# 臺北市停車管理工程處

# 停車場剩餘車位資訊上傳申請規範及說明

## 通訊協定：(範例詳附件)

現場設備送出

| 01H | 10H | 00H 00H | 00H 02H | 04H | 總車位數 | 剩餘車位數 | CRC(16) |
|---|---|---|---|---|---|---|---|
| ID | 指令碼 | 起始位址 | DATA WORD 數 | DATA BYTE 數 | 2byte | 2 byte | 2byte |

資料收集系統回應(資訊中心)

| 01H | 10H | 00H 00H | 00H 02H | CRC(16) |
|---|---|---|---|---|
| ID | 指令碼 | 起始位址 | DATA WORD 數 | 2 byte |

通訊規則：

1.現場設備**每 30 秒**依據通訊協定送出目前剩餘車位數，當送出車位數資料後，判斷中心端是否有回傳正常回應指令,若 5 秒後中心端尚未回應指令則再依據通訊協定傳送剩餘車位數(參照車位上傳流程架構圖)。

2. 現場設備無實際剩餘車位數字者,**每 30 秒**依據通訊協定送出目前剩餘車位狀態，其中剩餘車位數欄位資料填入 ffff 表示紅、ffee 表示黃、ffdd 表示綠。

## 注意事項

1.本案相關流程架構圖、通訊協定，如附錄。

2.申請時提供：

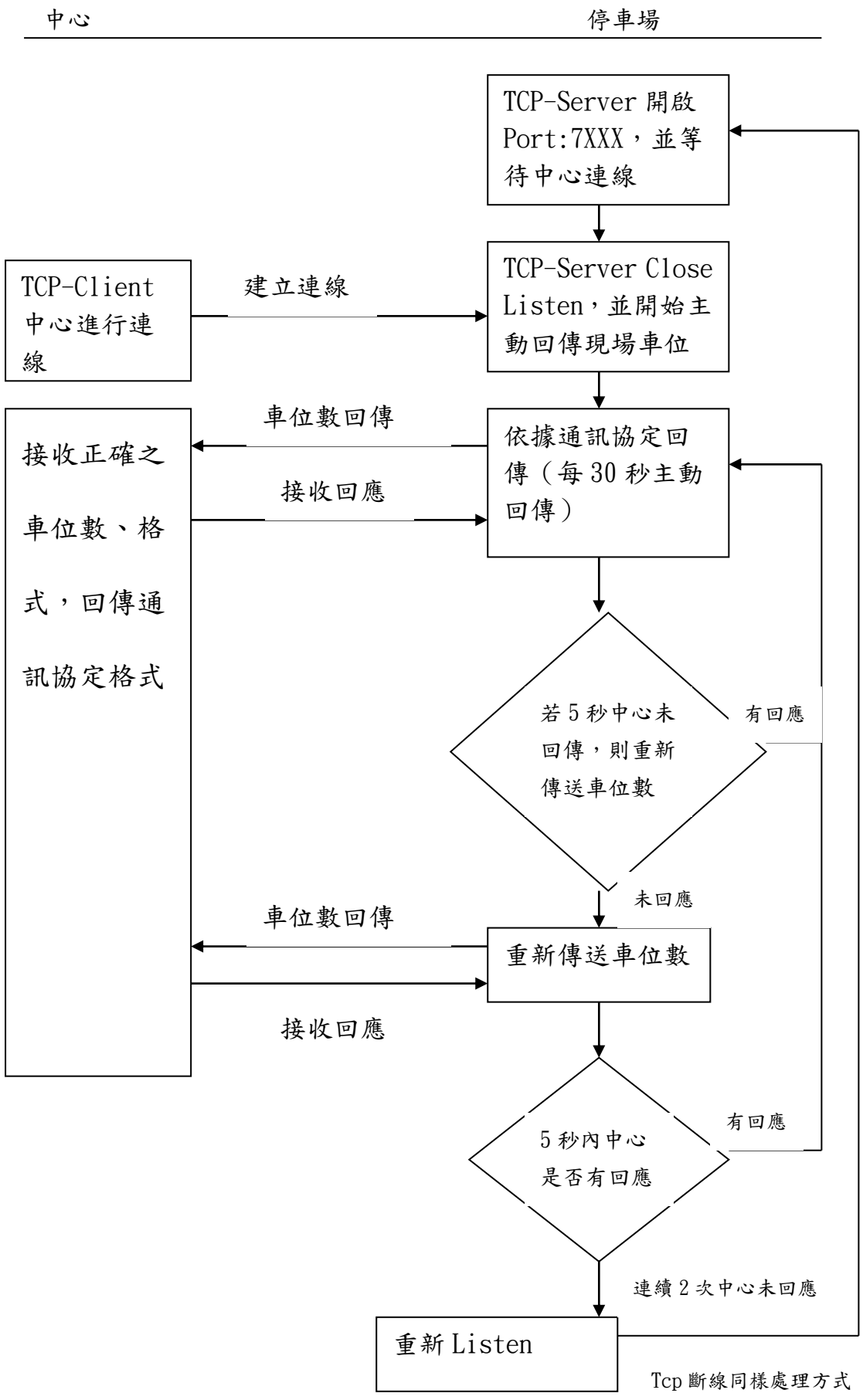
(1)臺北市停車場登記證影本。

(2)停車場限高。

(3)停車場管理室聯絡電話。

3.本處聯絡窗口：機電科 康益銘 27590666 轉 6522

# 車位上傳流程架構圖


中心
停車場
TCP-Server 開啟 Port:7XXX，並等待中心連線
TCP-Server Close Listen，並開始主動回傳現場車位
TCP-Client 中心進行連線
建立連線
車位數回傳
依據通訊協定回傳（每 30 秒主動回傳）
接收回應
接收正確之車位數、格式，回傳通訊協定格式
若 5 秒中心未回傳，則重新傳送車位數
有回應
未回應
車位數回傳
重新傳送車位數
接收回應
5 秒內中心是否有回應
有回應
連續 2 次中心未回應
重新 Listen
Tcp 斷線同樣處理方式

# 剩餘車位資訊連線範例

車場送出

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01H | 10H | 00H<br>00H | 00H<br>02H | 04H | 總車位數 | 剩餘車位數 | CRC(16) |
| ID | 指令碼 | 起始位址 | DATA WORD 數 | DATA BYTE 數 | 2byte | 2 byte | 2byte |

資料收集系統回應

| | | | | |
|---|---|---|---|---|
| 01H | 10H | 00H 00H | 00H 02H | CRC(16) |
| ID | 指令碼 | 起始位址 | DATA WORD 數 | 2 byte |

以下是車場連線資料的範例，其中第一行是車場所送出，第二行是系統收到後回應的資料。

PARKID:[0004] Send Data:[剩餘車位數 10，總車位數 100]
Send Data: 01 10 00 00 00 02 04 00 64 00 0a 32 77
遠端回應 Received Data: 01 10 00 00 00 02 41 c8
PARKID:[0005] Send Data:[剩餘車位數 18，總車位數 100]
Send Data: 01 10 00 00 00 02 04 00 64 00 12 32 7d
遠端回應 Received Data: 01 10 00 00 00 02 41 c8
PARKID:[0001] Send Data:[剩餘車位數 8，總車位數 100]
Send Data: 01 10 00 00 00 02 04 00 64 00 08 b3 b6
遠端回應 Received Data: 01 10 00 00 00 02 41 c8
PARKID:[0003] Send Data:[剩餘車位數 6，總車位數 100]
Send Data: 01 10 00 00 00 02 04 00 64 00 06 32 72
遠端回應 Received Data: 01 10 00 00 00 02 41 c8
PARKID:[0002] Send Data:[剩餘車位狀態為紅，總車位數 100]
Send Data: 01 10 00 00 00 02 04 00 64 ff ff b2 75
遠端回應 Received Data: 01 10 00 00 00 02 41 c8

# CRC16 計算程式

以下 C 語言程式片段為 CRC16 for MODBUS 的計算方式。

```
unsigned short CRC16 ( puchMsg, usDataLen ) /* The function returns the
CRC as a unsigned short type */
unsigned char *puchMsg ; /* message to calculate CRC upon */
unsigned short usDataLen ; /* quantity of bytes in message */
{
    unsigned char uchCRCHi = 0xFF ; /* high byte of CRC initialized */
    unsigned char uchCRCLo = 0xFF ; /* low byte of CRC initialized */
    unsigned uIndex ; /* will index into CRC lookup table */
    while (usDataLen--) /* pass through message buffer */
    {
        uIndex = uchCRCLo ^ *puchMsgg++ ; /* calculate the CRC */
        uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex} ;
        uchCRCHi = auchCRCLo[uIndex] ;
    }
    return (uchCRCHi << 8 | uchCRCLo) ;
}

/* Table of CRC values for high-order byte */
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1,
0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80,
0x41, 0x01, 0xC0,
```

```
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00,
0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1,
0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81,
0x40
} ;
/* Table of CRC values for low-order byte */
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7,
0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB,
0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE,
0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2,
0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32,
0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E,
0xFE, 0xFA, 0x3A,
```

```
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B,
0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27,
0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1,
0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD,
0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8,
0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4,
0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94,
0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59,
0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D,
0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83,
0x41, 0x81, 0x80,
0x40
};
```