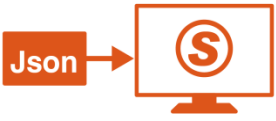
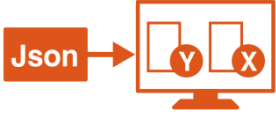

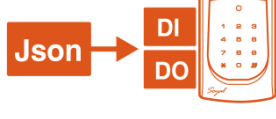





# **SOYAL**

## **SOYAL 701Server Json 格式萬用指令 2000 範例教學**

**用途:** SOYAL 支援第三方軟體平台透過 JSON 指令與 SOYAL 多功能控制器與工業版系列設備整合；第三方軟體可透過 SOYAL 軟體 701ServerSQL 發送 JSON 指令給 SOYAL 設備，由第三方軟體平台同時監控現場設備狀態，適用於訪客管理系統、考勤系統、BMS 系統整合…等。

C_CMD Value	功能	
1000	讀取主控軟體資料	
1001	讀取控制器類型與連線狀態	
1002	讀取遠端控制器 I/O 狀態	
1003	讀取遠端控制器 I/O 狀態	
1004	設定遠程控制器 I/O 狀態	
1021	設定訪客卡片 UID 和日期時間限制	
1022	刪除訪客卡片 UID 和日期時間限制	

3002	創建事件紀錄	
3003	設定控制器時間	
2000	HEX 格式的協議傳輸	

### 硬體準備:

支援 SOYAL 家用版 Home H 系列控制器，企業版 Enterprise E 控制多門控制器 AR-716-E 系列及 401/403-IO 系列，但是其中的兩個 **Json**

指令: 定訪客使用者資料 **Set Guest Tag UID and Date Time Limit**

與清除訪客使用者資料 **Clear Guest Tag UID and Date Time limit**

**僅支援**企業版 Enterprise E 控制器 跟多門控制器 AR-716-E16

(我們用企業版網絡型液晶顯示控制器 AR-837E-A 做範例教學)



### 軟體準備:

1. SOYAL 701Server/ 701Client 軟體版本 10V2 (File base 檔案模式 跟

SQL Database 模式都支援)



## 2. Json 格式轉換在線軟體

<https://jsonlint.com/>

格式驗證正確

The screenshot shows the JSONLint website interface. At the top, it says 'JSONLint - The JSON Validator'. Below this is a text area containing a valid JSON object. The JSON is a root object with a 'l\_user' property and a 'cmd\_array' property. The 'cmd\_array' is an array containing one object with various fields like 'c\_cmd', 'Area', 'Node', 'Addr', 'TagUID', 'Begin\_dt', 'End\_dt', 'Lift', 'DoorAccess', 'PIN', and 'Mode'. The 'Validate JSON' button is circled in green, and the 'Results' section shows 'valid JSON' in a green box.

```
1 {
2   "l_user": "login user",
3   "cmd_array": [{
4     "c_cmd": 1021,
5     "Area": 0,
6     "Node": 1,
7     "Addr": 1,
8     "TagUID": "0x00000000057B235",
9     "Begin_dt": "2021-5-11 00:00",
10    "End_dt": "2021-6-18 23:59",
11    "Lift": "0x3A5B67DE",
12    "DoorAccess": "0x00FF",
13    "PIN": 1111,
14    "Mode": 2
15  }]
16 }
```

Validate JSON Clear

Results

valid JSON

格式驗證錯誤

The screenshot shows the JSONLint website interface. The JSON object is identical to the one in the previous screenshot, but there is a missing closing brace for the 'Mode' property on line 14. The 'Validate JSON' button is circled in blue, and the 'Results' section shows an error message in a red box: 'Error: Parse error on line 14: ...1111, "Mode": 2, }}} Expecting 'STRING', got '''. The error message is highlighted with a blue box.

```
1 {
2   "l_user": "login user",
3   "cmd_array": [{
4     "c_cmd": 1021,
5     "Area": 0,
6     "Node": 1,
7     "Addr": 1,
8     "TagUID": "0x00000000057B235",
9     "Begin_dt": "2021-5-11 00:00",
10    "End_dt": "2021-6-18 23:59",
11    "Lift": "0x3A5B67DE",
12    "DoorAccess": "0x00FF",
13    "PIN": 1111,
14    "Mode": 2,
15  }]
16 }
```

Validate JSON Clear

Results

Error: Parse error on line 14:  
...1111, "Mode": 2, }}}  
-----^  
Expecting 'STRING', got ''

## 3. Protocol 通訊協議準備:

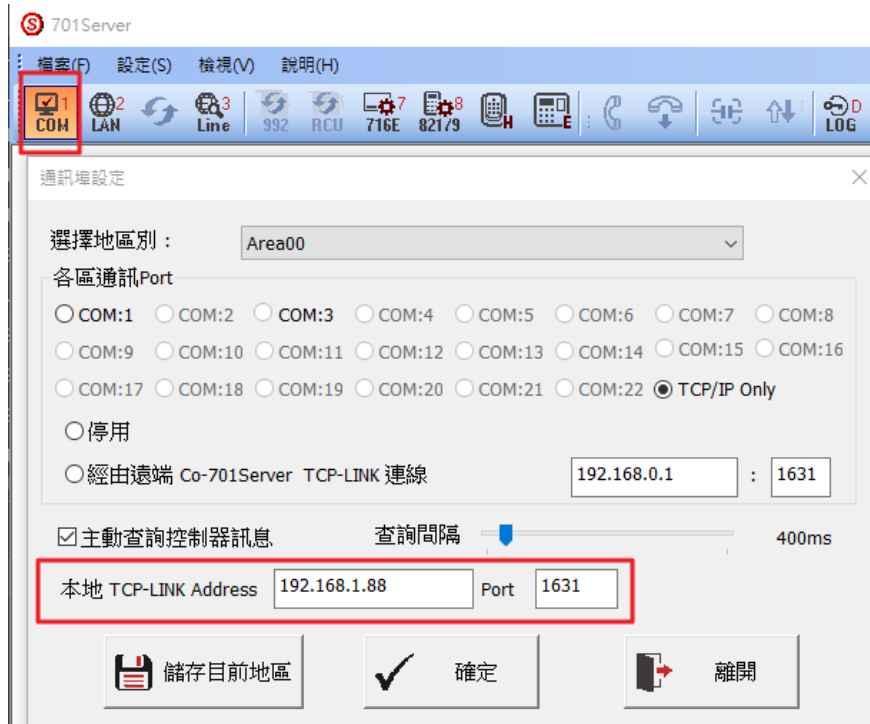
## 1. 701Server Json Command 1V04.pdf



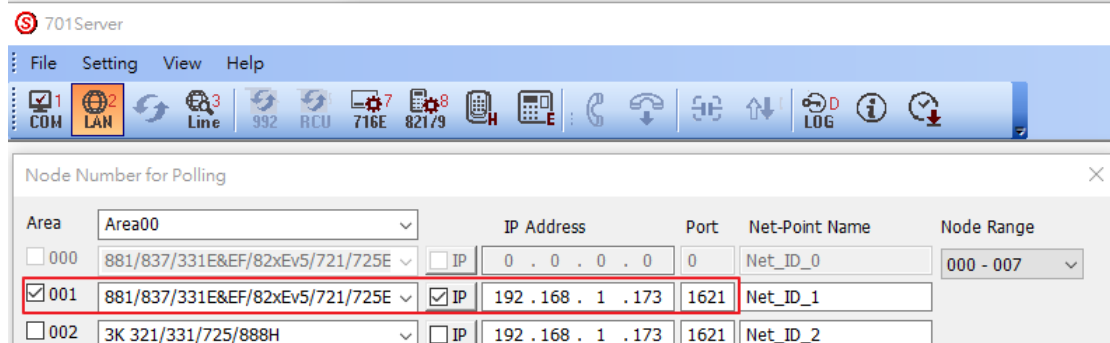
**範例：**用企業版網絡型液晶顯示控制器 AR-837E-A 做範例教學，透過  
Json 指令設定使用者資料

**1. 701Server 軟體設定，使 AR-837E-A 跟 701Server 保持通訊**

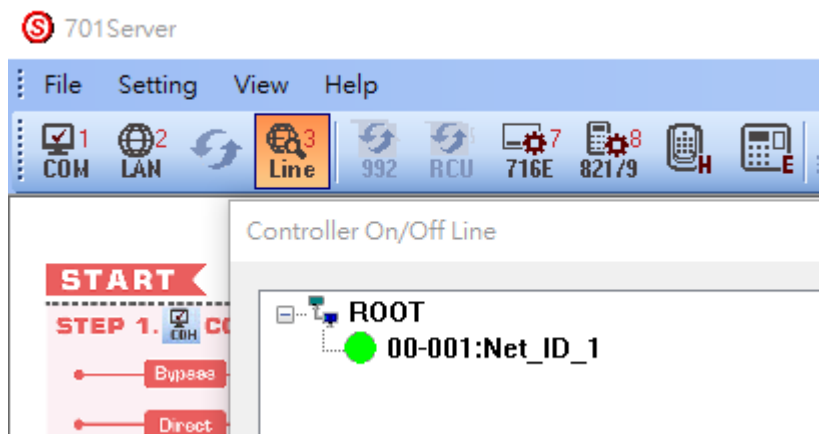
**1.1 選擇 TCP/IP Only ， 輸入本機電腦 IP Address 跟 Port Number**



1.2 選擇 881/837/331E&EF/82xEv5/721/725Ev2/727/327Hv5，輸入 IP Address, **Port 1621**



1.3 確認通訊正常



## 2. 調整控制器 AR-837E-A 顯示的卡片格式是 Hex



## 3. 以下我們用三個範例來示範 Json 萬用指令的功能

### ▶ 範例 1:

透過 Json 指令 “2000”和通訊協定指令 21H 控制門鎖繼電器開/關

### ▶ 範例 2:

透過 Json 指令 “2000”和通訊協定指令 8BH 設定訪客通行時間…等資訊

### ▶ 範例 3:

透過 Json 指令 “2000”和通訊協定 RTC 指令 23H 設定控制器時間

---

### 範例 1:

透過 Json 指令 “2000” 和通訊協定指令 21H 控制門鎖繼電器開/關



透過 Json 指令 “2000” 和通訊協定 RTC 指令 21H (7E 05 01 21 84 5B 01) 控制門鎖繼電器開/關

Jason Command:

```

{
  "l_user": "login user",
  "cmd_array": [
    {
      "c_cmd": 2000,
      "Area": 0,
      "Node": 1,
      "Hex": "0x2184"
    }
  ]
}

```

**Tx:**

```

{"l_user":"login user","cmd_array":[{"c_cmd":2000,"Area":0,"Node":1,"Hex":"0x2184"}]}

```

**Rx:**

```

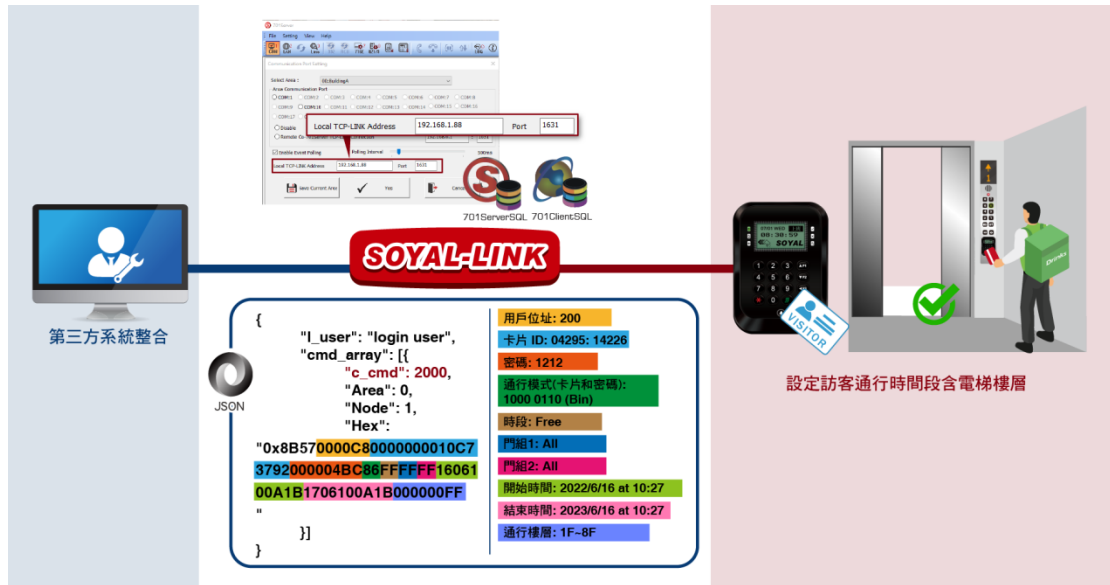
{"resp_array":[{"Area":0,"Hex":"0x7E0D000301440F90101000000262D","Node":1,"c_cmd":2000,"c_resp":3}]}

```

Time	From IP	From Port	To Address	To Port	Method	Error	ASCII
09:05:32.917	192.168...	1631	You	1067	TCP		{"resp_array":[{"Area":0,"Hex":"0x7E0D000301440F90101000000262D","Node":1,"c_cmd":2000,"c_resp":3}]}
09:05:32.916	You	1067	192.168.1.88	1631	TCP		{"l_user":"login user","cmd_array":[{"c_cmd":2000,"Area":0,"Node":1,"Hex":"0x2184"}]}

## 範例 2：

透過 Json 指令 “2000” 和通訊協定指令 8BH 設定訪客通行時間…等資訊



透過Json指令 “2000” 和通訊協定 RTC 指令 8BH (7E 26 01 8B 57 00 00 C8 00 00 00 00 10 C7 37 92 00 00 04 BC 86 FF FF FF 16 06 10 0A 1B 17 06 10 0A 1B 00 00 00 FF A7 D7) ，設定訪客通行時間…等資訊，以下範例設定內容為:

用戶位址: 200 / 卡片 ID: 04295:14226 / 密碼: 1212 / 通行模式: 卡片和密碼 / 開始時間: 2022/6/16 10:27 / 結束時間: 2023/6/16 10:27 / 通行樓層: 1F~8F

Jason Command:

```
{
  "_user": "login user",
  "cmd_array": [
    {
      "c_cmd": 2000,
      "Area": 0,
      "Node": 1,
      "Hex": "
```

0x8B570000C8000000010C73792000004BC86FFFFFF1606100A1B1706100A1

B000000FF

```
"  
  }  
}
```

Tx:

```
{"_user": "login  
user", "cmd_array": [{"c_cmd": 2000, "Area": 0, "Node": 1, "Hex": "0x8B570000C80000  
000010C73792000004BC86FFFFFFF1606100A1B1706100A1B000000FF "}]}
```

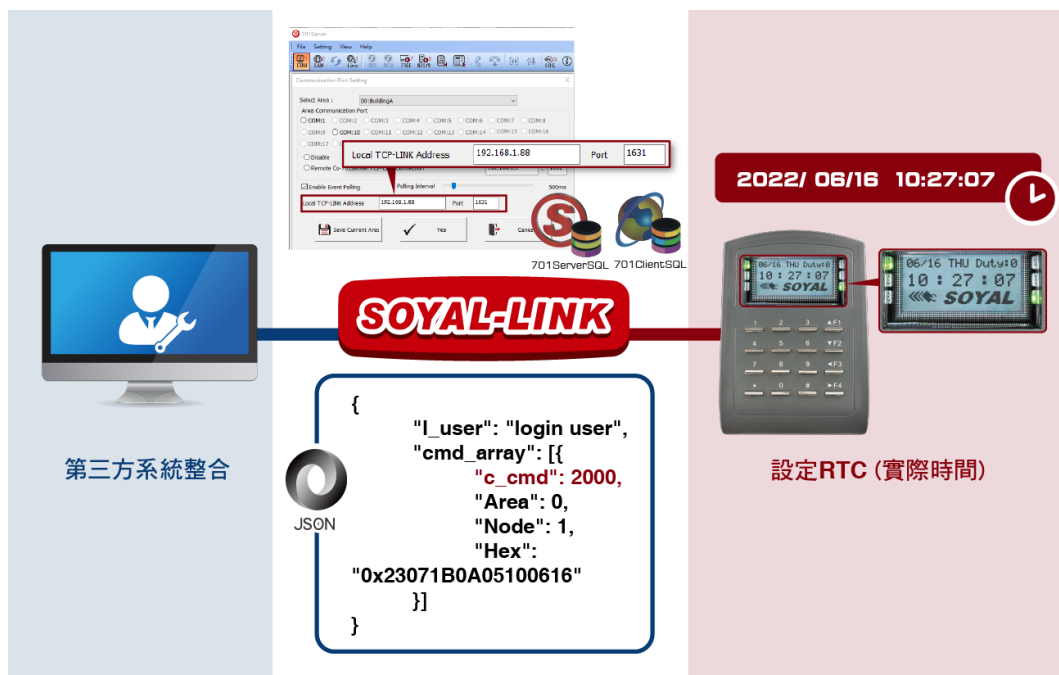
Rx:

```
{"resp_array": [{"Area": 0, "Hex": "0x7E0F000401C2440F90101000000000E3AD", "No  
de": 1, "c_cmd": 2000, "c_resp": 3}]}
```

Time	From IP	From Port	To Address	To Port	Method	Event
09:44:44.781	192.168.1.88	1631	You	1070	TCP	[Resp_array [{"Area": 0, "Hex": "0x7E0F000401C2440F90101000000000E3AD", "Node": 1, "c_cmd": 2000, "c_resp": 3}]]
09:44:44.776	You	1070	192.168.1.88	1631	TCP	[User {"_user": "login user", "cmd_array": [{"c_cmd": 2000, "Area": 0, "Node": 1, "Hex": "0x8B570000C80000000010C73792000004BC86FFFFFFF1606100A1B1706100A1B000000FF "}]}

### 範例 3 :

透過 Json 指令 “2000” 和通訊協定 RTC 指令 23H 設定控制器時間



透過 Json 指令 “2000” 和通訊協定 RTC 指令 23H (7E 0B 01 23 07 1B 0A 05 10 06 16 CE 4F)，設定控制器時間

Json Command:

```
{
  "_user": "login user",
  "cmd_array": [{
    "c_cmd": 2000,
    "Area": 0,
    "Node": 1,
    "Hex": "0x23071B0A05100616"
  }]
}
```

**Tx:**

```
{"_user":"login user","cmd_array":[{"c_cmd":2000,"Area":0,"Node":1,"Hex":"0x23071B0A05100616"}]}
```

**Rx:**

```
{"resp_array":[{"Area":0,"Hex":"0x7E07000401DF341129","Node":1,"c_cmd":2000,"c_resp":3}]}
```

( 07 1B 0A 05 10 06 16 in Hex format = 07 27 10 05 16 06 22 in Dec format means Sec.Min.Hr.Week.Day.Mon.Year 2022 Jun 16 at 10:27:07)

Time	From IP	From Port	To Address	To Port	Method	Error	ASCII
17:08:42.925	192.168...	1631	You	28200	TCP		{"resp_array":[{"Area":0,"Hex":"0x7E07000401DF341129","Node":1,"c_cmd":2000,"c_resp":3}]}
17:08:42.921	You	28200	192.168.1.88	1631	TCP		{"_user":"login user","cmd_array":[{"c_cmd":2000,"Area":0,"Node":1,"Hex":"0x23071B0A05100616"}]}

**RX :**

```
{"resp_array":[{"Area":0,"Hex":"0xFF005AA50021000B0117260E051006161100640000100092370100C71000000000000011ED25D1","Node":1,"c_cmd":2000,"c_resp":3}]}
```

