

臺北捷運環狀線（第一階段）
機電系統工程、軌道工程、自動收費系統工程
TAIPEI CIRCULAR LINE (PHASE 1)
E&M SYSTEM, TRACK WORKS, AFC SYSTEM
CF610/CF611/CF617

附件 二十
PGIS ICD 文件

APPENDIX 20
PGIS ICD

CMFT PGIS 通訊規格

| | | |
|---------|-----|-------------|
| 核准： | 審查： | 製作：謝瑞文 |
| 日期： | 日期： | 日期：106/3/24 |
| 版本：03.8 | | |

CMFT PGIS 通訊規格

| | | |
|-------|-----------------------------------|---|
| 1 | 介面與通訊協定 | 1 |
| 1.1 | 介面 | 1 |
| 1.1.1 | CMFT 之 MOXA CN2650-8-2AC..... | 1 |
| 1.1.2 | PGIS 之 MOXA CP-132EL-I-DB25M..... | 1 |
| 1.2 | 通訊協定 | 3 |
| 1.2.1 | Header..... | 3 |
| 1.2.2 | Packet..... | 3 |
| 1.2.3 | CRC Code..... | 4 |

1 介面與通訊協定

因介面採用單向通訊，也就是 CMFT 傳遞給 PGIS，其間不能有任何訊息從 PGIS 傳遞給 CMFT，所以由 CMFT 傳送而 PGIS 只有接收。

1.1 介面

通訊介面：RS-422

Baud：19200bps

同位檢查：n

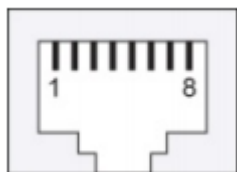
資料位元：8

停止位元：1

接線硬體，只有 CMFT 的 TX 而 PGIS 的 RX 接腳與地線

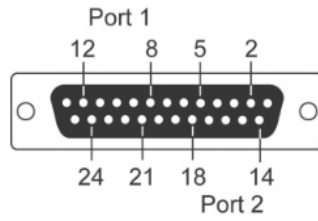
1.1.1 CMFT 之 MOXA CN2650-8-2AC

CMFT 之 MOXA CN2650-8-2AC



| Pin | Transmission Signals | | |
|-----|----------------------|----------------------|-----------|
| | RS-232 | RS-422/ RS-485-4w | RS-485-2w |
| 1 | DSR (in) | — | — |
| 2 | RTS (out) | TxD + | — |
| 3 | GND | GND | GND |
| 4 | TxD (out) | TxD - | — |
| 5 | RxD (in) | TxD + | Data+ |
| 6 | DCD (in) | TxD - | Data- |
| 7 | CTS (in) | — | — |
| 8 | DTR (out) | — | — |

1.1.2 PGIS 之 MOXA CP-132EL-I-DB25M



| Pin | Transmission Signals | | |
|-----|----------------------|---------------|---------------|
| | RS-422 | 4-wire RS-485 | 2-wire RS-485 |
| 2 | P2: TxD-(A) | P2: TxD-(A) | — |
| 3 | P2: SGND | — | — |
| 4 | — | — | — |
| 5 | P2: TxD+(B) | P2: TxD+(B) | — |
| 8 | — | — | — |
| 9 | P1: TxD-(A) | P1: TxD-(A) | — |
| 10 | — | — | — |
| 11 | — | — | — |
| 12 | P1: TxD+(B) | P1: TxD+(B) | — |
| 14 | — | — | — |
| 15 | P2: RxD-(A) | P2: RxD-(A) | P2: Data-(A) |
| 16 | — | — | — |
| 17 | — | — | — |
| 18 | P2: RxD+(B) | P2: RxD+(B) | P2: Data+(B) |
| 21 | P1: RxD-(A) | P1: RxD-(A) | P1: Data-(A) |
| 22 | P1: SGND | — | — |
| 23 | — | — | — |
| 24 | P1: RxD+(B) | P1: RxD+(B) | P1: Data+(B) |

1.2 通訊協定

SDS Message Definition

| | | |
|--------|--------|----------|
| Header | Packet | CRC Code |
|--------|--------|----------|

1.2.1 Header

| From CMFT To PGIS | | | | |
|-------------------|--------------------|--------|--------|----------------|
| Variable | | Values | | |
| Name | Bytes | No. | Choice | Name/Mean |
| Msg | 1 | 1 | | Data Source |
| | ASCII(ISO/IEC 646) | 2 | 0 | CMFT |
| MsgType | 1 | 3 | B | MsgType 以 B 表示 |

1.2.2 Packet

封包長度為動態長度

會發送目前所有月台,每個月台 20 分鐘內即將到站的列車群,如 20 分鐘內無即將到站的列車,會給最近一輛到此月台的列車

一個封包就是送 m 個月台,每個月台內含 1-n 組即將到站列車資訊

封包內含資訊為以下所述

| Packet Name | | 訊息傳送至 PGIS | | |
|------------------------|--------------|---------------|---------------|------------------------------------------------------------------------------------------------|
| Variable | | Values | | |
| Name | Bytes | No. | Choice | Name/Mean |
| Packet Num. | 1 | | | 訊息傳送至 PGIS |
| sequence number | 2 | | | 65535 最大值之後回復為零開始 |
| Packet Length | 2 | | | 計算以下的總長度 |
| StationID | 4 | == | == | refer to platform ID |
| Platform ID 1 | 4 | | | [車站和月台 ID] [4Byte] [前 3Byte 表示 StationID,第 4Byte 表示上下行(1=Up Track, 2=Down Track)] [ASCII 字元表示] |
| Status | 4-15 | | | [月台開放狀態] [4-15 Bytes] [Open, Closed, Single Tracking(單軌運行雙向,通常有這表示其中一軌不正常)] [ASCII 字元表示] |
| Car Count | 1 | | | 顯示該站接近的列車數量 |
| Train 1 PVID | 3 | | | [列車永久號] [ASCII 字元表示] |
| Train 1 Arrival Time | 6 | | | [預計到站時間] [hh:mm:ss 000000~235959] [ASCII 字元表示] |
| Train 1 Departure Time | 6 | | | [預計離站時間][hh:mm:ss 000000~235959] [ASCII 字元表示] |
| Train 1 Run ID | 3 | | | [供未來使用,用來表示列車跑過幾趟計畫] [ASCII 字元表示,從 000 開始] |
| Train 1 Stopping | 1 | | | [表示列車是否將停在此月台] [Y 或 N] [ASCII 字元表示] |
| Train 1 Accurate | 1 | | | [表示估計到站和離站時間是否準確(由 ATS 的 Vehicle Regulation 預估)] [Y 或 N] [ASCII 字元表示] |
| Train 1 Loading | 1 | | | [列車是否提供載客服務] [Y 或 N] [ASCII 字元表示] |

| | | | | |
|----------------------------------|-----|--|--|----------------------------------------------------------------------------|
| Train 1 Direction | 1 | | | [表示列車行駛方向] [0=方向未決定, 1=right to left 上行, 2=left to right 下行] [ASCII 字元表示] |
| Train 1 is Last Train on Track 1 | 1 | | | [是否為 Track 1(上行)上的最後一班列車] [Y 或 N] [ASCII 字元表示] |
| Train 1 is Last Train on Track 2 | 1 | | | [是否為 Track 2(下行)上的最後一班列車] [Y 或 N] [ASCII 字元表示] |
| Train 1 Destination Platform | 3 | | | [表示列車預計開往的終點站] [ASCII 字元表示] [ASCII 字元表示] |
| Train 1 is on Terminus Station | 1 | | | [表示該 Station 是否為 Terminus Station] [Y 或 N] [ASCII 字元表示] |
| Train 2 PVID | 3 | | | [列車永久號] [ASCII 字元表示] 車輛數量為一台以上就會重複 |
| | ... | | | ... |
| Platform ID 2 CRC | 4 | | | 若有第二站的話則會跟第一站一樣重複[假如有的話會重複 Platform ID 2 + Train 1~n] 跟第一站一樣的重複這個流程 |
| | | | | |
| | 2 | | | |

1.2.3 CRC Code

CRC 演算法如下表示：

```
public class CRC_Code
{
    static readonly ushort[] crcTab = new ushort[] {
        0x0000, 0x1189, 0x2312, 0x329b, 0x4624, 0x57ad, 0x6536, 0x74bf,
        0x8c48, 0x9dc1, 0xaf5a, 0xbed3, 0xca6c, 0xdbe5, 0xe97e, 0xf8f7,
        0x1081, 0x0108, 0x3393, 0x221a, 0x56a5, 0x472c, 0x75b7, 0x643e,
        0x9cc9, 0x8d40, 0xbfdb, 0xae52, 0xdaed, 0xcb64, 0xf9ff, 0xe876,
        0x2102, 0x308b, 0x0210, 0x1399, 0x6726, 0x76af, 0x4434, 0x55bd,
        0xad4a, 0xbcc3, 0x8e58, 0x9fd1, 0xeb6e, 0xfae7, 0xc87c, 0xd9f5,
        0x3183, 0x200a, 0x1291, 0x0318, 0x77a7, 0x662e, 0x54b5, 0x453c,
        0xbdc b, 0xac42, 0x9ed9, 0x8f50, 0xfbef, 0xea66, 0xd8fd, 0xc974,
        0x4204, 0x538d, 0x6116, 0x709f, 0x0420, 0x15a9, 0x2732, 0x36bb,
        0xce4c, 0xdfc5, 0xed5e, 0xfcd7, 0x8868, 0x99e1, 0xab7a, 0xbaf3,
        0x5285, 0x430c, 0x7197, 0x601e, 0x14a1, 0x0528, 0x37b3, 0x263a,
        0xdec d, 0xcf44, 0xfddf, 0xec56, 0x98e9, 0x8960, 0xbbfb, 0xaa72,
        0x6306, 0x728f, 0x4014, 0x519d, 0x2522, 0x34ab, 0x0630, 0x17b9,
        0xef4e, 0xfec7, 0xcc5c, 0xdd5, 0xa96a, 0xb8e3, 0x8a78, 0x9bf1,
        0x7387, 0x620e, 0x5095, 0x411c, 0x35a3, 0x242a, 0x16b1, 0x0738,
        0xffcf, 0xee46, 0xdcdd, 0xcd54, 0xb9eb, 0xa862, 0x9af9, 0x8b70,
        0x8408, 0x9581, 0xa71a, 0xb693, 0xc22c, 0xd3a5, 0xe13e, 0xf0b7,
        0x0840, 0x19c9, 0x2b52, 0x3adb, 0x4e64, 0x5fed, 0x6d76, 0x7cff,
        0x9489, 0x8500, 0xb79b, 0xa612, 0xd2ad, 0xc324, 0xf1bf, 0xe036,
        0x18c1, 0x0948, 0x3bd3, 0x2a5a, 0x5ee5, 0x4f6c, 0x7df7, 0x6c7e,
        0xa50a, 0xb483, 0x8618, 0x9791, 0xe32e, 0xf2a7, 0xc03c, 0xd1b5,
        0x2942, 0x38cb, 0x0a50, 0x1bd9, 0x6f66, 0x7eef, 0x4c74, 0x5dfd,
        0xb58b, 0xa402, 0x9699, 0x8710, 0xf3af, 0xe226, 0xd0bd, 0xc134,
        0x39c3, 0x284a, 0x1ad1, 0x0b58, 0x7fe7, 0x6e6e, 0x5cf5, 0x4d7c,
        0xc60c, 0xd785, 0xe51e, 0xf497, 0x8028, 0x91a1, 0xa33a, 0xb2b3,
        0x4a44, 0x5bcd, 0x6956, 0x78df, 0x0c60, 0x1de9, 0x2f72, 0x3efb,
```

```

0xd68d, 0xc704, 0xf59f, 0xe416, 0x90a9, 0x8120, 0xb3bb, 0xa232,
0x5ac5, 0x4b4c, 0x79d7, 0x685e, 0x1ce1, 0x0d68, 0x3ff3, 0x2e7a,
0xe70e, 0xf687, 0xc41c, 0xd595, 0xa12a, 0xb0a3, 0x8238, 0x93b1,
0x6b46, 0x7acf, 0x4854, 0x59dd, 0x2d62, 0x3ceb, 0x0e70, 0x1ff9,
0xf78f, 0xe606, 0xd49d, 0xc514, 0xb1ab, 0xa022, 0x92b9, 0x8330,
0x7bc7, 0x6a4e, 0x58d5, 0x495c, 0x3de3, 0x2c6a, 0x1ef1, 0x0f78
};

```

//以整個array長度下去計算

```

public static ushort compute_crc(byte[] data)
{
    return compute_crc(data, 0, data.Length);
}

```

//從起始位置，往後包含幾byte下去計算，以CMFT為例，start為0，length為35

```

public static ushort compute_crc(byte[] data, int start, int length)
{
    ushort fcs = 0xFFFF;

    int end = start + length;

    for (int i = start; i < end; i++)
    {
        fcs = (ushort)((ushort)(fcs >> 8) ^ crcTab[(fcs ^ data[i]) & 0xFF]);
    }

    return (ushort)(~fcs);
}

```

```

}

```

