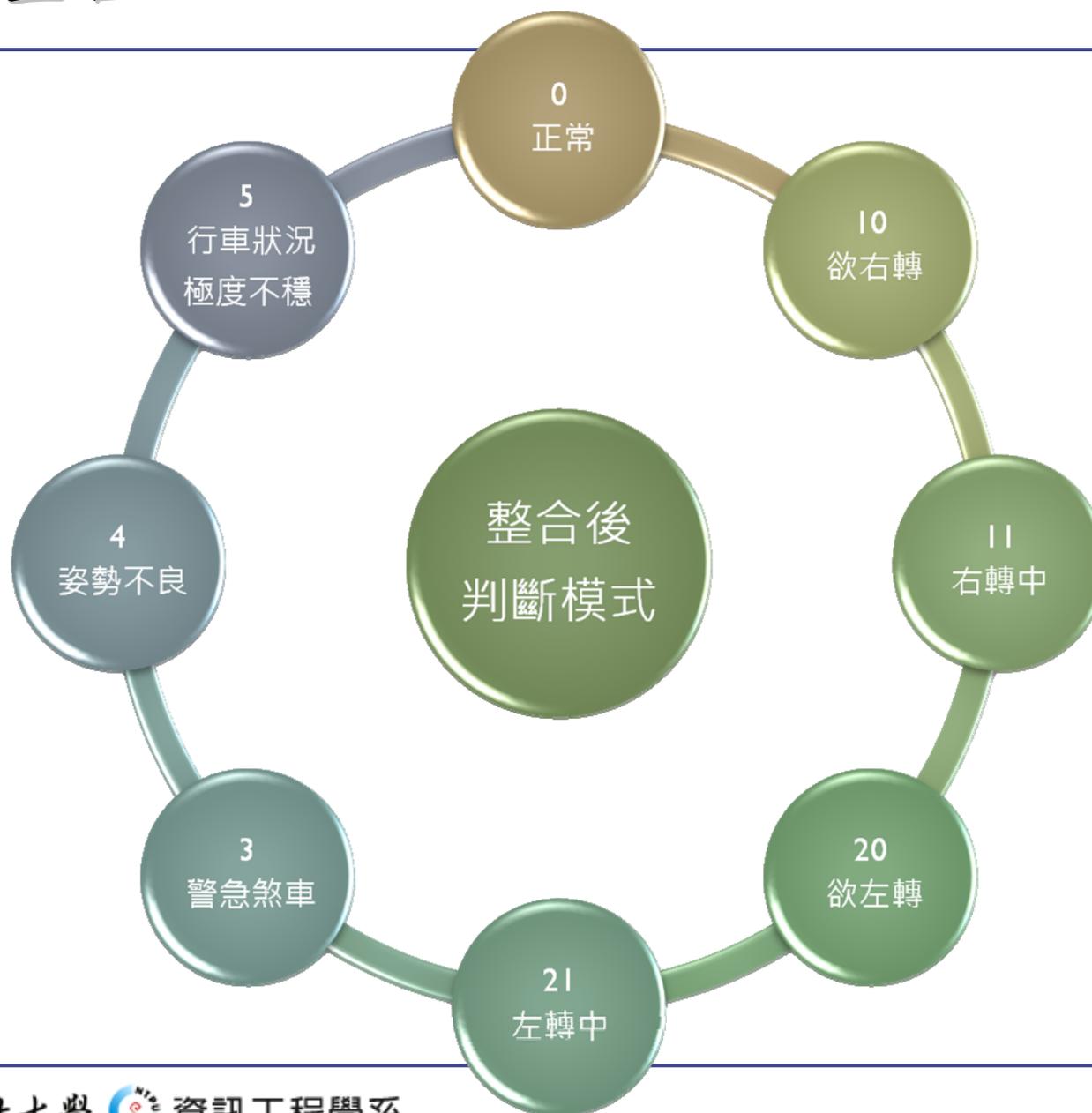
後方車輛介面顯示示意圖



場景三：緊急煞車



系統整合



整合場景

- 將上面所述個別場景，按照其裝置分類，可分為下列事件
 - 耳部(平衡)感測
 - 方向桿
 - 方向盤
 - 煞車踏板
- 另外將「皆無觸發」的「正常狀況」算為一類事件
- 將耳部感測、方向盤、煞車踏板接觸發的「最嚴重狀況」也加入

整合場景

- 將上述事件，依照嚴重性及發佈範圍由低到高分為下列等級
 - 等級0：正常狀況
 - 等級1：方向燈桿
 - 等級2：方向盤
 - 等級3：煞車踏板
 - 等級4：耳部感測
 - 等級5：最嚴重狀況
- 等級五發佈範圍以及警告方式與等級4相同
- 若僅符合等級1~等級4的任一狀況，則視為個別事件，但若為等級5、或有上述兩項或兩項以上的事件發生，則為綜合事件，則此時發佈範圍，警告方式以最高等級事件為主
- 例如：同時符合等級1、等級3和等級4時，則發佈範圍、警告方式比照等級4的方式，警告內容則為「駕駛人姿勢嚴重不良」

各類警告圖示

