



## GetBoardVersion

BOOL GetBoardVersion(int nModel, int nBoard, int \*version)

**Parameters:**

nModel :

nBoard :

DIP

\*version :

**Return Value:**

“TRUE”,

“FALSE”

## PLL API Functions

### **Overview**

BOOL PLL\_SetClock(int nModel, int nBoard, int dwVal)

BOOL PLL\_GetClock(int nModel, int nBoard, int\* dwVal)

### **PLL\_SetClock**

BOOL PLL\_SetClock(int nModel, int nBoard, int dwVal)

AD

#### **Parameters:**

nModel :

nBoard : DIP

dwVal : 1,040~67,000,000hz

#### **Return Value:**

“TRUE”,

“FALSE”

### **PLL\_GetClock**

BOOL PLL\_GetClock(int nModel, int nBoard, int\* dwVal)

#### **Parameters:**

nModel :

nBoard : DIP

\*dwVal :

#### **Return Value:**

“TRUE”,

“FALSE”

## DAC API Functions

### Overview

**BOOL** DAC\_WaveGen(int nModel, int nBoard, int nChannel,  
int nMode, float fFreq, float peak, float offset,  
int \*dwBuf)

**BOOL** DAC\_SetFrequency(int nModel, int nBoard, int nChannel, float fFreq)

### DAC\_WaveGen

**BOOL** DAC\_WaveGen(int nModel, int nBoard, int nChannel,  
int nMode, float fFreq, float peak, float offset,  
int \*dwBuf)

DAC

#### Parameters:

nModel :

nBoard : DIP

nChannel : 가 "0"

nMode :

0	
1	
2	
3	
4	DC
5	

fFreq : .  $0 < fFreq \leq 1,000$ .

peak : .  $0 < peak \leq 10$ .

offset : DC

\*dwVal : . 1,000

#### Return Value:

"TRUE", "FALSE"

## DAC\_SetFrequency

BOOL DAC\_SetFrequency(int nModel, int nBoard, int nChannel, float fFreq)

DAC

**Parameters:**

nModel :

nBoard : DIP

nChannel : 가 "0"

fFreq :

**Return Value:**

"TRUE", "FALSE"

## DAC\_GetCycle

BOOL DAC\_GetCycle(int nModel, int nBoard, int nChannel, int \*nCycle)

DAC 가

**Parameters:**

nModel :

nBoard : DIP

nChannel : 가 "0"

\*nCycle : 가 16,777,215

**Return Value:**

"TRUE", "FALSE"

## DAC\_ClearCycle

BOOL DAC\_ClearCycle(int nModel, int nBoard, int nChannel)

DAC

**Parameters:**

nModel :

nBoard : DIP

nChannel : 가 "0"

**Return Value:**

"TRUE", "FALSE"



## ADC\_Reset

BOOL ADC\_Reset(int nModel, int nBoard, int nChannel)

AD

**Parameters:**

nModel :

nBoard : DIP

nChannel : 가 "0"

**Return Value:**

"TRUE", "FALSE"

## ADC\_ClockSelect

BOOL ADC\_ClockSelect(int nModel, int nBoard, int nChannel, int nSelect)

AD

**Parameters:**

nModel :

nBoard : DIP

nChannel : 가 "0"

nSelect : "0" 40Mhz OSC, "1" programmable clock

**Return Value:**

"TRUE", "FALSE"

## ADC\_SetSampleRate

BOOL ADC\_SetSampleRate(int nModel, int nBoard, int nChannel, int nSampleRate)

AD

**Parameters:**

nModel :

nBoard : DIP

nChannel : 가 "0"

nSampleRate : .  $5 \leq nSampleRate \leq 1,000$ .

**Return Value:**

"TRUE", "FALSE"





## AMP\_SetFeedback

BOOL AMP\_SetFeedback(int nModel, int nBoard, int nChannel, int nSelect)

(feedback) .(AIN0, AIN1)

### Parameters:

nModel : .  
nBoard : . DIP .  
nChannel : 가 "0"  
nSelect : "0" AIN0, "1" AIN1.

### Return Value:

"TRUE", "FALSE"





## ENC\_GetRpmCount

BOOL ENC\_GetRpmCount(int nModel, int nBoard, int nChannel, int \*nCnt)

가

### Parameters:

nModel :

nBoard : DIP

nChannel : 가 "0"

\*nCnt : 40Mhz 27- 10 z- 가

### Return Value:

"TRUE", "FALSE"

## ENC\_SetRpmClearCount

BOOL ENC\_GetRpmCount(int nModel, int nBoard, int nChannel, int \*nCnt)

rpm z-

### Parameters:

nModel :

nBoard : DIP

nChannel : 가 "0"

\*nCnt : 40Mhz 32- 10 (0x17D78400) ,

### Return Value:

"TRUE", "FALSE"

## ENC\_GetDistanceCount

BOOL ENC\_GetDistanceCount(int nModel, int nBoard, int nChannel, int \*nDir, int \*nCnt)

Quadrature( ) /

**Parameters:**

nModel :

nBoard : DIP

nChannel : 가 "0"

\*nDir : a/b . '0' , '1'

\*nCnt : a/b / 27- 0xFFFFFFFF ,

가 0x0 가 ( )

0x0 가 0xFFFFFFFF

**Return Value:**

"TRUE", "FALSE"

## PWM API Functions

### **Overview**

**BOOL PWM\_Enable(int nModel, int nBoard, int nChannel, int nDirection, BOOL bEnable)**  
**BOOL PWM\_CounterClear(int nModel, int nBoard, int nChannel, int nDirection, BOOL bClear)**  
**BOOL PWM\_SetSquare(int nModel, int nBoard, int nChannel, int nDirection, int nTime)**  
**BOOL PWM\_SetPattern(int nModel, int nBoard, int nChannel, int nDirection, int nActTime, int nPeriod)**  
**BOOL PWM\_SetUser(int nModel, int nBoard, int nChannel, int nDirection, int nSignal)**  
**BOOL PWM\_SetShotCount(int nModel, int nBoard, int nChannel, int nDirection, int nCount)**  
**BOOL PWM\_GetShotCount(int nModel, int nBoard, int nChannel, int nDirection, int \*nCount)**

### **PWM\_Enable**

BOOL PWM\_Enable(int nModel, int nBoard, int nChannel, int nDirection, BOOL bEnable);

PWM /

#### **Parameters:**

nModel :  
nBoard : DIP  
nChannel : 가 "0"  
nDirection : "0" CW( ), "1" CCW( ).  
bEnable : "TRUE", "FALSE".

#### **Return Value:**

"TRUE", "FALSE"

### **PWM\_CounterClear**

BOOL PWM\_CounterClear(int nModel, int nBoard, int nChannel, int nDirection, BOOL bClear);

PWM

#### **Parameters:**

nModel :  
nBoard : DIP  
nChannel : 가 "0"  
nDirection : "0" CW( ), "1" CCW( ).  
bClear : "1", "0"

#### **Return Value:**

"TRUE", "FALSE"

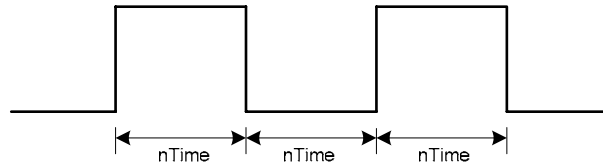
## PWM\_SetSquare

BOOL PWM\_SetSquare(int nModel, int nBoard, int nChannel, int nDirection, int nTime);

PWM

### Parameters:

nModel :  
nBoard : DIP  
nChannel : 가 "0"  
nDirection : "0" CW( ), "1" CCW( ).  
nTime: Set, Reset 40Mhz  
= 25nsec \* nTime[nsec]



### Return Value:

"TRUE", "FALSE"





## PWM\_SetShotCount

BOOL PWM\_SetShotCount(int nModel, int nBoard, int nChannel, int nDirection, int nCount);

PWM

**Parameters:**

nModel :

nBoard :

DIP

nChannel :

가

“0”

nDirection :

“0” CW(            ), “1” CCW(            ).

nCount : 27-

‘0’(LO)

**Return Value:**

“TRUE”,

“FALSE”

## PWM\_GetShotCount

BOOL PWM\_GetShotCount(int nModel, int nBoard, int nChannel, int nDirection, int \*nCount);

PWM

**Parameters:**

nModel :

nBoard :

DIP

nChannel :

가

“0”

nDirection :

“0” CW(            ), “1” CCW(            ).

nCount : 27-

**Return Value:**

“TRUE”,

“FALSE”

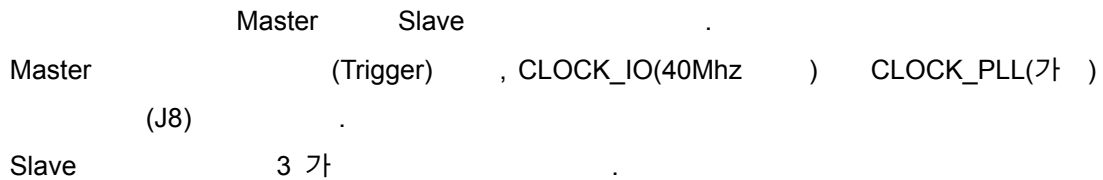
## **SYNC API Functions**

### **Overview**

- BOOL SYNC\_SetMode(int nModel, int nBoard, int mode)**
- BOOL SYNC\_SetRun(int nModel, int nBoard, int run)**
- BOOL SYNC\_GetParam(int nModel, int nBoard, int \*mode, int \*action)**
- BOOL SYNC\_SetDelayTime(int nModel, int nBoard, int delay)**
- BOOL SYNC\_GetDelayTime(int nModel, int nBoard, int \*delay)**

### **SYNC\_SetMode**

**BOOL SYNC\_SetMode(int nModel, int nBoard, int mode);**



#### **Parameters:**

- nModel :
- nBoard : DIP
- mode : '1'- Master, '0'-Slave.

#### **Return Value:**

"TRUE", "FALSE"

### **SYNC\_SetRun**

**BOOL SYNC\_SetRun(int nModel, int nBoard, int run);**

Master 가

#### **Parameters:**

- nModel :
- nBoard : DIP
- run : '1'- H('1')I, '0'- LO('0').

#### **Return Value:**

"TRUE", "FALSE"

## SYNC\_GetParam

```
BOOL SYNC_GetParam(int nModel, int nBoard, int *mode, int *action);
```

**Parameters:**

nModel :  
 nBoard : DIP  
 \*mode : '1'- Master, '0'-Slave.  
 \*action : '1'- , '0'-

**Return Value:**

"TRUE", "FALSE"

## SYNC\_SetDelayTime

```
BOOL SYNC_SetDelayTime(int nModel, int nBoard, int delay);
```

가  
 Master SYNC\_SetRun( , Slave (Trigger)  
 (25nsec\*delay) PID

**Parameters:**

nModel :  
 nBoard : DIP  
 delay: 0x00000000~0x7FFFFFFF.

**Return Value:**

"TRUE", "FALSE"

## SYNC\_GetDelayTime

```
BOOL SYNC_GetDelayTime(int nModel, int nBoard, int *delay);
```

**Parameters:**

nModel :  
 nBoard : DIP  
 \*delay: 0x00000000~0x7FFFFFFF.

**Return Value:**

"TRUE", "FALSE"

## References

**DAQ system Co., LTD.**

#913, Woolim lions valley 1'st  
Sanddaewon-dong, Jungwon-gu,  
Sungnam-si, Kyunggi-do, Korea, 462-806

<http://www.daqsystem.com>

Phone) +82 31 737 2018

Fax)+82 32 232 0501