

**INIKI**

**MAPSV6 Upstream UART Protocol  
User Guide**

*Ver 1.0, 12-2019*

MAPSV6 Upstream UART Protocol V1.0

- 1. UART Format N81，115200 bps Baud-Rate。
- 2. 採用 Master-Slave 半雙工通訊模式，所有資料傳遞由上位機 (Host) 完全主導及控制，MAPSV6 自身不會主動發送資料至 Host。
- 3. Host Command 大致分為三類: 取得資料 (GET data)、設定參數 (SET paras) 及中轉通訊 (PROTOCOL, bridge)，

(1) 取得資料 (GET data) 通訊格式: Host 發送: Leading+Command MAPSV6 回傳: Leading+Command+Data+Checksum	Host 發送:	Leading Code (2-Byte)	Command (2-Byte)		MAPSV6 回傳:	Leading (1-Byte)	Command (1-Byte)	Data (N-Byte)	Checksum (2-Byte)	
(2) 設定參數 (SET paras) 通訊格式: Host 發送: Leading+Command+Data+Checksum MAPSV6 回傳: Leading+Command+Result	Host 發送:	Leading Code (2-Byte)	Command (2-Byte)	Data (K-Byte)	Checksum (2-Byte)	MAPSV6 回傳:	Leading (1-Byte)	Command (1-Byte)	Result (2-Byte)	
(3) 中轉通訊 (PROTOCOL, bridge)通訊格式: Host 發送: Leading+Command+Data+Checksum MAPSV6 回傳: Leading+Command+Data+Checksum	Host 發送:	Leading Code (2-Byte)	Command (2-Byte)	Data (K-Byte)	Checksum (2-Byte)	MAPSV6 回傳:	Leading (1-Byte)	Command (1-Byte)	Data (N-Byte)	Checksum (2-Byte)

4. 通訊格式中的 Leading Code 及 Command 均為 1-Byte，Leading 固定為 0xAA；Command 範圍為 0xB0~0xCF，共可定義 32組 Function。

5. Host 發送的 Leading Code 及 Command，皆會各自加送 1-Byte 反相資料 (~Leading 及 ~Command)；MAPSV6 回傳資料無 ~Leading 及 ~Command。

6. Checksum 由 Leading Code 開始計算至所有 Data 完畢，Checksum 為 1-Byte 資料 =  $\sum (Byte-n \text{ xor } (n\%256))$ ，n= 1~N。發送 Checksum 會加送 1-Byte 反相資料 ~Checksum，做為 Double-Check。  
計算例1: 假設 Leading Code 至 Data 共計 5-Byte，Checksum= (Byte-1 xor 0x01) + (Byte-2 xor 0x02) + (Byte-3 xor 0x03) + (Byte-4 xor 0x04) + (Byte-5 xor 0x05)。  
計算例2: 假設 Leading Code 至 Data 共計 258-Byte，Checksum= (Byte-1 xor 0x01) + (Byte-2 xor 0x02) + ... + (Byte-254 xor 0xFE) + (Byte-255 xor 0xFF) + (Byte-256 xor 0x00) + (Byte-257 xor 0x01) + (Byte-258 xor 0x02)。

- 7. 原則上 MAPSV6 同一時間僅會執行一項功能任務，因此建議 Host 發送命令後，須等到回傳資料後再進行下一組命令傳送。

MAPSV6 Upstream Command Table:

Function	Host (Pi) Transmitting Command			MAPSV6 Response			Description
<b>GET_TEMP_HUM</b> Command= 0xB0  讀取溫度、濕度	Byte-01	Leading	0xAA	Byte-01 Echo Leading Byte-02 Echo Command Byte-03, Byte-04 TEMPx100 (°C) Byte-05, Byte-06 HUMx100 (%RH) Byte-07 Checksum Byte-08 ~Checksum	0xAA 0xB0 -1000 ~14000 0~10000 0~255 0~255	0xB0 0~255 0~255	1. Leading Code 為固定常數 0xAA，以Command 指定功能。 2. 反相資料 ~Leading 及 ~Command 為輔助偵錯。 3. Host 固定發送 4-Byte 命令組，MAPSV6 會檢驗收到的 4-Byte 是否符合規則，若不符合則直接跳過無任何回應。 4. 如果命令OK，固定回應2-Byte確認碼(Leading + Command)，之後為 Data，最後為 Checksum。 5. TEMPx100 為溫度值的100倍，2567表示 25.67°C 6. HUMx100 為濕度值的100倍，6789 表示 67.89% RH 7. Checksum (Byte-07) = (Byte-01 xor 1) + (Byte-02 xor 2) + ... + (Byte-06 xor 6) 8. 反相 ~Checksum (Byte-08) 做為 Double-Check。 註1. 溫度及濕度 Sensor (SHT31) 每1秒鐘取樣一次。
	Byte-02	~Leading	0x55				
	Byte-03	Command	0xB0				
	Byte-04	~Command	0x4F				
<b>GET_CO2</b> Command= 0xB1  讀取 CO2 濃度	Byte-01	Leading	0xAA	Byte-01 Echo Leading Byte-02 Echo Command Byte-03, Byte-04 CO2 (ppm) Byte-05, Byte-06 AVE_CO2 (ppm) Byte-07 Checksum Byte-08 ~Checksum	0xAA 0xB1 0~65535 0~65535 0~255 0~255	0xB1 0~255 0~255	1. Host 發送及MAPSV6回應同上。 2. CO2 資料為 UINT16 = (Byte-04 x 256) + Byte-03，467表示467ppm。 3. AVE_CO2 資料為 UINT16 = (Byte-06 x 256) + Byte-05，501表示501ppm，此數據為最後1分鐘的量測平均值。 4. Checksum 及 ~Checksum 同上。 註1. CO2 Sensor (S8 LP) 每4秒鐘取樣一次。 註2. AVE_CO2 為1分鐘的平均值(含最近的CO2量測值)，此數值由 Firmware 計算得之。
	Byte-02	~Leading	0x55				
	Byte-03	Command	0xB1				
	Byte-04	~Command	0x4E				

Page: 3

				Byte-23, Byte24 Byte-25, Byte-26 Byte-27, Byte-28 Byte-29, Byte-30 Byte-31, Byte-32 Byte-33, Byte-34 Byte-35, Byte36 Byte-37, Byte-38 Byte-39, Byte-40 Byte-41, Byte-42 Byte-43, Byte-44 Byte-45, Byte-46 Byte-47 Byte-48	Illuminance (Lux) Color Temperature (°K) CH_R (RAW) CH_G (RAW) CH_B (RAW) CH_C (RAW) PM1.0_AE (ug/m3) PM2.5_AE (ug/m4) PM10.0_AE (ug/m5) PM1.0_SP (ug/m3) PM2.5_SP (ug/m4) PM10.0_SP (ug/m5) Checksum ~Checksum	0~65535 0~65535 0~65535 0~65535 0~65535 0~65535 0~65535 0~65535 0~65535 0~65535 0~65535 0~255 0~255	
<b>GET_INFO_VERSION</b> Command= 0xB6 讀取 Firmware 版本	Byte-01 Byte-02 Byte-03 Byte-04	Leading ~Leading Command ~Command	0xAA 0x55 0xB6 0x49	Byte-01 Byte-02 Byte-03, Byte04 Byte-05 Byte-06	Echo Leading Echo Command VERSION Checksum ~Checksum	0xAA 0xB6 0~65535 0~255 0~255	1. Host 發送及MAPSV6回應同上。 2. 回應的 VERSION 資料為 UINT16 = (Byte-04 x 256) + Byte-03，1102 表示 V1.102。 3. Checksum 及 ~Checksum 同上。
<b>GET_INFO_RUNTIME</b> Command= 0xB7 讀取系統 Run-Time	Byte-01 Byte-02 Byte-03 Byte-04	Leading ~Leading Command ~Command	0xAA 0x55 0xB7 0x48	Byte-01 Byte-02 Byte-03, Byte04 Byte-05 Byte-06 Byte-07 Byte-08 Byte-09	Echo Leading Echo Command RT_DAY RT_HOUR RT_MIN RT_SEC Checksum ~Checksum	0xAA 0xB7 0~65535 0~23 0~59 0~59 0~255 0~255	1. Host 發送及MAPSV6回應同上。 2. RT_DAY 資料為 UINT16 = (Byte-04 x 256) + Byte-03，開機後累積工作時間: 日。 3. RT_HOUR 資料為 UINT8，開機後累積工作時間: 時。 4. RT_MIN 資料為 UINT8，開機後累積工作時間: 分。 5. RT_SEC 資料為 UINT8，開機後累積工作時間: 秒。 6. Checksum 及 ~Checksum 同上。 註1. 此RUNTIME 為上電開機後的累積工作時間，提供 "日、時、分、秒" 的個別數據。 註2. 每次重新上電時，RUNTIME 均會清除為0。 註3. 此RUNTIME 配合 ERROR_LOG 的資料，可得出各Sensor 的失誤率，進一步評估各Sensor 的可靠性，做為是否檢查、維修或更換的參考依據。
<b>GET_INFO_ERROR_LOG</b> Command= 0xB8 讀取各 Sensor 發生Protocol Error 的累計次數	Byte-01 Byte-02 Byte-03 Byte-04	Leading ~Leading Command ~Command	0xAA 0x55 0xB8 0x47	Byte-01 Byte-02 Byte-03, Byte04 Byte-05, Byte-06 Byte-07, Byte-08 Byte-09, Byte-10 Byte-11, Byte-12 Byte-13, Byte-14 Byte-15 Byte-16	Echo Leading Echo Command ERROR_TEMP_HUM ERROR_CO2 ERROR_TVOC ERROR_LIGHT ERROR_PMS ERROR_RTC Checksum ~Checksum	0xAA 0xB8 0~65535 0~65535 0~65535 0~65535 0~65535 0~65535 0~255 0~255	1. Host 發送及MAPSV6回應同上。 2. ERROR_TEMP_HUM 資料為 UINT16，溫濕度 Sensor 開機後的累積錯誤次數。 3. ERROR_CO2 資料為 UINT16，CO2 Sensor 開機後的累積錯誤次數。 4. ERROR_TVOC 資料為 UINT16，TVOC Sensor 開機後的累積錯誤次數。 5. ERROR_LIGHT 資料為 UINT16，LIGHT Sensor 開機後的累積錯誤次數。 6. ERROR_PMS 資料為 UINT16，PMS Sensor 開機後的累積錯誤次數。 7. ERROR_RTC 資料為 UINT16，RTC 開機後的累積錯誤次數。 8. Checksum 及 ~Checksum 同上。 註1. 此ERROR_LOG 為統計上電開機後各Sensor，發生 Protocol Error 的累積次數。 註2. 每次重新上電時，ERROR_LOG 均會全部清除為0。 註3. 配合RUNTIME 的資料，可評估各Sensor 的可靠性。 註4. 累計次數達 65535 即停止再計數。
<b>GET_INFO_SENSOR_POR</b> Command= 0xB9 讀取各 Sensor 開機時狀態	Byte-01 Byte-02 Byte-03 Byte-04	Leading ~Leading Command ~Command	0xAA 0x55 0xB9 0x46	Byte-01 Byte-02 Byte-03 Byte-04	Echo Leading Echo Command POR_TEMP_HUM POR_CO2	0xAA 0xB9 0 / 1 0 / 1	1. Host 發送及MAPSV6回應同上。 2. POR_TEMP_HUM 資料為 UINT16，溫濕度 Sensor 開機後的累積錯誤次數。 3. POR_CO2 資料為 UINT8，CO2 Sensor 開機時的狀態，0=錯誤，1=正常。 4. POR_TVOC 資料為 UINT8，TVOC Sensor 開機時的狀態，0=錯誤，1=正常。 5. POR_LIGHT 資料為 UINT8，LIGHT Sensor 開機時的狀態，0=錯誤，1=正常。 6. POR_PMS 資料為 UINT8，PMS Sensor 開機時的狀態，0=錯誤，1=正常。 7. POR_RTC 資料為 UINT8，RTC 開機時的狀態，0=錯誤，1=正常。 8. Checksum 及 ~Checksum 同上。

				Byte-05 Byte-06 Byte-07 Byte-08 Byte-09 Byte-10	POR_TVOC POR_LIGHT POR_PMS POR_RTC Checksum ~Checksum	0 / 1 0 / 1 0 / 1 0 / 1 0~255 0~255	註1. 檢查每次重新上電時，各Sensor 的初始化是否正常。	
GET_RTC_DATE_TIME Command= 0xBA  讀取 RTC 日期/時間	Byte-01 Byte-02 Byte-03 Byte-04	Leading ~Leading Command ~Command	0xAA 0x55 0xBA 0x45		Byte-01 Byte-02 Byte-03 Byte-04 Byte-05 Byte-06 Byte-07 Byte-08 Byte-09 Byte-10	Echo Leading Echo Command RTC_YY RTC_MM RTC_DD RTC_hh RTC_mm RTC_ss Checksum ~Checksum	0xAA 0xBA 0~199 1~12 1~31 0~23 0~59 0~59 0~255 0~255	1. Host 發送及MAPSV6回應同上。 2. RTC_YY 資料為 UINT8，年，0~199 表示 2000~2199年。 3. RTC_MM 資料為 UINT8，月，1~12 表示 1~12月。 4. RTC_DD 資料為 UINT8，日，1~12 表示 1~12日。 5. RTC_hh 資料為 UINT8，時，1~12 表示 1~12時。 6. RTC_mm 資料為 UINT8，分，1~12 表示 1~12分。 7. RTC_ss 資料為 UINT8，秒，1~12 表示 1~12秒。 8. Checksum 及 ~Checksum 同上。 註1. 當 Pi 插入MAPSV6 時，RTC的I2C通道會自動導向 Pi，所以此命令將無效，回傳的 RTC 資料均為 0xFF。
SET_PIN_CO2_CAL Command= 0xC0  設定 CO2 Sensor 的 Calibration Pin	Byte-01 Byte-02 Byte-03 Byte-04 Byte-05 Byte-06 Byte-07 Byte-08 Byte-09 Byte-10 Byte-11	Leading ~Leading Command ~Command UNLOCK1 UNLOCK2 UNLOCK3 UNLOCK4 PIN_SETUP Checksum ~Checksum	0xAA 0x55 0xC0 0x3F 0x53 0x38 0x4C 0x50 0 / 1 0~255 0~255		Byte-01 Byte-02 Byte-03 Byte-04	Echo Leading Echo Command RESULT ~RESULT	0xAA 0xC0 0~4 251~255	1. Host 固定發送 4-Byte (Leading, ~Leading, Command, ~Command) 後，接著發送設定資料，最後再送出 Checksum 做為 MAPSV6 偵錯之用。 2. 由於 CO2_CAL Pin 信號為 CO2 Sensor Hardware Calibration 之控制信號，直接影響 Sensor 正常取樣，須確認命令無誤，所以加入 4-Byte Unlock 檢查碼。 3. PIN_SETUP 控制 CO2_CAL Pin 的輸出位準為 LOW (0) 或 HIGH (1)：必須 UNLOCK1~4 檢查碼完全無誤方可改變輸出位準，PIN_SETUP 設定值如下： PIN_SETUP = 0: CO2_CAL Pin= LOW (0) PIN_SETUP = 1: CO2_CAL Pin= HIGH (1) 4. Checksum (Byte-10) = (Byte-01 xor 1) + (Byte-02 xor 2) + ... + (Byte-09 xor 9) ~Checksum (Byte-11) 為 Checksum 反相資料可供 Double-Check。 5. MAPSV6 接收命令後，將以 RESULT (Byte-03) 回應結果如下： RESULT= 0: 接收命令無誤，正確執行動作。 RESULT= 1: Timeout Error，Host 送出的前 4-Byte 命令解碼後，後續資料未能完全接收。 RESULT= 2: Checksum Error。 RESULT= 3: Security Error，檢查 UNLOCK 4-Byte 發現錯誤。 RESULT= 4: Setup Error，PIN_SETUP 設定值錯誤，必須為 0 或 1。 6. ~RESULT 反相做為 Double-Check。 註1. CO2_CAL Pin 正常動作位準為 1，Host 必須以此命令使該 Pin 產生 Low-Pulse。 註2. Pulse 的寬度決定 S8 LP CO2 Sensor 的校正動作如下： 16 sec > Low-Pulse > 8 sec: 400ppm 校正。 Low-Pulse > 16 sec: 0ppm 校正。
SET_PIN_PMS_RESET Command= 0xC1  設定 PMS Sensor 的 Reset Pin	Byte-01 Byte-02 Byte-03 Byte-04 Byte-05 Byte-06 Byte-07 Byte-08 Byte-09 Byte-10 Byte-11	Leading ~Leading Command ~Command UNLOCK1 UNLOCK2 UNLOCK3 UNLOCK4 PIN_SETUP Checksum ~Checksum	0xAA 0x55 0xC1 0x3E 0x50 0x4D 0x53 0x33 0 / 1 0~255 0~255		Byte-01 Byte-02 Byte-03 Byte-04	Echo Leading Echo Command RESULT ~RESULT	0xAA 0xC1 0~4 251~255	1. Host 發送命令資料格式同上 "SET_PIN_CO2_CAL" 命令所述。 2. MAPSV6 接收命令回應格式同上 "SET_PIN_CO2_CAL" 命令所述。 註1. PMS_RESET Pin 正常動作位準為 1，Host 必須以此命令使該 Pin 產生 Low-Pulse，可使 PMS Module 產生 Reset，建議 Low-Pulse > 100mS。

<b>SET_PIN_PMS_SET</b> Command= 0xC2 設定 PMS Sensor 的 SET Pin	Byte-01 Byte-02 Byte-03 Byte-04 Byte-05 Byte-06 Byte-07 Byte-08 Byte-09 Byte-10 Byte-11	Leading ~Leading Command ~Command UNLOCK1 UNLOCK2 UNLOCK3 UNLOCK4 PIN_SETUP Checksum ~Checksum	0xAA 0x55 0xC2 0x3D 0x33 0x30 0x30 0x33 0 / 1 0~255 0~255	Byte-01 Byte-02 Byte-03 Byte-04	Echo Leading Echo Command RESULT ~RESULT	0xAA 0xC2 0~4 251~255	1. Host 發送命令資料格式同上 "SET_PIN_CO2_CAL" 命令所述。 2. MAPSV6 接收命令回應格式同上 "SET_PIN_CO2_CAL" 命令所述。 註1. PMS_SET Pin 正常動作位準為 1，其動作如下： PMS_SET= 1: PMS 正常工作，每秒取樣一次並主動發送量測資料。 PMS_SET= 0: PMS 進入 Idle 狀態，停止工作。
<b>SET_PIN_NBIOT_PWRKEY</b> Command= 0xC3 設定 NB-IOT 的 PWRKEY Pin	Byte-01 Byte-02 Byte-03 Byte-04 Byte-05 Byte-06 Byte-07 Byte-08 Byte-09 Byte-10 Byte-11	Leading ~Leading Command ~Command UNLOCK1 UNLOCK2 UNLOCK3 UNLOCK4 PIN_SETUP Checksum ~Checksum	0xAA 0x55 0xC3 0x3C 0x4E 0x42 0x2D 0x49 0 / 1 0~255 0~255	Byte-01 Byte-02 Byte-03 Byte-04	Echo Leading Echo Command RESULT ~RESULT	0xAA 0xC3 0~4 251~255	1. Host 發送命令資料格式同上 "SET_PIN_CO2_CAL" 命令所述。 2. MAPSV6 接收命令回應格式同上 "SET_PIN_CO2_CAL" 命令所述。
<b>SET_PIN_NBIOT_SLEEP</b> Command= 0xC4 設定 NB-IOT 的 SLEEP Pin	Byte-01 Byte-02 Byte-03 Byte-04 Byte-05 Byte-06 Byte-07 Byte-08 Byte-09 Byte-10 Byte-11	Leading ~Leading Command ~Command UNLOCK1 UNLOCK2 UNLOCK3 UNLOCK4 PIN_SETUP Checksum ~Checksum	0xAA 0x55 0xC4 0x3B 0x2D 0x49 0x4F 0x54 0 / 1 0~255 0~255	Byte-01 Byte-02 Byte-03 Byte-04	Echo Leading Echo Command RESULT ~RESULT	0xAA 0xC4 0~4 251~255	1. Host 發送命令資料格式同上 "SET_PIN_CO2_CAL" 命令所述。 2. MAPSV6 接收命令回應格式同上 "SET_PIN_CO2_CAL" 命令所述。
<b>SET_PIN_LED_ALL</b> Command= 0xC5 Enable / Disable 所有 LED	Byte-01 Byte-02 Byte-03 Byte-04 Byte-05 Byte-06 Byte-07 Byte-08 Byte-09 Byte-10 Byte-11	Leading ~Leading Command ~Command UNLOCK1 UNLOCK2 UNLOCK3 UNLOCK4 PIN_SETUP Checksum ~Checksum	0xAA 0x55 0xC5 0x3A 0x53 0x4C 0x45 0x44 0 / 1 0~255 0~255	Byte-01 Byte-02 Byte-03 Byte-04	Echo Leading Echo Command RESULT ~RESULT	0xAA 0xC5 0~4 251~255	1. Host 發送命令資料格式同上 "SET_PIN_CO2_CAL" 命令所述。 2. MAPSV6 接收命令回應格式同上 "SET_PIN_CO2_CAL" 命令所述。 註1. LED_ALL Pin 可控制 MAPSV6 板上 6 顆 LED 的 ON/OFF，其動作如下： LED_ALL= 1: 所有 LED 均正常工作點亮或閃爍，開機預設值。 LED_ALL= 0: 所有 LED 均熄滅 OFF。 註2. 將所有 LED 熄滅，可避免在黑暗環境下 LED 的亮度會影響 LIGHT Sensor 的照度測量，因為目前 LIGHT Sensor 的積分時間設定為 700mS，取樣時間為 1 sec，所以 LED 熄滅 2 sec 後的照度量測值才不會受 LED 影響。

				Byte-03 Byte-04	RESULT ~RESULT	0~4 251~255	
<b>SET_POLLING_SENSOR</b> Command= 0xC6  Enable / Disable Sensor Polling	Byte-01 Byte-02 <b>Byte-03</b> <b>Byte-04</b>  <b>Byte-05</b> <b>Byte-06</b> <b>Byte-07</b> <b>Byte-08</b> <b>Byte-09</b> <b>Byte-10</b>  Byte-11 Byte-12	Leading ~Leading <b>Command</b> <b>~Command</b>  <b>POLL_TEMP</b> <b>POLL_CO2</b> <b>POLL_TVOC</b> <b>POLL_LIGHT</b> <b>POLL_PMS</b> <b>POLL_RTC</b>  Checksum ~Checksum	0xAA 0x55 <b>0xC6</b> <b>0x39</b>  0 / 1 0 / 1 0 / 1 0 / 1 0 / 1 0 / 1  0~255 0~255				1. Host 發送命令資料格式同上 " <b>SET_PIN_CO2_CAL</b> " 命令所述。 2. POLL_TEMP 資料為 UINT8，溫濕度 Sensor，1= 正常取樣，0= 停止取樣。 3. POLL_CO2 資料為 UINT8，CO2 Sensor，1= 正常取樣，0= 停止取樣。 4. POLL_TVOC 資料為 UINT8，TVOC Sensor，1= 正常取樣，0= 停止取樣。 5. POLL_LIGHT 資料為 UINT8，LIGHT Sensor，1= 正常取樣，0= 停止取樣。 6. POLL_PMS 資料為 UINT8，PMS Sensor，1= 正常取樣，0= 停止取樣。 7. POLL_RTC 資料為 UINT8，RTC，1= 正常取樣，0= 停止取樣。 <b>註1. MAPSV6每次開機，預設所有Sensor 均為正常取樣。</b> <b>註2.停止取樣意思為:Firmware 停止該Sensor 的 Protocol Polling，輸出數據將維持最後一次取樣數據，並停止 ERROR_LOG 的計數。</b>
				Byte-01 <b>Byte-02</b> <b>Byte-03</b> <b>Byte-04</b>	Echo Leading <b>Echo Command</b> <b>RESULT</b> <b>~RESULT</b>	0xAA <b>0xC6</b> <b>0~4</b> <b>251~255</b>	
<b>SET_RTC_DATE_TIME</b> Command= 0xC7  設定 RTC 日期/時間	Byte-01 Byte-02 <b>Byte-03</b> <b>Byte-04</b>  <b>Byte-03</b> <b>Byte-04</b> <b>Byte-05</b> <b>Byte-06</b> <b>Byte-07</b> <b>Byte-08</b>  Byte-09 Byte-10	Leading ~Leading <b>Command</b> <b>~Command</b>  <b>RTC_YY</b> <b>RTC_MM</b> <b>RTC_DD</b> <b>RTC_hh</b> <b>RTC_mm</b> <b>RTC_ss</b>  Checksum ~Checksum	0xAA 0x55 <b>0xC7</b> <b>0x38</b>  0~199 1~12 1~31 0~23 0~59 0~59  0~255 0~255				1. Host 發送及MAPSV6回應同上。 2. RTC_YY 資料為 UINT8，年，0~199 表示 2000~2199年。 3. RTC_MM 資料為 UINT8，月，1~12 表示 1~12月。 4. RTC_DD 資料為 UINT8，日，1~12 表示 1~12日。 5. RTC_hh 資料為 UINT8，時，1~12 表示 1~12時。 6. RTC_mm 資料為 UINT8，分，1~12 表示 1~12分。 7. RTC_ss 資料為 UINT8，秒，1~12 表示 1~12秒。 8. Checksum 及 ~Checksum 同上。 9. MAPSV6 接收命令後，將以 RESULT (Byte-03) 回應結果如下： RESULT= 0: 接收命令無誤，正確執行動作。 RESULT= 1: Device Error，RTC未連線。 RESULT= 2: Timeout Error，Host 送出的前 4-Byte 命令解碼後，後續資料未能完全接收。 RESULT= 3: Checksum Error。 RESULT= 4: Over Date Error，Date (YY/MM/DD) 的設定數值超出指定範圍。 RESULT= 5: Over Time Error，Time (hh/mm/ss) 的設定數值超出指定範圍。 10. ~RESULT 反相做為 Double-Check。 <b>註1. 當 Pi 插入MAPSV6 時，RTC的I2C通道會自動導向 Pi，所以此命令將無效，回傳的 RESULT= 0x01。</b>
				Byte-01 <b>Byte-02</b> <b>Byte-03</b> <b>Byte-04</b>	Echo Leading <b>Echo Command</b> <b>RESULT</b> <b>~RESULT</b>	0xAA <b>0xC7</b> <b>0~5</b> <b>250~255</b>	
<b>PROTOCOL_I2C_WRITE</b> Command= 0xCA  進行 I2C 通訊 WRITE 序列	Byte-01 Byte-02 <b>Byte-03</b> <b>Byte-04</b>  <b>Byte-05</b> <b>Byte-06</b> <b>Byte-07</b> <b>Byte-08 ~ (07+N)</b>  Byte-(08+N) Byte-(09+N)	Leading ~Leading <b>Command</b> <b>~Command</b>  <b>FREQ_INDEX</b> <b>I2C_ADDRESS</b> <b>LENGTH_N</b> <b>DATA[N]</b>  Checksum ~Checksum	0xAA 0x55 <b>0xCA</b> <b>0x35</b>  0~4 0~127 1~32 <b>(0~255) x N</b>  0~255 0~255				1. Host 發送命令前 4-Byte 資料格式同上。 2. FREQ_INDEX (Byte-05) 可設定 I2C SCL 的工作頻率如下： FREQ_INDEX= 0: 使用目前工作頻率，不變更。 FREQ_INDEX= 1: SCL= 400Khz。 FREQ_INDEX= 2: SCL= 200Khz。 FREQ_INDEX= 3: SCL= 100Khz。 FREQ_INDEX= 4: SCL= 50Khz。 3. I2C_ADDRESS (Byte-06) 為 7-Bit I2C Address，MAPSV6 會強制 bit7= 0。 4. LENGTH_N (Byte-07) 指定 WRITE DATA 的長度，數值必須 1~32 之間。 5. 起始位置為 Byte-08 的 DATA[N] 為準備寫入 I2C 裝置的 N-Byte 資料陣列，其長度 N 由 Byte-07 的 LENGTH_N 所指定。 6. Checksum (Byte-(08+N)) = (Byte-01 xor 1) + (Byte-02 xor 2) + ... + (Byte-(07+N) xor (07+N)) ~Checksum (Byte-(09+N)) 為 Checksum 反相資料可供 Double-Check。 7. MAPSV6 接收命令後，將以 RESULT (Byte-03) 回應結果如下： RESULT= 0: 接收命令無誤，正確執行動作。 RESULT= 1: Device Error，接收命令無誤，但是 Write to I2C Device 時回應為 NAK。 RESULT= 2: Timeout Error，Host 送出的前 4-Byte 命令解碼後，後續資料未能完全接收。 RESULT= 3: Checksum Error。 RESULT= 4: Over FREQ_INDEX Error，FREQ_INDEX 的設定數值超出指定範圍。 RESULT= 5: Over LENGTH_N Error，LENGTH_N 的設定數值超出指定範圍。 8. ~RESULT 反相做為 Double-Check。
				Byte-01 <b>Byte-02</b> <b>Byte-03</b> <b>Byte-04</b>	Echo Leading <b>Echo Command</b> <b>RESULT</b> <b>~RESULT</b>	0xAA <b>0xCA</b> <b>0~5</b> <b>250~255</b>	



							註1. 當發生 <b>RESULT= 2~5</b> 的錯誤時，不會執行 <b>Write to I2C Device</b> 的動作。	
<b>PROTOCOL_I2C_READ</b> <b>Command= 0xCB</b>  進行 I2C 通訊 READ 序列	Byte-01 Byte-02 <b>Byte-03</b> <b>Byte-04</b>  <b>Byte-05</b> <b>Byte-06</b> <b>Byte-07</b> Byte-08 Byte-09	Leading ~Leading <b>Command</b> <b>~Command</b>  <b>FREQ_INDEX</b> <b>I2C_ADDRESS</b> <b>LENGTH_N</b> Checksum ~Checksum	0xAA 0x55 <b>0xCB</b> <b>0x34</b>  0~4 0~127 1~32 0~255 0~255		Byte-01 <b>Byte-02</b> <b>Byte-03</b> <b>Byte-04</b> <b>Byte-05 ~ (04+N)</b> Byte-(05+N) Byte-(06+N)	Echo Leading <b>Echo Command</b>  <b>RESULT</b> <b>~RESULT</b> <b>DATA[N]</b> Checksum ~Checksum	0xAA <b>0xCB</b>  <b>0~5</b> <b>250~255</b> <b>(0~255) x N</b> 0~255 0~255	1. Host 發送命令前 4-Byte 資料格式同上。 2. FREQ_INDEX (Byte-05) 可設定 I2C SCL 的工作頻率如下： FREQ_INDEX= 0: 使用目前工作頻率，不變更。 FREQ_INDEX= 1: SCL= 400Khz。 FREQ_INDEX= 2: SCL= 200Khz。 FREQ_INDEX= 3: SCL= 100Khz。 FREQ_INDEX= 4: SCL= 50Khz。 3. I2C_ADDRESS (Byte-06) 為 7-Bit I2C Address，MAPSV6 會強制 bit7= 0。 4. LENGTH_N (Byte-07) 指定 READ DATA 的長度，數值必須 1~32 之間。 5. Checksum (Byte-08) = (Byte-01 xor 1) + (Byte-02 xor 2) + ... + (Byte-07 xor 7) ~Checksum (Byte-09) 為 Checksum 反相資料可供 Double-Check。 6. MAPSV6 接收命令後，將以 RESULT (Byte-03) 回應結果如下： RESULT= 0: 接收命令無誤，正確執行動作。 RESULT= 1: Device Error，接收命令無誤，但是 Write to I2C Device 時回應為 NAK。 RESULT= 2: Timeout Error，Host 送出的前 4-Byte 命令解碼後，後續資料未能完全接收。 RESULT= 3: Checksum Error。 RESULT= 4: Over FREQ_INDEX Error，FREQ_INDEX 的設定數值超出指定範圍。 RESULT= 5: Over LENGTH_N Error，LENGTH_N 的設定數值超出指定範圍。 7. ~RESULT 反相做為 Double-Check。 8. 起始位置為 Byte-05 的 DATA[N] 為從 I2C 裝置讀取的 N-Byte 資料陣列，其長度 N 由 Host 命令中 Byte-07 的 LENGTH_N 所指定。 9. Checksum (Byte-(05+N)) = (Byte-01 xor 1) + (Byte-02 xor 2) + ... + (Byte-(04+N) xor (04+N)) ~Checksum (Byte-(06+N)) 為 Checksum 反相資料可供 Double-Check。 註1. 只有 <b>RESULT= 0</b> 正確執行動作時，會回傳後續的讀取資料 <b>DATA[N]</b> 及 <b>Checksum</b> ，其餘的錯誤發生時 ( <b>RESULT= 1~5</b> ) 均不會再回應後續的資料。 註2. 通常 <b>I2C READ</b> 前會先執行 <b>I2C WRITE</b> 以指定準備讀取的位址或暫存器。
<b>PROTOCOL_UART_BEGIN</b> <b>Command= 0xCC</b>  設定各個預留UART Port 格式	Byte-01 Byte-02 <b>Byte-03</b> <b>Byte-04</b>  <b>Byte-05</b> <b>Byte-06</b> <b>Byte-07</b> Byte-08 Byte-09	Leading ~Leading <b>Command</b> <b>~Command</b>  <b>UART_PORT</b> <b>BAUD_INDEX</b> <b>UART_FORMAT</b> Checksum ~Checksum	0xAA 0x55 <b>0xCC</b> <b>0x33</b>  0~2 0~4 0~5 0~255 0~255		Byte-01 <b>Byte-02</b> <b>Byte-03</b> <b>Byte-04</b>	Echo Leading <b>Echo Command</b>  <b>RESULT</b> <b>~RESULT</b>	0xAA <b>0xCC</b>  <b>0~5</b> <b>250~255</b>	1. Host 發送命令前 4-Byte 資料格式同上。 2. UART_PORT (Byte-05) 指定欲設定的UART port，MAPSV6 板上提供三組 UART 如下： UART_PORT= 0: 板上保留的 NB-IOT UART port。 UART_PORT= 1: 板上預留的 AUX-UART1 port (Hardware UART)。 UART_PORT= 2: 板上預留的 AUX-UART2 port (Software UART)。 3. BAUD_INDEX (Byte-06) 指定 Baudrate 如下： BAUD_INDEX= 0: Baud 9600 bps。 BAUD_INDEX= 1: Baud 19200 bps。 BAUD_INDEX= 2: Baud 38400 bps。 BAUD_INDEX= 3: Baud 57600 bps。 BAUD_INDEX= 4: Baud 115200 bps。 4. UART_FORMAT (Byte-07) 指定Format 如下： UART_FORMAT= 0: N81。 UART_FORMAT= 1: N71。 UART_FORMAT= 2: E81。 UART_FORMAT= 3: E71。 UART_FORMAT= 4: O81。 UART_FORMAT= 5: O71。 5. Checksum (Byte-08) = (Byte-01 xor 1) + (Byte-02 xor 2) + ... + (Byte-07 xor 7) ~Checksum (Byte-09) 為 Checksum 反相資料可供 Double-Check。 6. MAPSV6 接收命令後，將以 RESULT (Byte-03) 回應結果如下： RESULT= 0: 接收命令無誤，正確執行動作。 RESULT= 1: Timeout Error，Host 送出的前 4-Byte 命令解碼後，後續資料未能完全接收。 RESULT= 2: Checksum Error。 RESULT= 3: Over UART_PORT Error: UART_PORT 的設定數值超出指定範圍。 RESULT= 4: Over BAUD_INDEX Error，BAUD_INDEX 的設定數值超出指定範圍。 RESULT= 5: Over UART_FORMAT Error，UART_FORMAT 的設定數值超出指定範圍。 7. ~RESULT 反相做為 Double-Check。



							註1. 當發生 RESULT= 1-5 的錯誤時，不會執行 UART_BEGIN 的設定動作。
PROTOCOL_UART_TX_RX Command= 0xCD  進行 UART 一次批量半雙工通訊	Byte-01	Leading	0xAA				1. Host 發送命令前 4-Byte 資料格式同上。 2. UART_PORT (Byte-05) 指定欲通訊的UART port，MAPSV6 板上提供三組 UART 如下： UART_PORT= 0: 板上保留的 NB-IOT UART port。 UART_PORT= 1: 板上預留的 AUX-UART1 port (Hardware UART)。 UART_PORT= 2: 板上預留的 AUX-UART2 port (Software UART)。 3. TX_LENGTH_N 為UINT16，指定發送 TX_DATA 的長度，數值必須 1~1024 之間。 4. RX_LENGTH_N 為UINT16，指定接收RX_DATA 的長度，數值必須 0~1024 之間。 5. RX_TIMEOUT 為 UINT16，指定 MAPSV6 接收RX_DATA 的TIMEOUT，定義如下： 當MAPSV6接收完整命令資料後會立即對指定的 UART port 發送N-Byte 的TX_DATA[N]， 等 TX_DATA[N] 發送完畢後，開始等待接收指定UART port 回傳K-Byte 的 RX_DATA[K]， 從開始等待直到接收RX_DATA[K] 完畢所需的最長預估時間即為TIMEOUT (mS)，等完成後， MAPSV6才會回應 Host。 6. TX_DATA[N] 為N-Byte 的TX資料陣列，啟始位置為 Byte-12，長度由TX_LENGTH_N指定。 7. Checksum 及 ~Checksum 同前。 8. MAPSV6 接收命令後，將以 RESULT (Byte-03) 回應結果如下： RESULT= 0: 接收命令無誤，正確執行動作。 RESULT= 1: Timeout Error，Host 送出的前 4-Byte 命令解碼後，後續資料未能完全接收。 RESULT= 2: Checksum Error。 RESULT= 3: Over UART_PORT Error: UART_PORT 的設定數值超出指定範圍。 RESULT= 4: Over TX_LENGTH_N Error，TX_LENGTH_N 的設定數值超出指定範圍。 RESULT= 5: Over RX_LENGTH_N Error，RX_LENGTH_N 的設定數值超出指定範圍。 RESULT= 6: RX Timeout Error，MAPSV6 接收 Device RX 時間超出 RX_TIMEOUT 的設定值。 9. ~RESULT 反相做為 Double-Check。 10. RX_DATA[K] 為K-Byte 的RX資料陣列，啟始位置為回傳資料的Byte-12，資料長度 由 Host 命令中的 RX_LENGTH_K 所指定。 11. 回應的 Checksum 及 ~Checksum 同前。 註1. 如果 Host 的命令序列總長度超過 64-Byte，發送前請先執行 "SET_POLLING_SENSOR" 命令，設定POL_CO2= 0 及 POL_LIGHT= 0，暫時停止二者需佔用長時間的取樣動作， 讓 MAPSV6 可即時接收 Host 命令列無誤。 註2. 只有 RESULT= 0 正確執行動作時，會回傳後續的資料RX_DATA[K] 及 Checksum， 其餘的錯誤發生時 (RESULT= 1~6) 均不會再回應後續的資料。 唯一的例外為:當命令指定RX_LENGTH_K= 0時，表示無RX_DATA[K]，自然不會 回傳後續的資料RX_DATA[K] 及 Checksum。
	Byte-02 Byte-03 Byte-04	~Leading Command ~Command	0x55 0xCD 0x32				
	Byte-05 Byte-06, Byte-07 Byte-08, Byte-09 Byte-10, Byte-11 Byte-12 ~ (11+N)	UART_PORT TX_LENGTH_N RX_LENGTH_K RX_TIMEOUT TX_DATA[N]	0~2 1~1024 0~1024 0~65535 (0~255) x N				
	Byte-(12+N) Byte-(12+N)	Checksum ~Checksum	0~255 0~255				
				Byte-01 Byte-02 Byte-03 Byte-04 Byte-05 ~ (04+N)	Echo Leading Echo Command RESULT ~RESULT RX_DATA[K]	0xAA 0xCD 0~6 249~255 (0~255) x N	
				Byte-(05+N) Byte-(06+N)	Checksum ~Checksum	0~255 0~255	

Figure.1 單一命令發送 "查詢(GET\_xxx)" 及 "設定(SET\_xxx)" 功能:

1. Host 送出命令序列時必須將所有 Byte 一次同時送出，Byte 之間的間隔時間必須小於 3mS，MAPSV6 Firmware 設定接收每個Byte的等待時間為3mS (RX Byte Timeout)。
2. MAPSV6 接收完整命令序列後，將延遲一段時間 (Echo Time) 才送出回應資料；此 Echo Time 包含: 資料確認、準備，或剛好內部忙碌進行 Sensor Protocol，MAPSV6 幾乎都能在 5mS 之內回應，  
偶而 (約1%機率) 會超出 5mS，最長時間為 30mS。
3. Echo Time 是 Host 設定等待回應 Timeout 的重要依據，此 Echo Time 會因連線方式略有差異如下:  
(1) 直接 Hardware UART 連線: 最大值即為上述討論的 30mS。  
(2) 透過 USB-UART Bridge 連線: 因為 USB interrupt interval 時間約 8.4mS，如果 TX 及 RX 時間點剛好都在後緣發送，則 Echo Time 還須再加 8.4mSx2= 16.8mS。



Figure.2 多組命令同時發送 "查詢(GET\_xxx)" 及 "設定(SET\_xxx)" 功能:

- 1. Host 同時送出多組命令可有效改進通訊效率，以 USB Bridge 連線時更明顯 (每多加一組命令至少可省略一組 USB interrupt interval 約 8.4mS)。
- 2. MAPSV6 Firmware 的機制為一次只處理一組命令，完成後才會解碼下一組命令。
- 3. Host 一次發送多組命令時，必須控制總長度不得超過 64 Bytes。



Figure.3 I2C Bridge Protocol:

- 1. Echo Time 同上，I2C WRITE 及 READ 的資料長度均不可超過 32 Bytes。
- 2. Host 可根據 SCL 頻率及 WRITE/READ 總 Byte 數 (含 Addressing Byte) 約略算出 Transfer Time。
- 3. Verify Time 固定約 1~3mS。

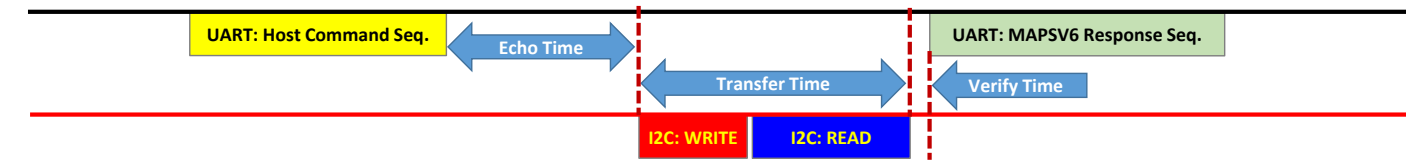


Figure.4 UART Bridge Protocol:

- 1. Echo Time 同上，UART TX 及 RX 的資料長度均不可超過 1024 Bytes。
- 2. Host 可根據 Baud-Rate、TX/RX 總 Byte 數及 Device 最長的 Response Time，約略算出 Transfer Time，並且將參數 "Host Specified TIMEOUT" 傳送 MAPSV6，做為 Device Timeout 之依據。
- 3. Verify Time 固定約 1~3mS。

