

EmETXe-i89U0-F75111 DIO SDK

版本：DIO-EmETXe-i89U0-F75111-V2.0.1

1. 安裝 DIO-EmETXe-i89U0-F75111-Setup

執行 DIO-EmETXe-i89U0-F75111-Setup.exe

2. DIO 版本

```
string DIO_Version75111();
```

範例：

```
string sVersion = DIO_Version75111();
```

3. DIO 開啟

```
bool DIO_Open75111()
```

範例：

```
bool bDIO = DIO_Open75111 ();
```

4. DIO 模式

設定 DIO 輸入輸出模式 (請參考電路對應 75111 設定)

```
void DIO_Mode75111(int iAdd,int iMode)
```

範例：

```
DIO_Mode75111(0x10, 0x20);           // GPIO15 為輸出, 其他 GPIO1X 為輸入  
DIO_Mode75111(0x20, 0xE0);         // GPIO25、GPIO26 及 GPIO27 為輸出, 其他 GPIO2X 為輸入  
DIO_Mode75111(0x40, 0x00);         // GPIO31~ GPIO31 為輸入
```

5. 設定 DO 輸出電位

```
void DIO_Data75111(int iAdd,int iData,int iMode)
```

範例：

```
DIO_Data75111(0x11, 0xDF, 0);       // 設 DO1 為低電平  
DIO_Data75111(0x11, 0x20, 1);      // 設 DO1 為高電平  
DIO_Data75111(0x21, 0x7F, 0);      // 設 DO2 為低電平  
DIO_Data75111(0x21, 0x80, 1);      // 設 DO2 為高電平  
DIO_Data75111(0x21, 0xBF, 0);      // 設 DO3 為低電平  
DIO_Data75111(0x21, 0x40, 1);      // 設 DO3 為高電平  
DIO_Data75111(0x21, 0xDF, 0);      // 設 DO4 為低電平  
DIO_Data75111(0x21, 0x20, 1);      // 設 DO4 為高電平
```

6. 取得 DIO 電位

```
int DIO_Status75111(int iAdd)
```

範例：

```
int iStatus12 = DIO_Status75111(0x12); // 取得 GPIO1X 電位
int iStatus22 = DIO_Status75111(0x22); // 取得 GPIO2X 電位
int iStatus42 = DIO_Status75111(0x42); // 取得 GPIO3X 電位

int iDI1 = (iStatus12 >> 6) & 0x1; // 取得 DI1 = GPIO16 電位
int iDI2 = (iStatus12 >> 7) & 0x1; // 取得 DI2 = GPIO17 電位
int iDI3 = (iStatus22 >> 0) & 0x1; // 取得 DI3 = GPIO20 電位
int iDI4 = (iStatus22 >> 1) & 0x1; // 取得 DI4 = GPIO21 電位
int iDI5 = (iStatus22 >> 2) & 0x1; // 取得 DI5 = GPIO22 電位
int iDI6 = (iStatus12 >> 4) & 0x1; // 取得 DI6 = GPIO14 電位
int iDI7 = (iStatus12 >> 0) & 0x1; // 取得 DI7 = GPIO10 電位
int iDI8 = (iStatus12 >> 1) & 0x1; // 取得 DI8 = GPIO11 電位
int iDI9 = (iStatus12 >> 2) & 0x1; // 取得 DI9 = GPIO12 電位
int iDI10 = (iStatus42 >> 1) & 0x1; // 取得 DI10 = GPIO31 電位
int iDI11 = (iStatus42 >> 2) & 0x1; // 取得 DI11 = GPIO32 電位
int iDI12 = (iStatus42 >> 3) & 0x1; // 取得 DI12 = GPIO33 電位

int iDO1 = (iStatus12 >> 5) & 0x1; // 取得 DO1 = GPIO15 電位
int iDO2 = (iStatus22 >> 7) & 0x1; // 取得 DO2 = GPIO27 電位
int iDO3 = (iStatus22 >> 6) & 0x1; // 取得 DO3 = GPIO26 電位
int iDO4 = (iStatus22 >> 5) & 0x1; // 取得 DO4 = GPIO25 電位
```