MICROCHIP

Regional Training Centers

Core Independent Peripherals (CIPs)

帶來更多強化的功能

軟體與硬體需求

- **MPLAB X IDE v3.05 (**含**)** 或更新版本
- **XC8** 編譯器 **v1.35 (**本版本如需 **PIC18F** 週邊函數庫的支援需另外下載安裝**)** 或使用 **v1.34 (**內含 **PIC18F** 函數庫**)**
- **MPLAB MCC v2.25.2 (**請用此版本**)**
- 酷奇板**(Curiosity) + PIC16F1619**
- **CIP V2B** 子卡

程度需求

本實做課程不屬於基礎課程，為達到學習的目標建議參加此實做的學員需具備底下基本技能:

- 熟悉 XC8 編譯器及一定的 C 語言撰寫能力
- 熟悉 MPLAB X IDE 的使用及各項除錯功能用法
- 熟悉 MCC 基本周邊的設定及其周邊函數的使用
- 熟悉使用 PICkit3 或 ICD3 除錯工具
- 需具備電子電路硬體電路的設計知識

課程學習目標

- 了解 **CIPs** 的功能
- 在 **MPLAB X IDE** 下配置 **CIPs**
- 真實範例來欣賞 **CIPs** 的強大功能
- 讓 **CIPs** 的靈感充分展現在您的設計

Agenda

- 新硬體架構
- 什麼是 **CIP?**
- **CLC (Configurable Logic Cell)**
- **SMT (Signal Measurement Timer)**
- **TMR2,4,6 w/HLT**
- **PWM**
- **ZCD (Zero Cross Detector)**
- **AT (Angular Timer)**

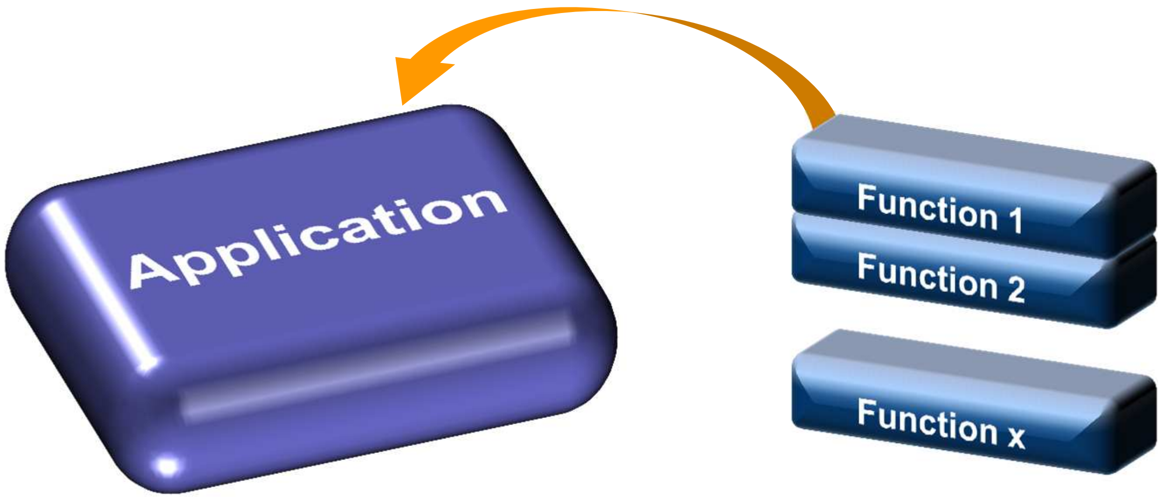
實做練習

- Lab 0 - 低電壓燒錄模式設定
- Lab 1 - 基本的 LED 閃爍
- Lab 2 - RGB 三色 LED 使用 CLC 的串列控制
- Lab 3 - SMT Duty Cycle 或週期的量測
- Lab 4 - ZCD 電壓零點偵測
- Lab 5 - HLT/Timer2 應用
- Lab 6 - ZCD 及 HLT/TMR2 觸發 TRIAC
- Lab 7 - Angular Timer Compare Mode
- Lab 8 - ZCD 相位移動觸發 TRIAC

Application Building Blocks
Functions & Peripherals

Application Building Blocks

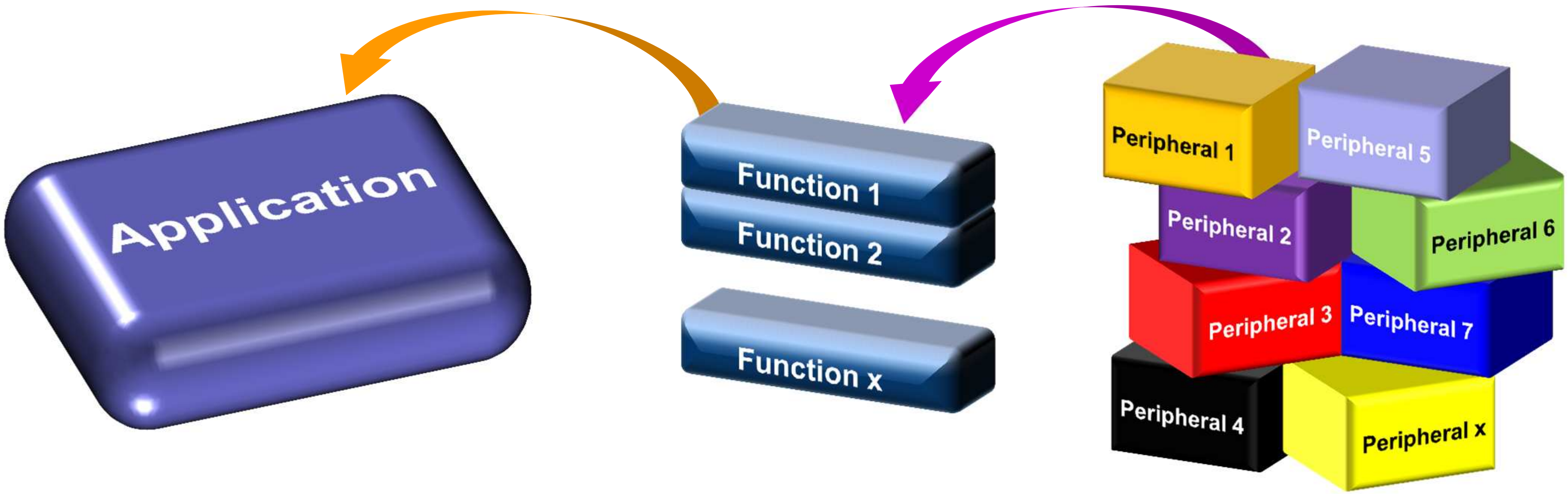
- 應用是由各種不同的功能方塊所集合成的
  - Application = Function 1 + Function 2 + Function x …



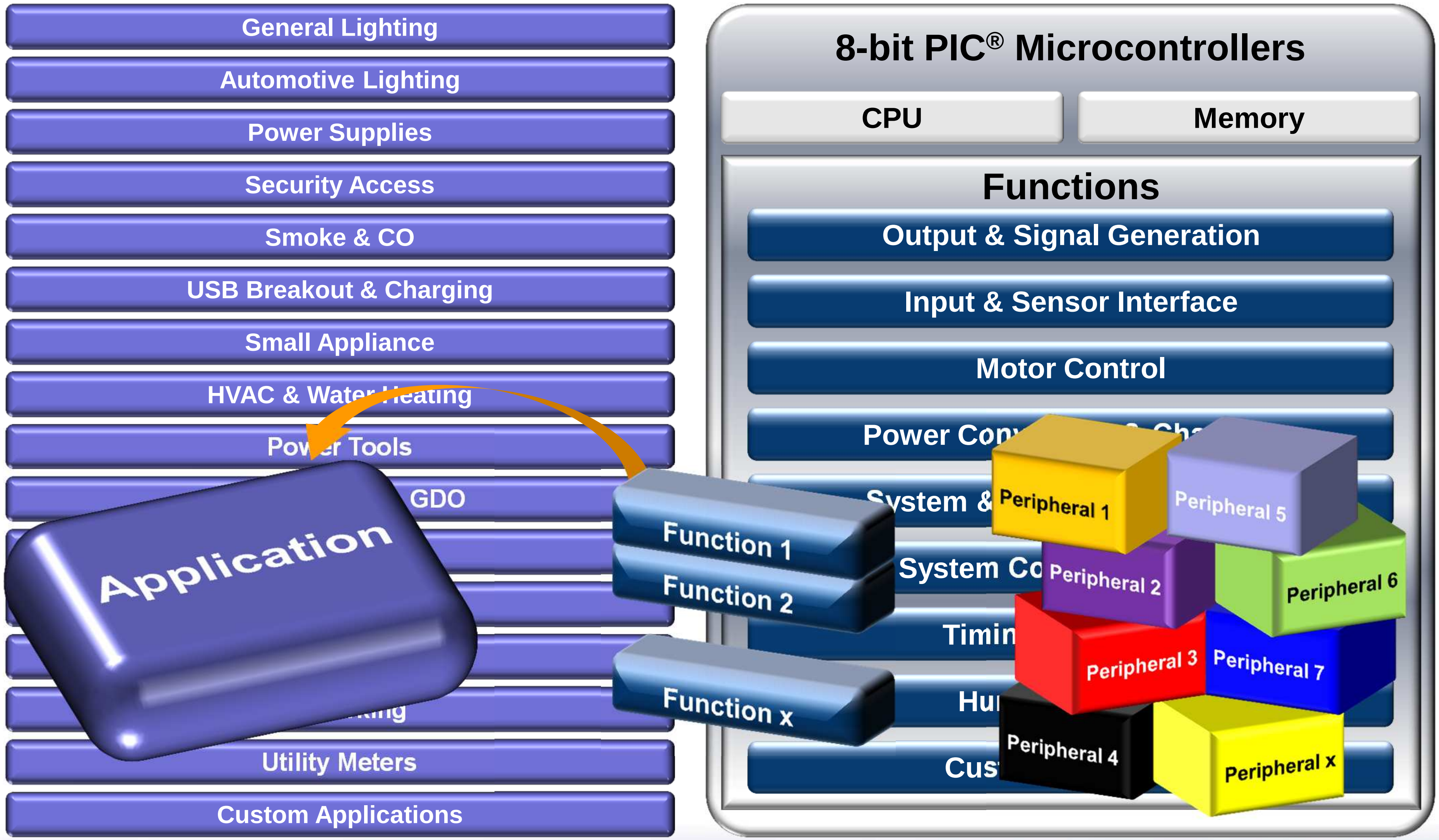
Application Building Blocks
Functions & Peripherals

Application Building Blocks

- 功能方塊也可能是更多週邊的集合體
  - Function x = Peripheral 1 + Peripheral 2 + Peripheral x …



Application Building Blocks
Functions & Peripherals



Microchip Regional Training Centers

彈性的組合
功能需求 & 週邊模組的契合

Intelligent Analog
Sensor Interfacing & Signal Conditioning

Waveform Control
PWM Drive & Waveform Generation

Timing & Measurements
Signal Measurement with Timing & Counter Control

Logic & Math
Customizable Logic & Math Functions

Safety & Monitoring
Hardware Monitoring & Fault Detection

Communications
Wired, Wireless & Encryption

User Interface
Capacitive Touch Sensing & LCD Control

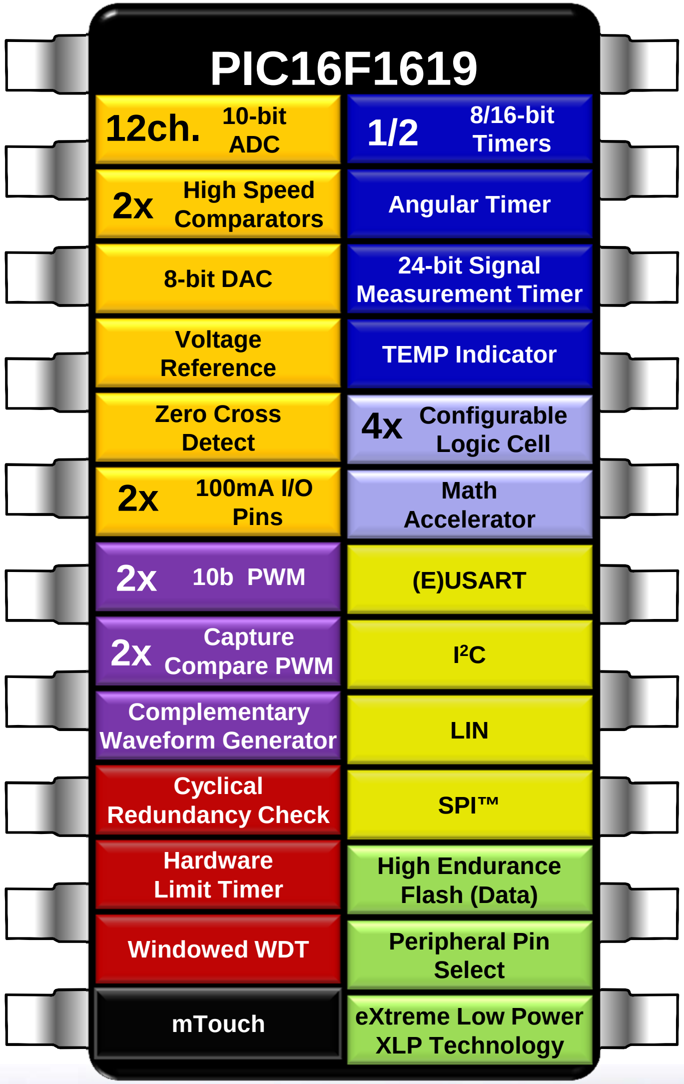
Low Power & System Flexibility
XLP Low Power Technology, Peripheral & Interconnects

8-bit PIC Microcontrollers

CPU		Memory	
ADC	(Enhanced) Capture Compare PWM	High Endurance Flash (Data)	Configurable Logic Cell
ADC^2	Complementary Output Generator	IDLE & DOZE	Hardware Multiply
Comparators	Complementary Waveform Generator	Peripheral Module Disable	Math Accelerator
DAC	Data Signal Modulator	Peripheral Pin Select	Crystal Free USB
High Speed Comparators	Numerically Controlled Oscillator	eXtreme Low Power XLP Technology	CAN
Operational Amplifiers	Programmable Switch Mode Controller	Angular Timer	(E)USART
Ramp Generator	10-bit PWM	Charge Time Measurement	ETHERNET MAC
Slope Compensation	16-bit PWM	RTCC	I^2C
Voltage Reference	Cyclical Redundancy Check	Signal Measurement Timer	LIN
Zero Cross Detect	Hardware Limit Timer	TEMP Indicator	SPI™
High Current I/O	Windowed WDT	8/16/20/24-bit Timers	Keeloq® Sub-GHz RF
LCD	mTouch		

PIC16(L)F1619 Overview

- **Small form-factor 20-pin packages:**
  - PDIP, SOIC, SSOP, QFN, UQFN
- **Memory:**
  - 14KB Flash
  - 1KB RAM
  - 128B HEF with 100K Erase/Write Cycles
- **Operating Characteristics:**
  - Configurable 32 kHz to 32 MHz operating speed
  - 1.8V (LF) to 5.5V (F) voltage range
  - -40 to 125ºC temp range
  - 50 nA SLEEP Current
  - 35 uA/MHz active current

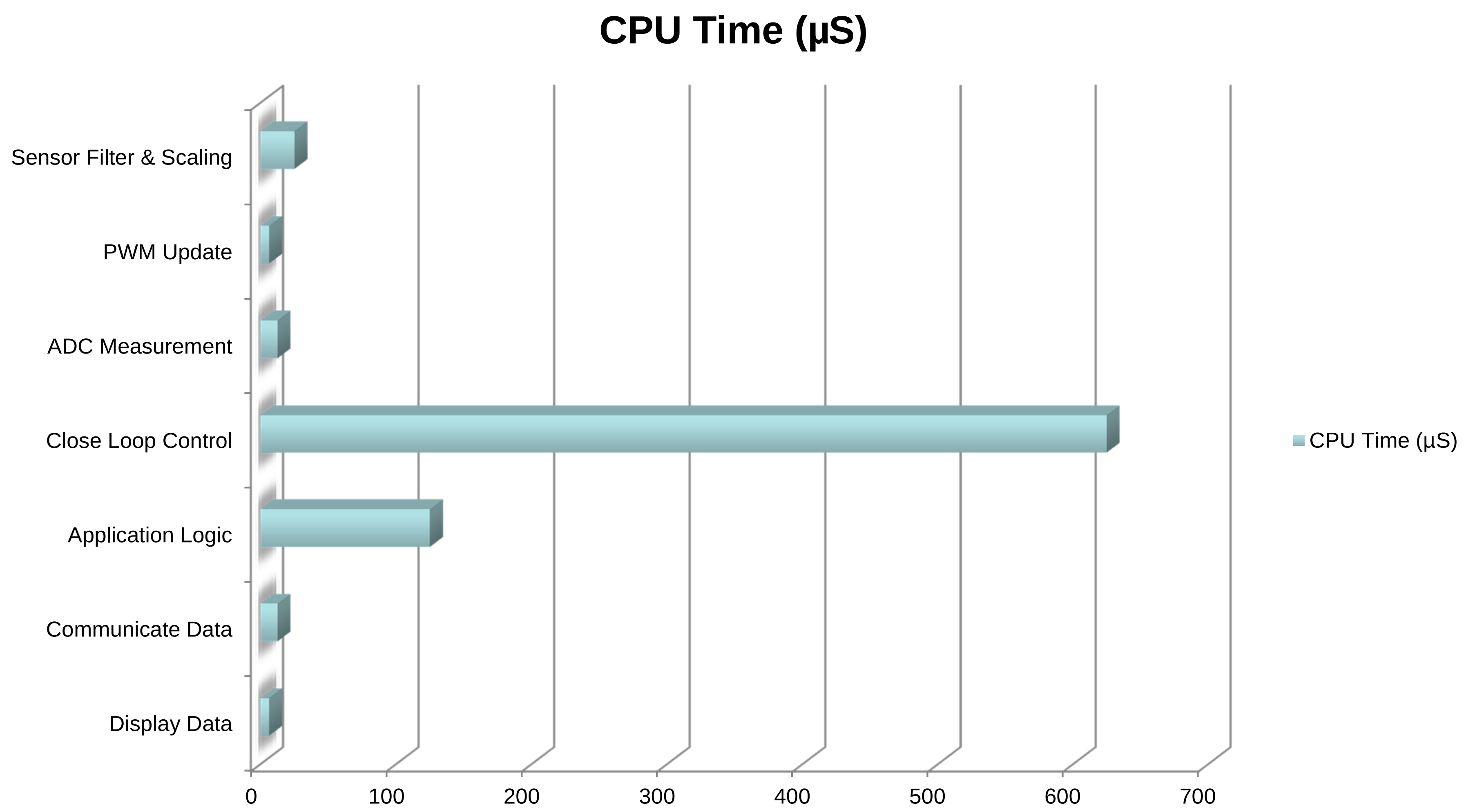


Regional Training Centers

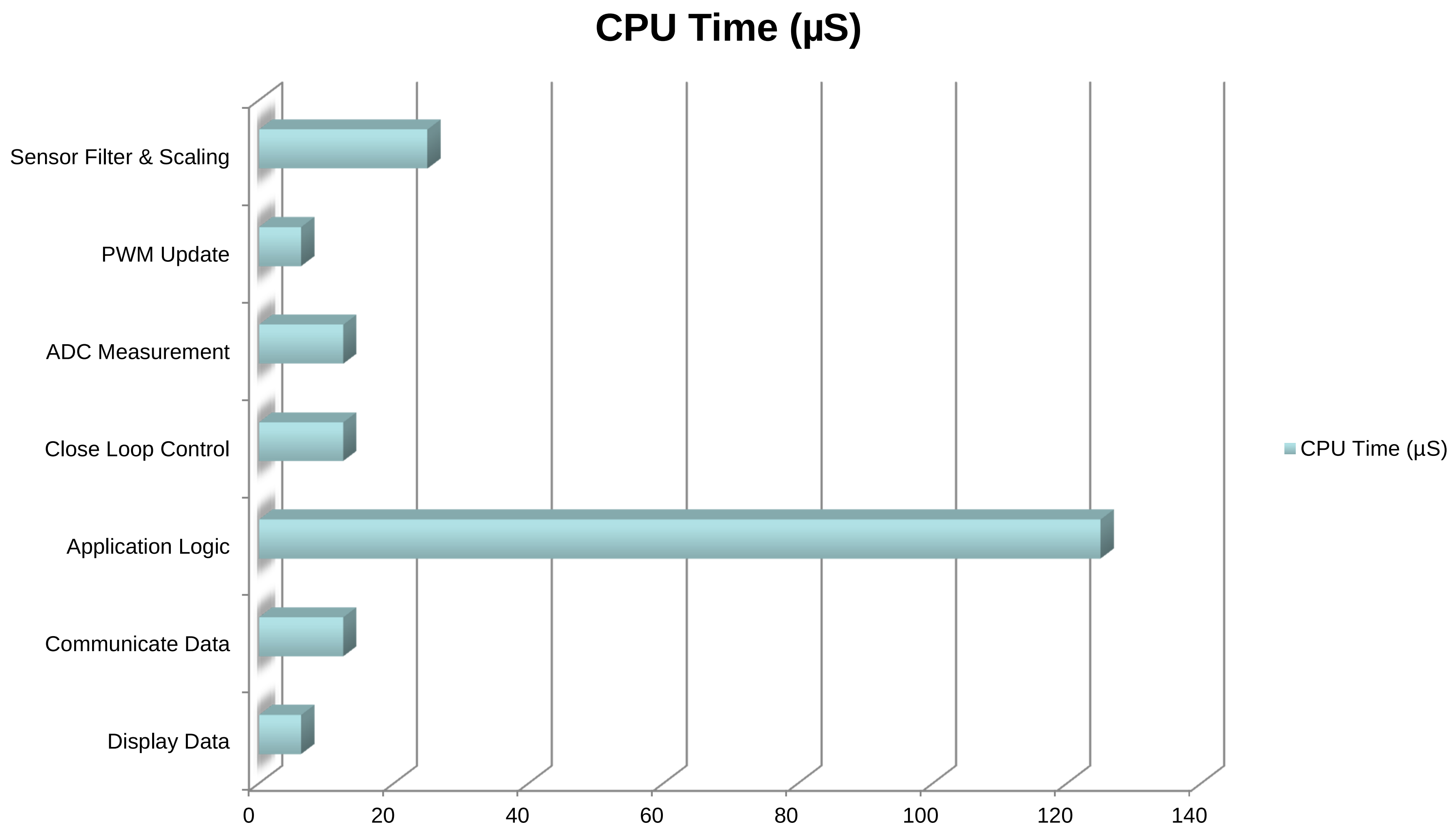
什麼是 CIP

- 獨立於核心外的週邊
  - CIP 週邊規劃後即可自行運做， 無需 CPU 常態性的藉入
  - 週邊裝置的集合，實現一個完整的系統的運作

一般 MCU 所需的效能分佈



使用 CIP 的效能需求



CPU Time (µS)
Sensor Filter & Scaling
PWM Update
ADC Measurement
Close Loop Control
Application Logic
Communicate Data
Display Data
0
20
40
60
80
100
120
140
CPU Time (µS)

使用 CIP 後優點概述

- 假設使用 **CIP** 前的應用，主程式迴圈為 **1mS**
  - It could sleep for 800usecs in every loop and save power.
  - It could slow down to ¼ the clock speed and save power or EMI.
  - You could add more secret sauce and be more competitive.

Lab 0
低電壓燒錄模式設定

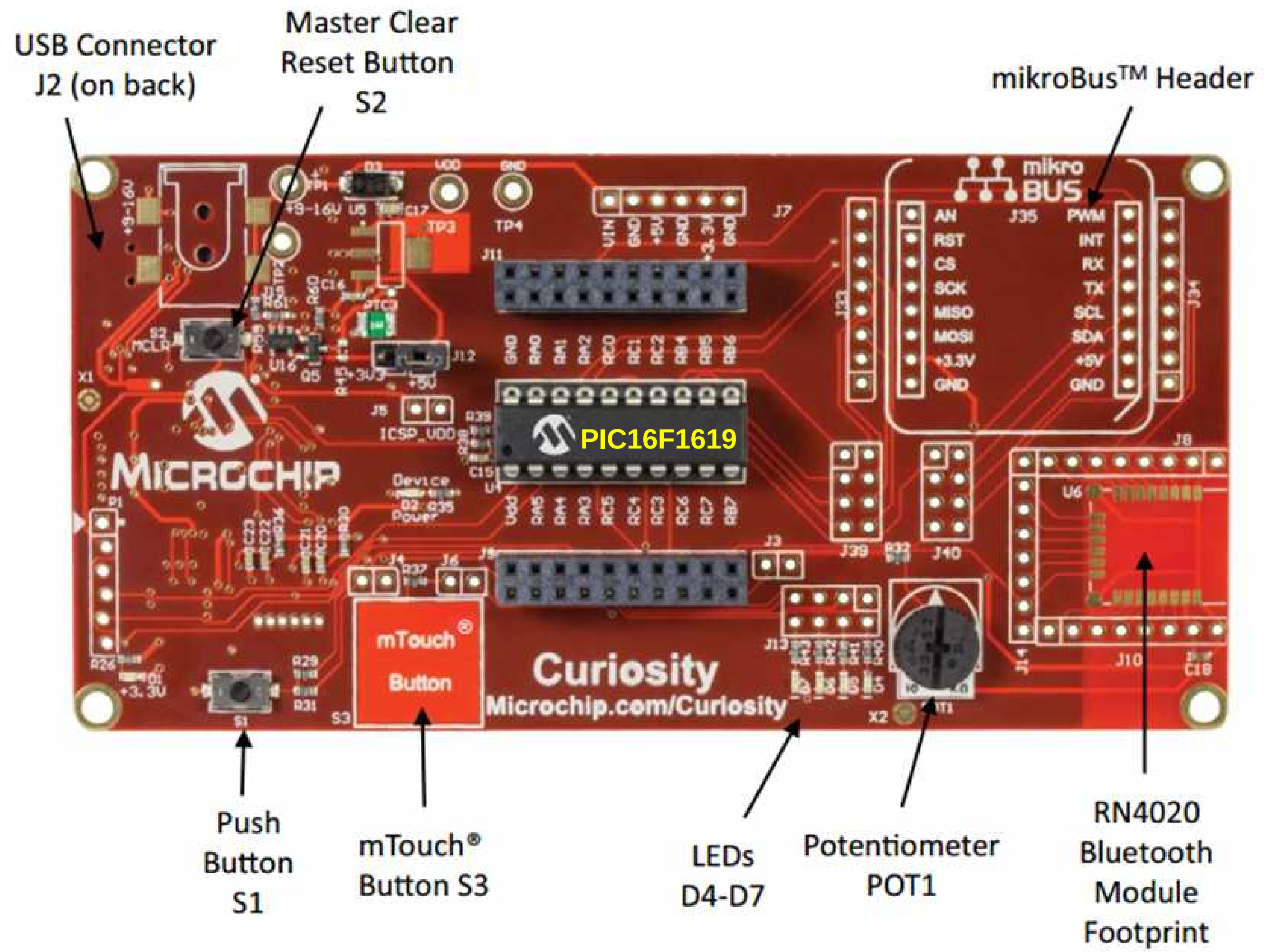
驗證:

- MPLAB X IDE 正常
- MPLAB XC 編譯正常
- Curiosity 板連線正常
- MCC 功能正常

實驗手冊第 **7** 頁

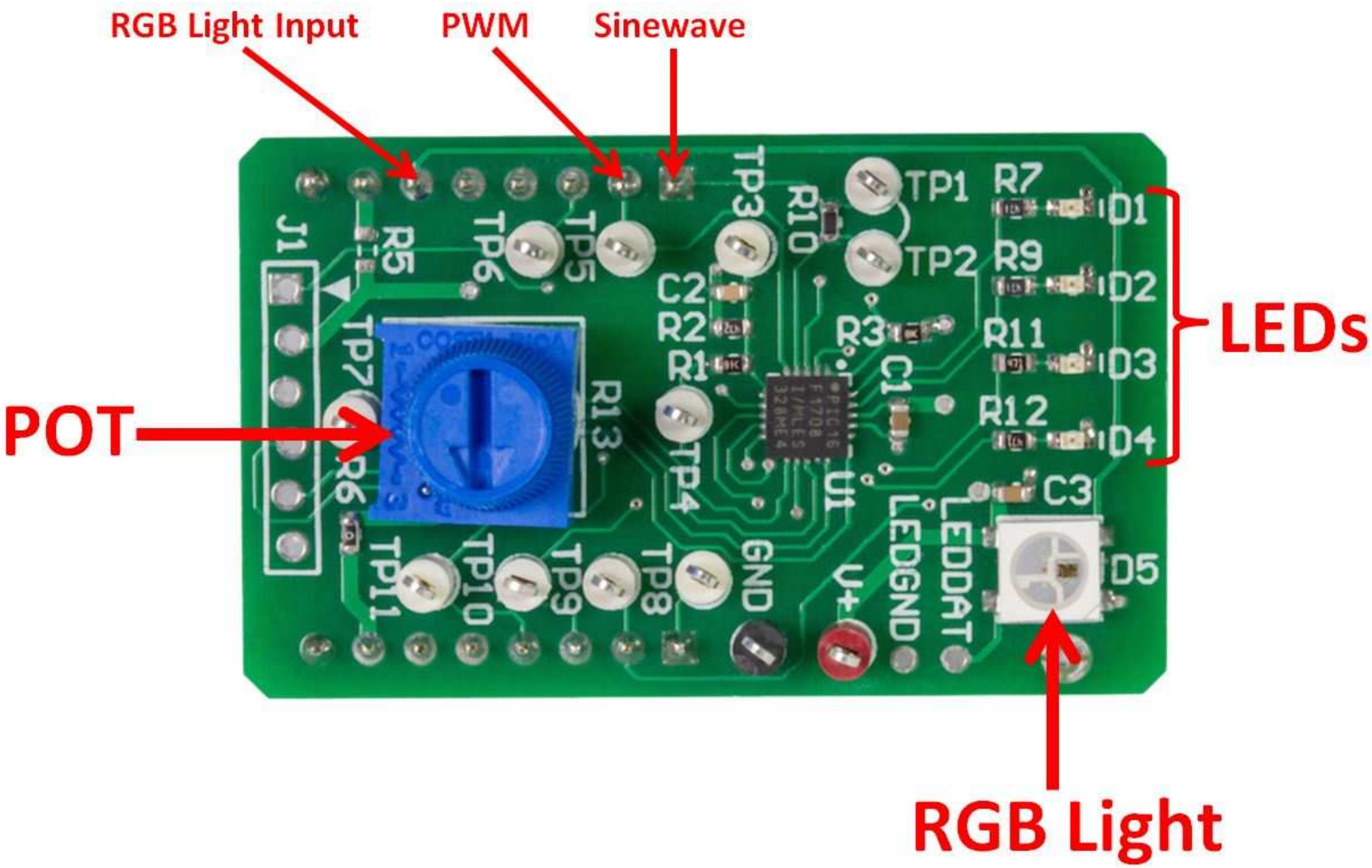
MICROCHIP Regional Training Centers

Curiosity Board Overview



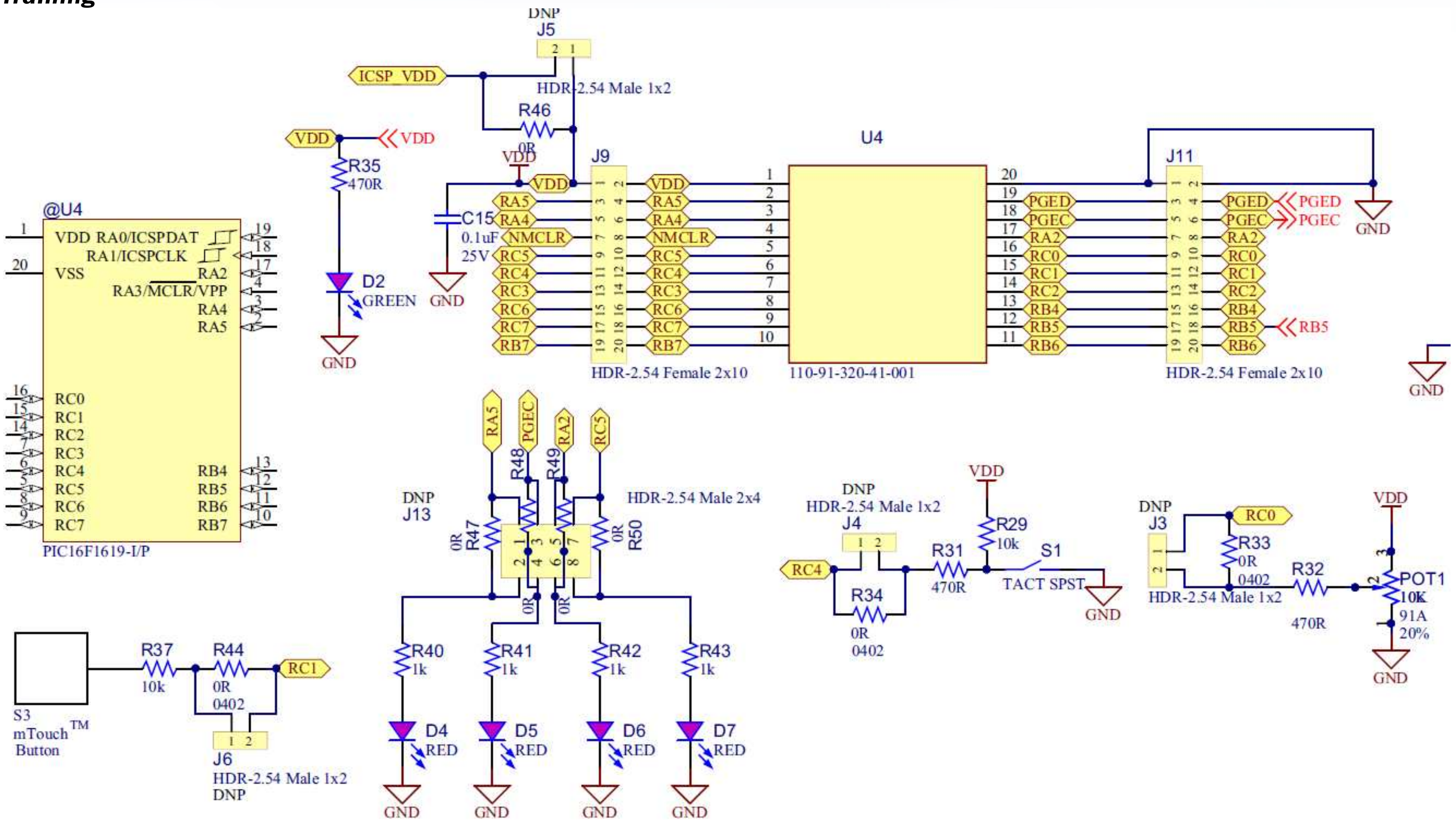
Regional Training Centers

Click Board Overview
CIP V2B



MICROCHIP
Regional Training Ce

Curiosity 板電路圖



Lab 1

Lab1 是 MCC 基礎練習，試著使用 MCC 來設定 Timer1 的計時 0.25 秒，並在中斷裡將一顆 LED 做轉態做測試。

實驗手冊第 11 頁

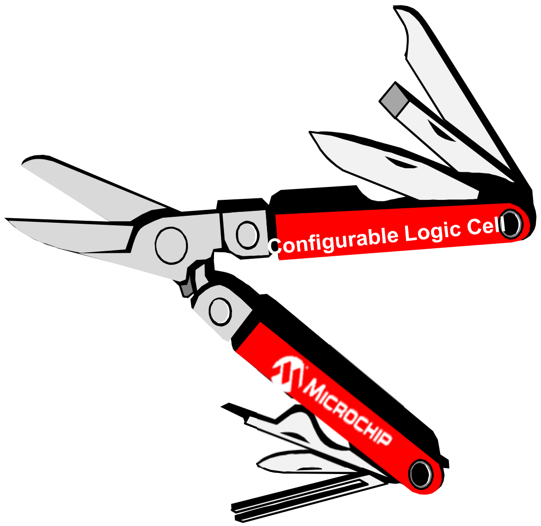
Regional Training Centers

Configurable Logic Cell (CLC)

Microchip Regional Training Centers

CLC 的效益

- 一種可程式化的內建邏輯及正反器電路，可因應設計需求加以設定其簡潔又高速的硬體功能
- 降低外部零件需求及 **PCB** 尺寸
- 純硬體的快速事件反應
- 獨自運作的週邊
- 即使在睡眠模式下仍可低功耗運作



Microchip Regional Training Centers

CLC 功能

- **4** 個輸入及 **1** 個輸出的邏輯電路方塊
- 邏輯電路八種設定**:**
  - AND-OR
  - OR-XOR
  - AND
  - S-R Latch
  - D-Flipflop
  - OR into D-Flipflop
  - J-K Flipflop
  - D-Latch
- **Inputs selectable from nearly every peripheral and I/O pin on device**
- **Output steerable to many peripherals and most pins.**

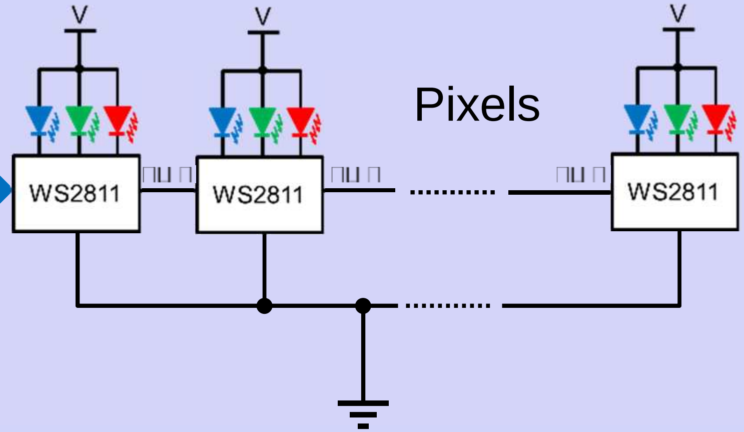
三色 LED 驅動範例

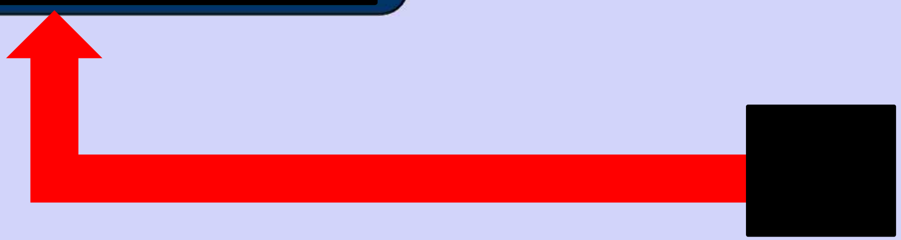
Microchip Regional Training Centers

Application

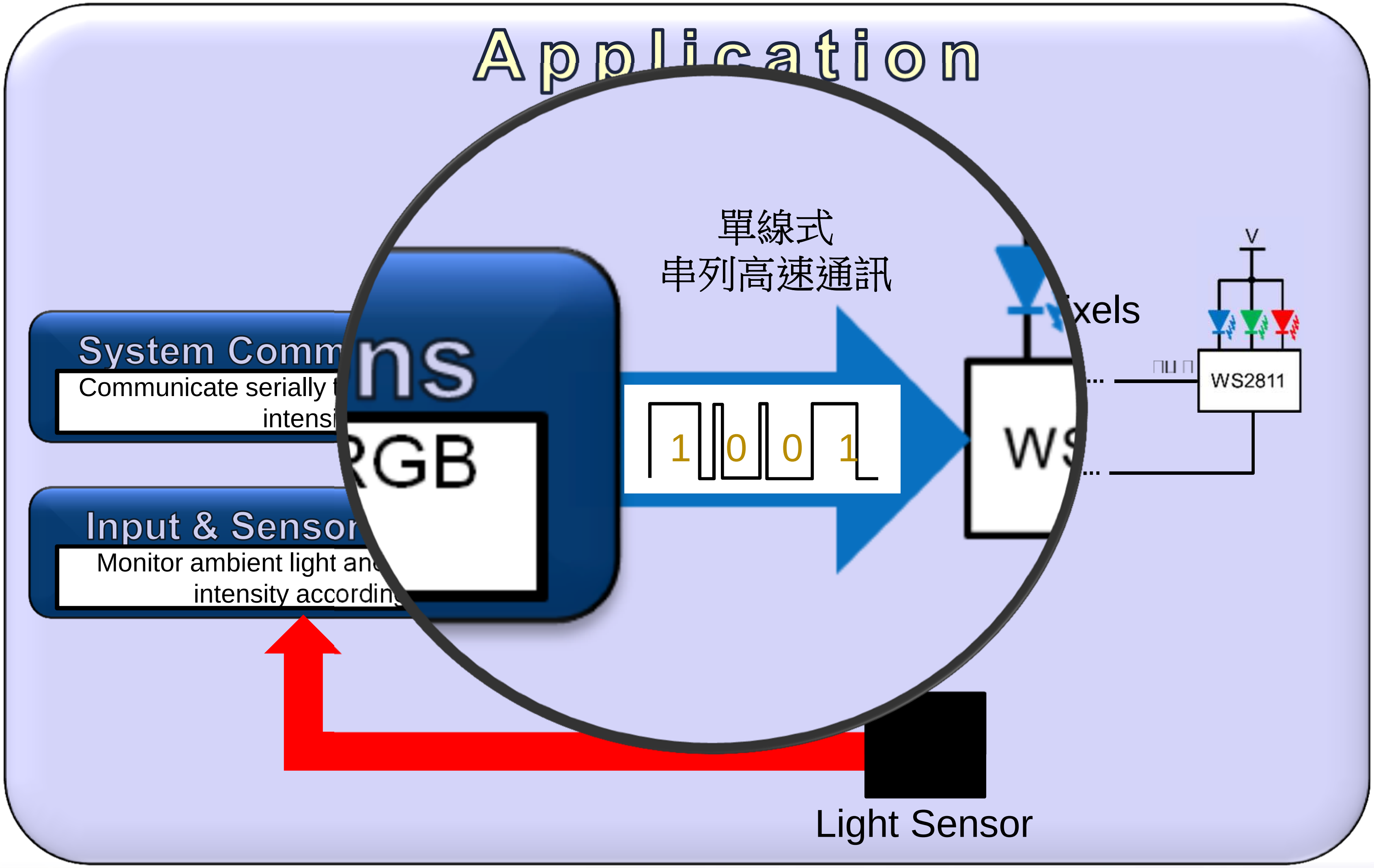
System Communications
Communicate serially to each pixel RGB intensity

Input & Sensor Interface
Monitor ambient light and adjust pixel intensity accordingly

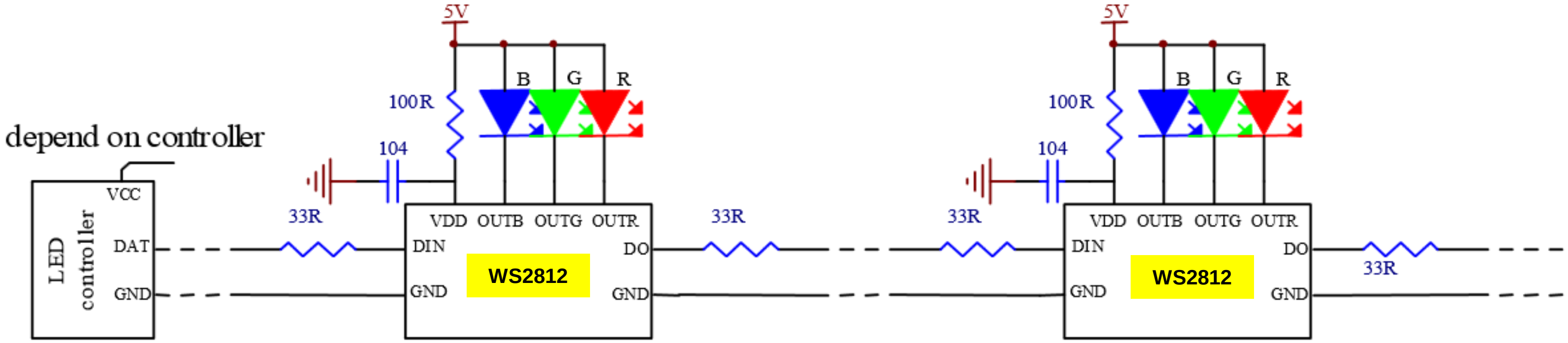




三色 LED 驅動範例

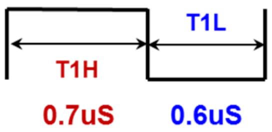

Application
單線式
串列高速通訊
System Comm
ns
Communicate serially t
intensi
RGB
Input & Sensor
Monitor ambient light an
intensity accordin
1
0
0
1
WS
ixels
V
WS2811
Light Sensor

WS2812 三色 LED

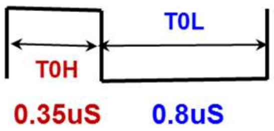


WS2812 24 位元傳輸方式 : 綠，紅，藍

G7, G6, G5, G4, G3, G2, G1, G0, R7, R6, R5, R4, R3, R2, R1, R0, B7, B6, B5, B4, B3, B2, B1, B0



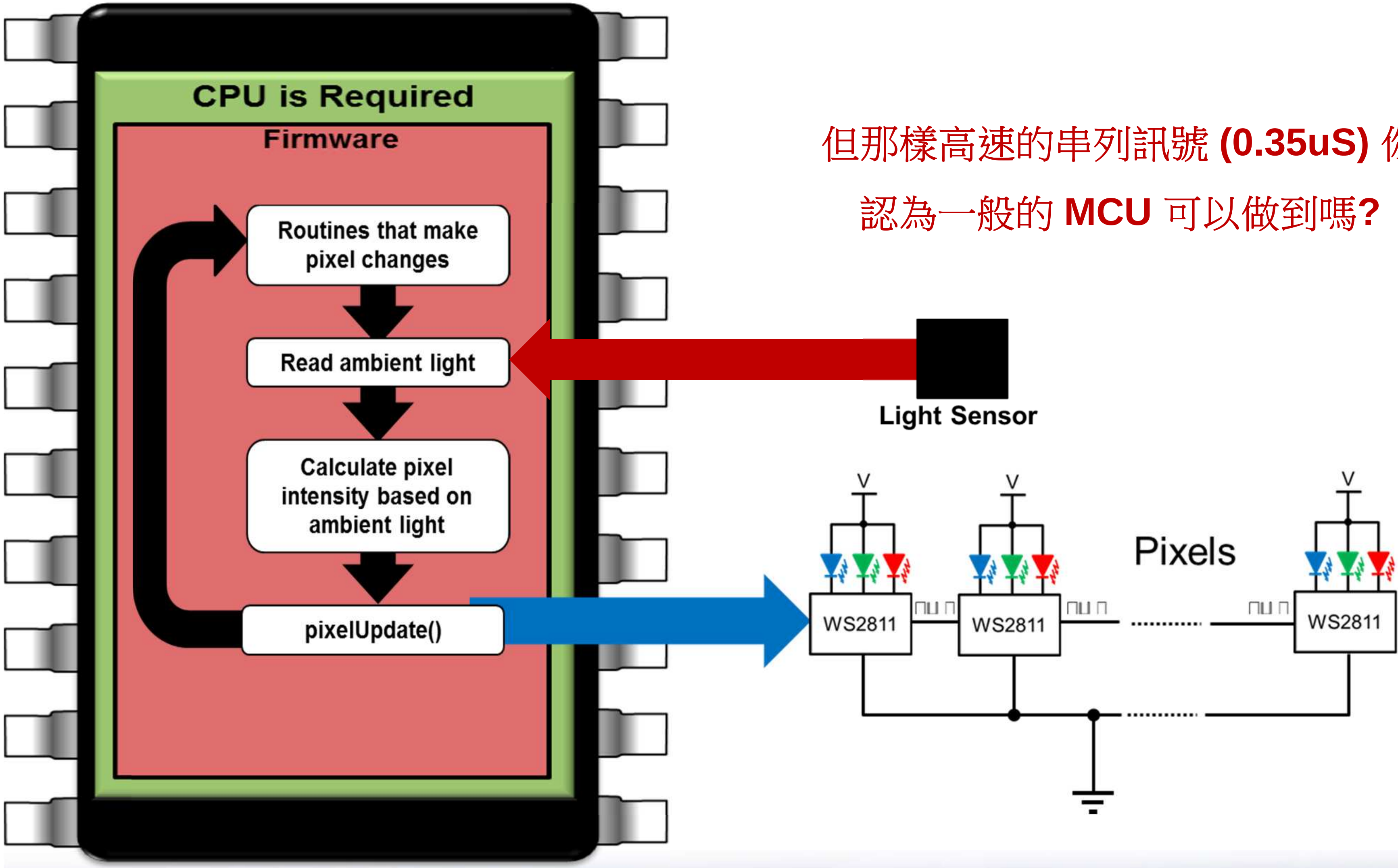
位元 **1** 的編碼



位元 **0** 的編碼

T1H 時間	位元 1 的編碼，Hi 週期所暫的時間	0.7uS	+ - 150nS
T1L 時間	位元 1 的編碼，Low 週期所暫的時間	0.6uS	+ - 150nS
T0H 時間	位元 0 的編碼，Hi 週期所暫的時間	0.35uS	+ - 150nS
T0L 時間	位元 0 的編碼，Low 週期所暫的時間	0.8uS	+ - 150nS
Reset	Low 電壓持續時間	大於 50uS 以上	

一般認為的解決方案

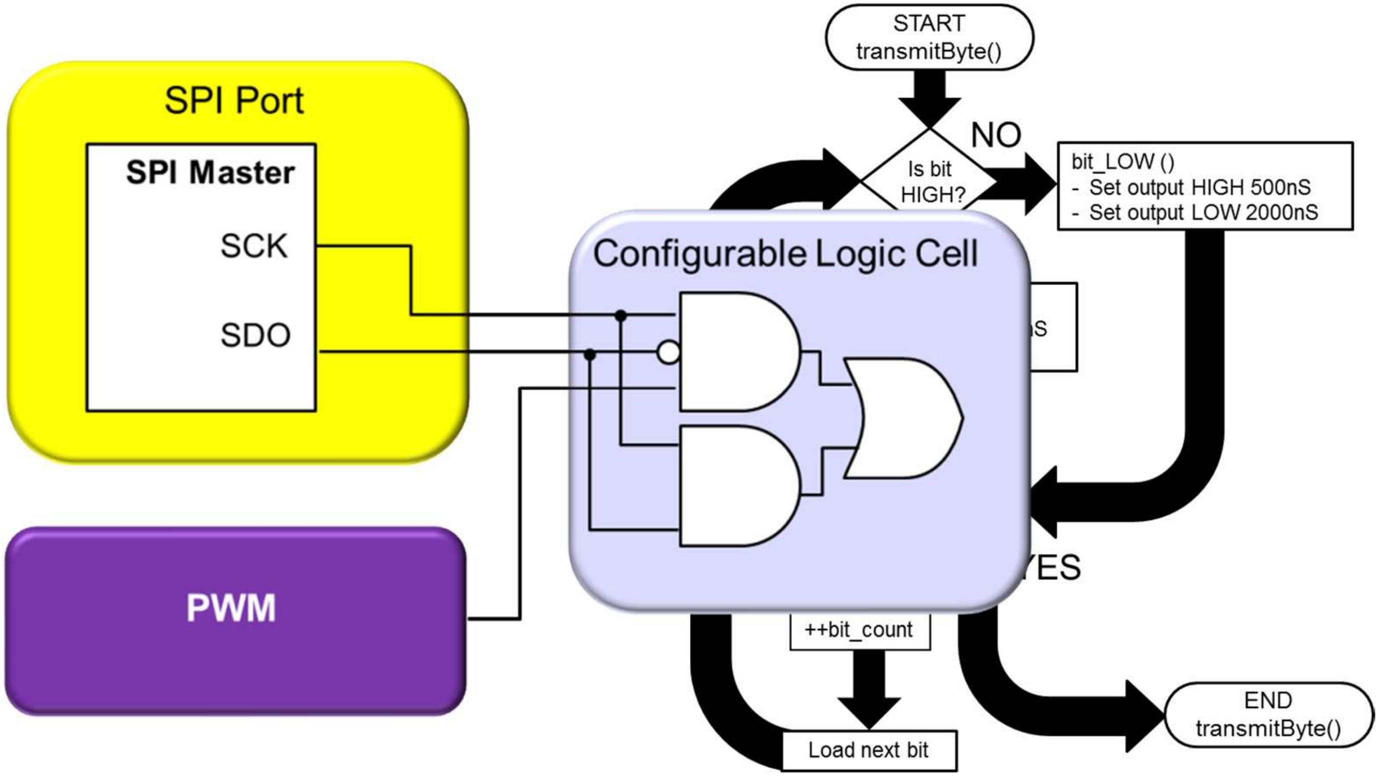


但那樣高速的串列訊號 **(0.35uS)** 你

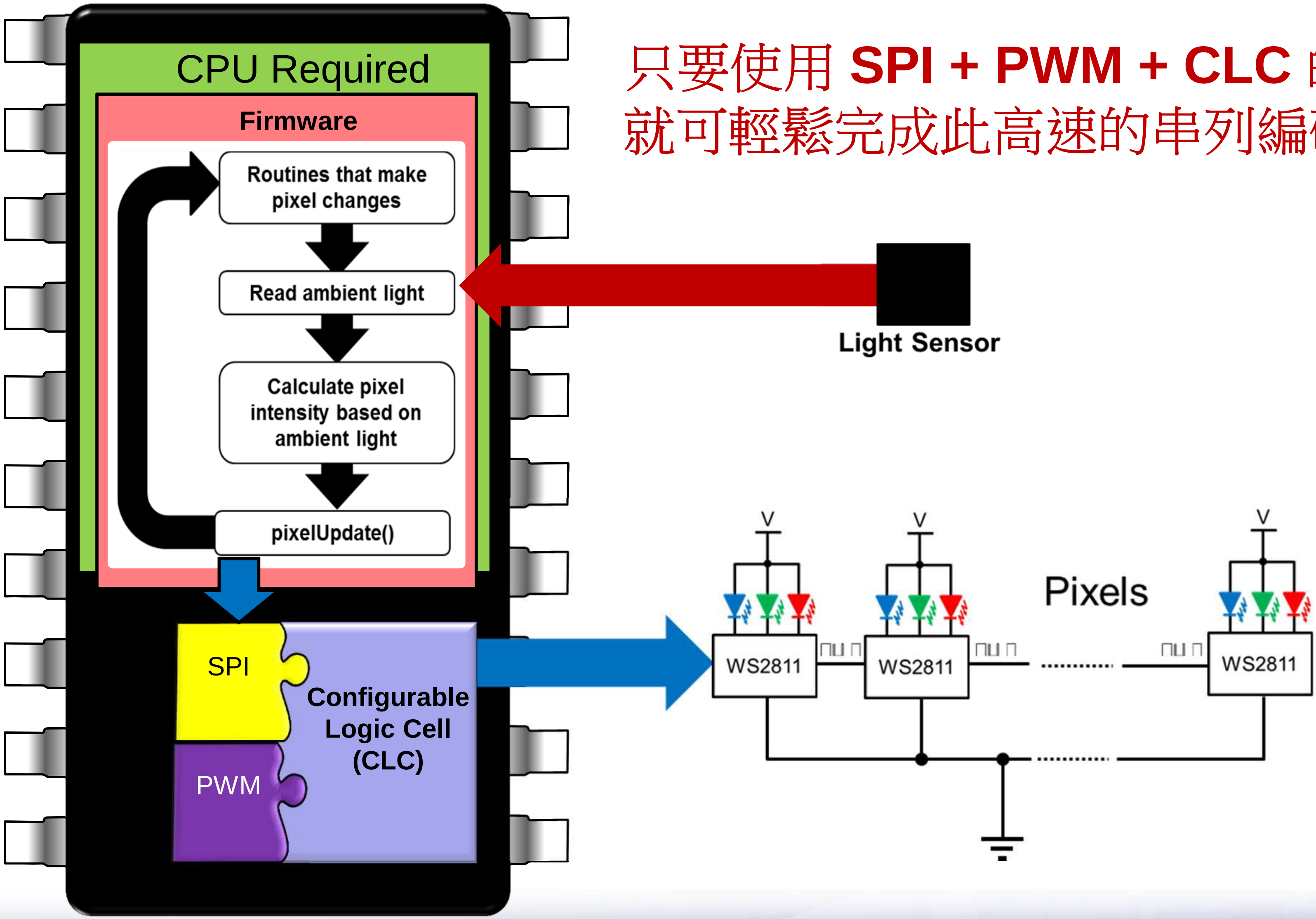
認為一般的 **MCU** 可以做到嗎**?**

使用 PIC16F1 xx 解決方案

CIPs 可以輕鬆的解決韌體與硬體的問題



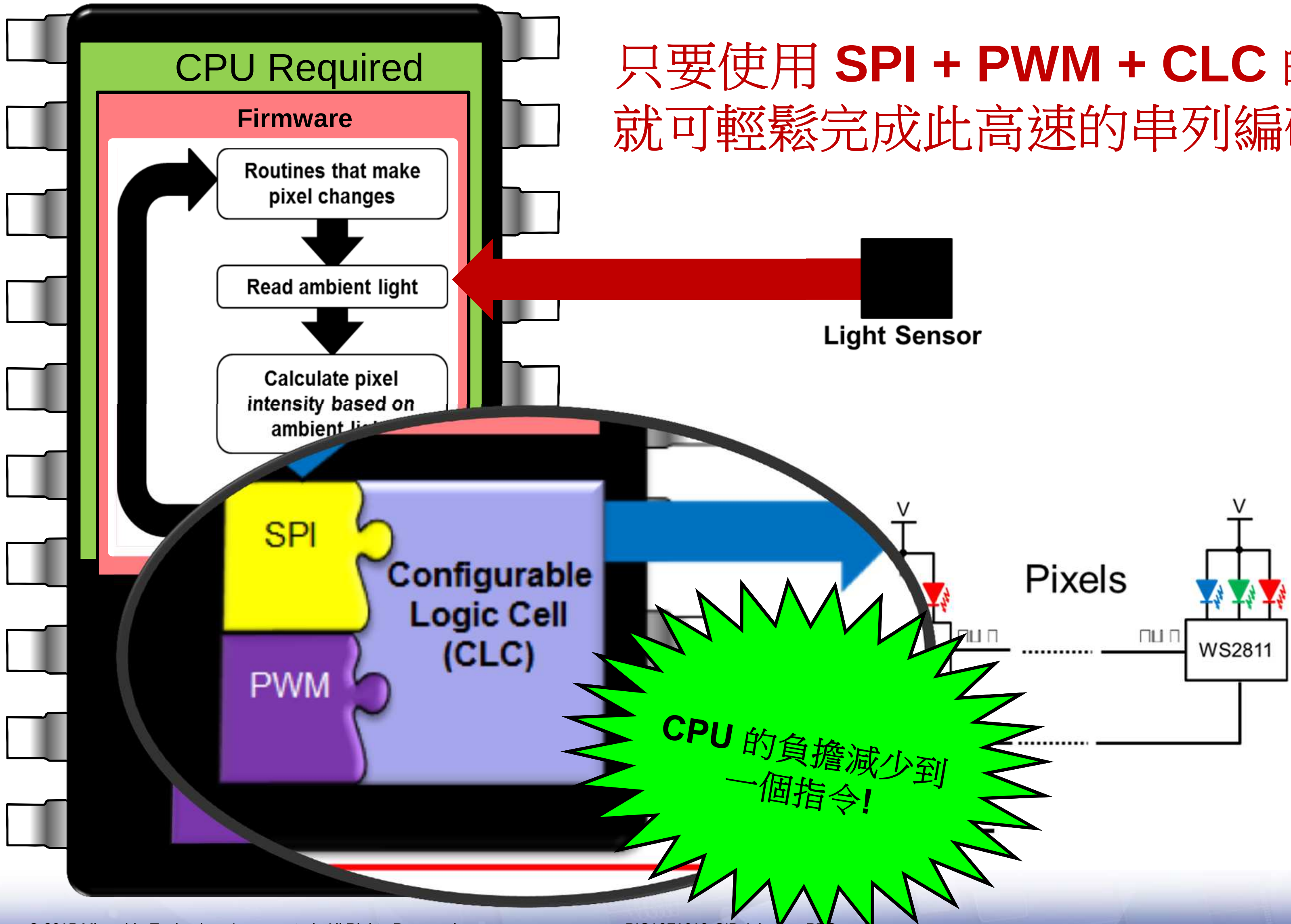
使用 PIC16F1 xx 解決方案



只要使用 **SPI + PWM + CLC** 的組合
就可輕鬆完成此高速的串列編碼傳輸

使用 PIC16F1 xx 解決方案

只要使用 **SPI + PWM + CLC** 的組合
就可輕鬆完成此高速的串列編碼傳輸



Lab 2

利用 CLC，PWM，SPI 周邊來控制 RGB WS2812 單線式串列 LED 的顯示

實驗手冊第 **14** 頁

Lab 2-1

藉由修改 main.c 的程式來達到 G、R、B 及白色的呼吸燈顯示範例。

實驗手冊第 **25** 頁

Regional Training Centers

Signal Measurement Timer (SMT)

訊號量測計時器概述

- 24 bit Timer/Counter @ FOSC resolution (32 MHz)
- Edge to edge measurements
- Measure Period & Duty cycle simultaneously
- Compare with set value – overflow Trigger
- Gate timer by window modes
- Generate interrupts on any event
- Pulse-Width Acquisition
- Period Acquisition
- Counter Overflow

訊號量測計時器 (SMT)

改良自 **Timer1 Gate** 的功能:

- Extended to 24-bits
- 2 independent gated counters with individual capture registers
- Interrupts on all events
- 11 Modes of operation
- Multitude of Input AND Clock sources

SMT 的輸入選項

Input and.or Clock Sources

- PWMs
- Other timer overflows
- CCPs
- CLC Outputs
- ZCD
- Comparators
- I/O pins (any pin via PPS – ST or TTL based on pin setup)

Lab 3

利用 SMT Gated Timer周邊來測量子板上 PWM 輸出的 Duty Cycle

實驗手冊第 **26** 頁

Lab 3-1

設定 SMT 工作模式為 “Period and Duty Cycle Acquisition” 及啟用資料獲取模式為 “Repeat”，並啟用 “Pulse Width Acquisition Interrupt” 量測 Duty Cycle

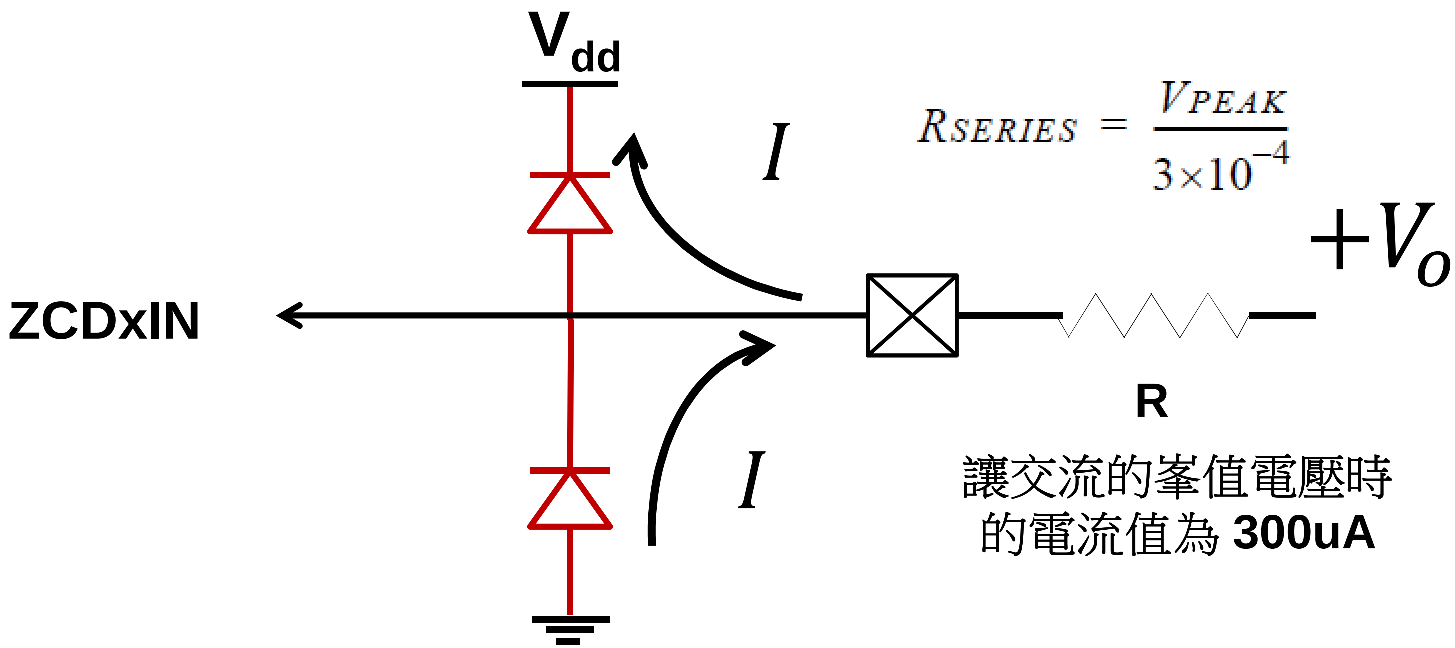
實驗手冊第 **32** 頁

Regional Training Centers

Zero Cross Detector (ZCD)

一般 I/O 方式

- 在高壓輸入偵測，一般需使用保護二極體對輸入的電壓做箝位的保護
- 但這種電流注入會導致不穩定的類比現像



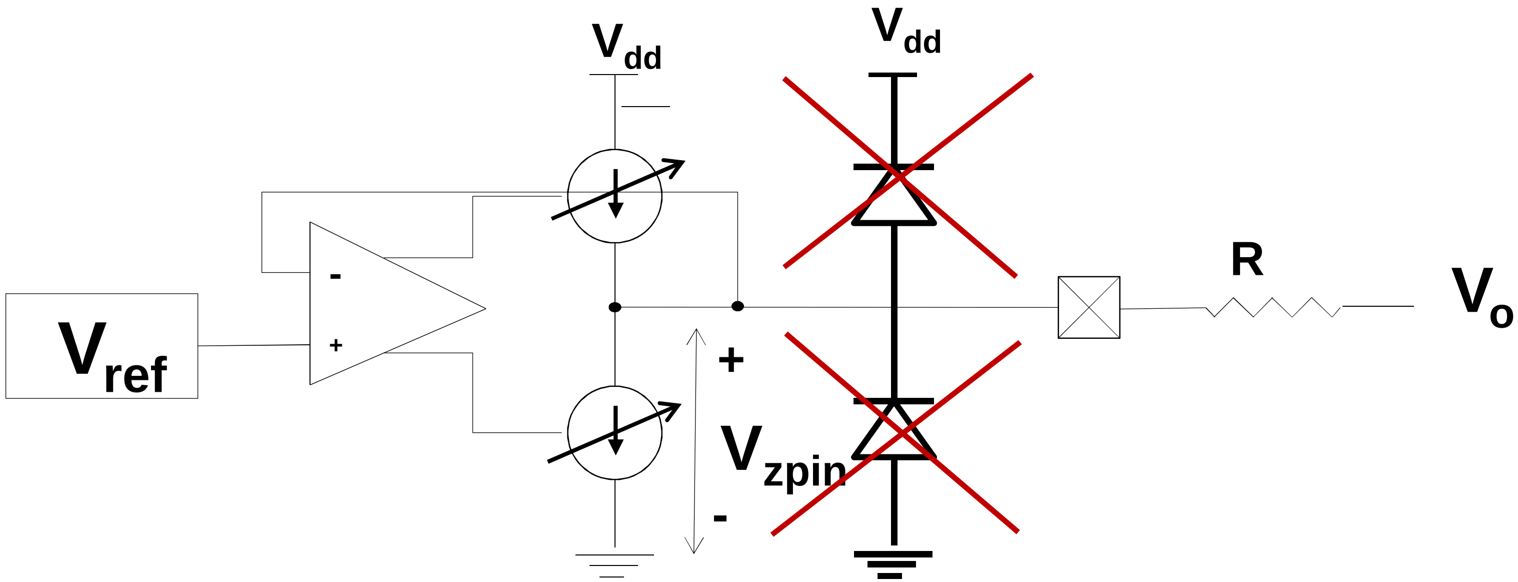
$$R_{SERIES} = \frac{V_{PEAK}}{3\times10^{-4}}$$

讓交流的峯值電壓時
的電流值為 **300uA**

TB3013 for more information on current injection.

改良後的方式

- 注意在 **ZCD** 架構下輸入腳永遠會有 **~ 750mV** 的電壓 **(Vzpin)**
- 保護二極體無需再連接



Lab 4

利用 ZCD 周邊來測量子板

上所輸入的弦波偵測出電壓

零交越點

實驗手冊第 **33** 頁

Regional Training Centers

Enhanced Timer 2

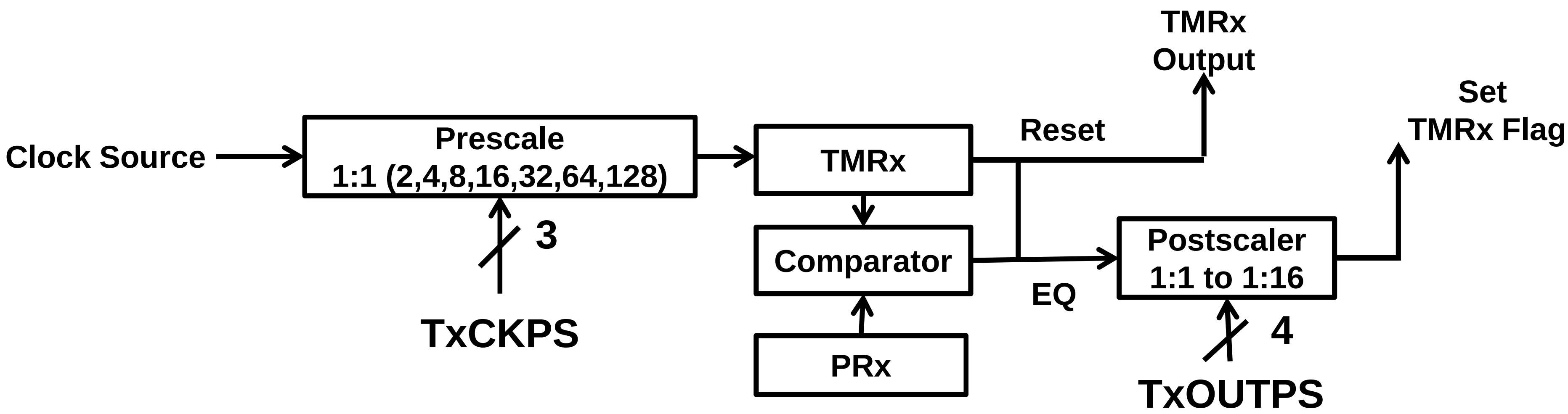
Adding HLT

基本簡介

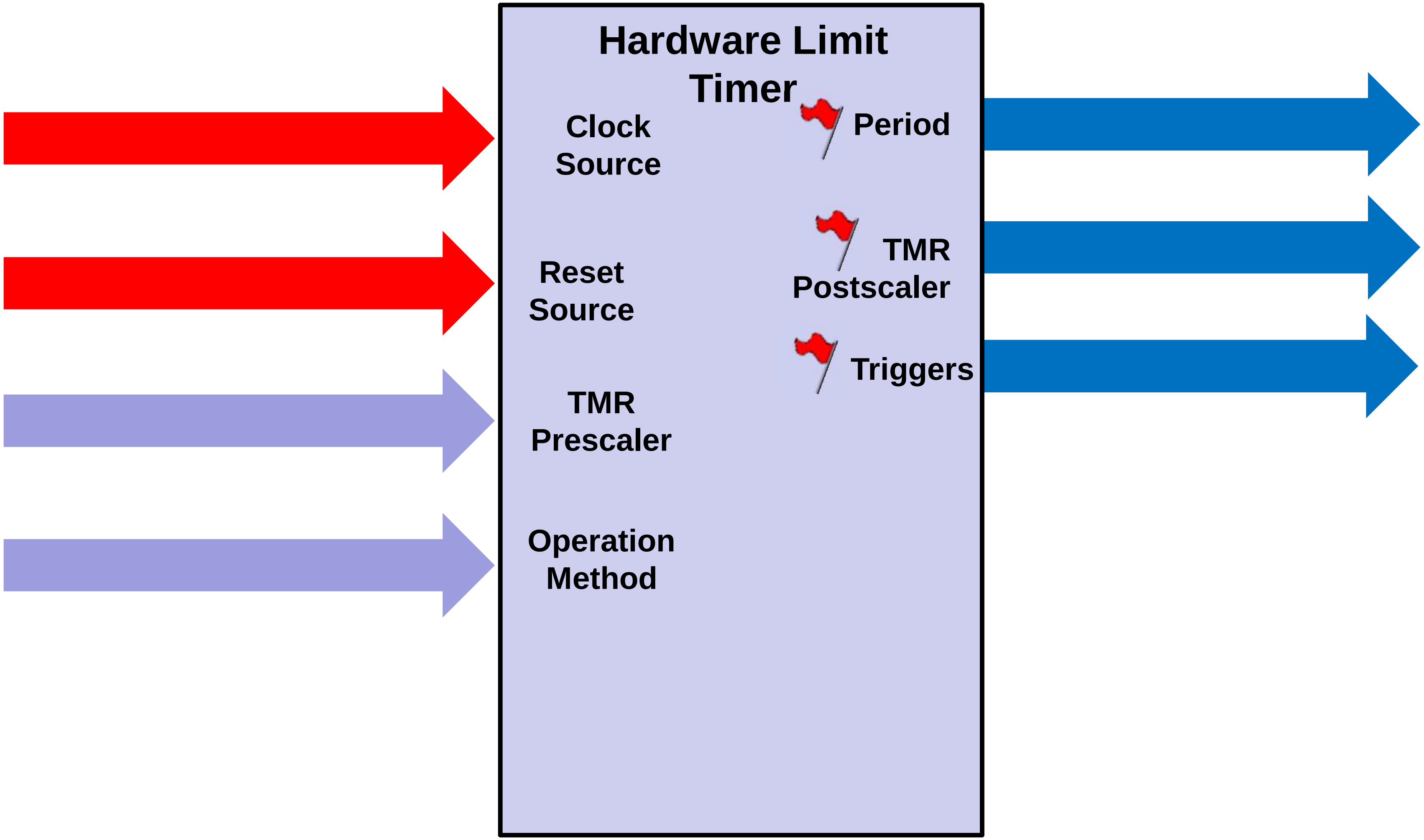
- 強化現有的 TMR2(4/6) 計時器週邊電路
- 三種主要工作模式:
  - 傳統比較模式 : Roll-Over Pulse *(Classic Method)*
  - 單穩態觸發模式 : Monostable *(8 variations)*
  - 單次觸發模式 : One-Shot *(8 variations)*
- External/Internal Triggered Start/Stop/Reset control options
- Increased options for Clock Sources and Prescaler

Roll Over Pulse Mode

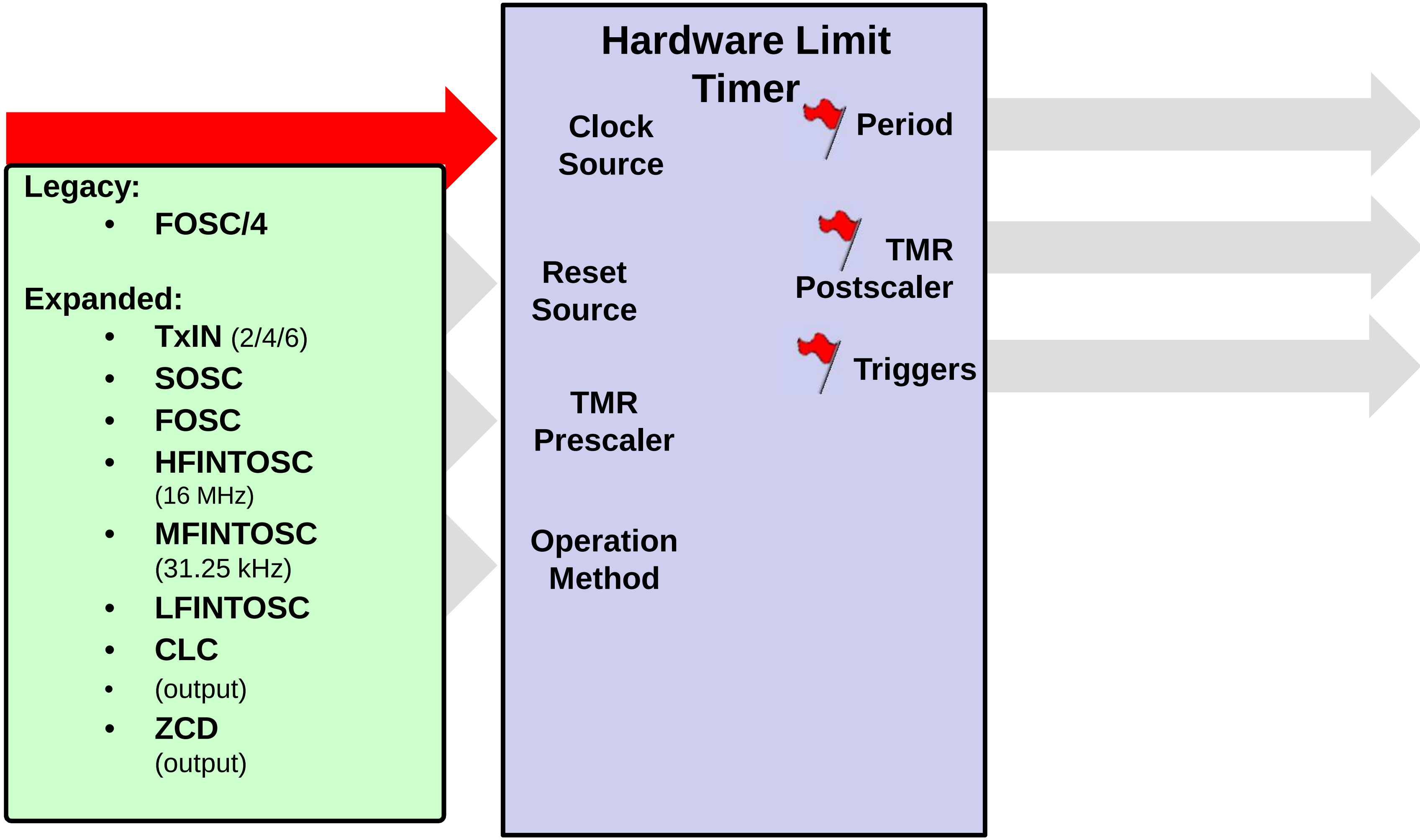
- 一般傳統 TMR2 比較模式
- PRx 設定週期時間；自動重新載入
- 收動啟動或關閉
- 輸出可串接到其他週邊或做中斷處理



HLT 簡易方塊圖


Hardware Limit Timer
Clock Source
Reset Source
TMR Prescaler
Operation Method
Period
TMR Postscaler
Triggers

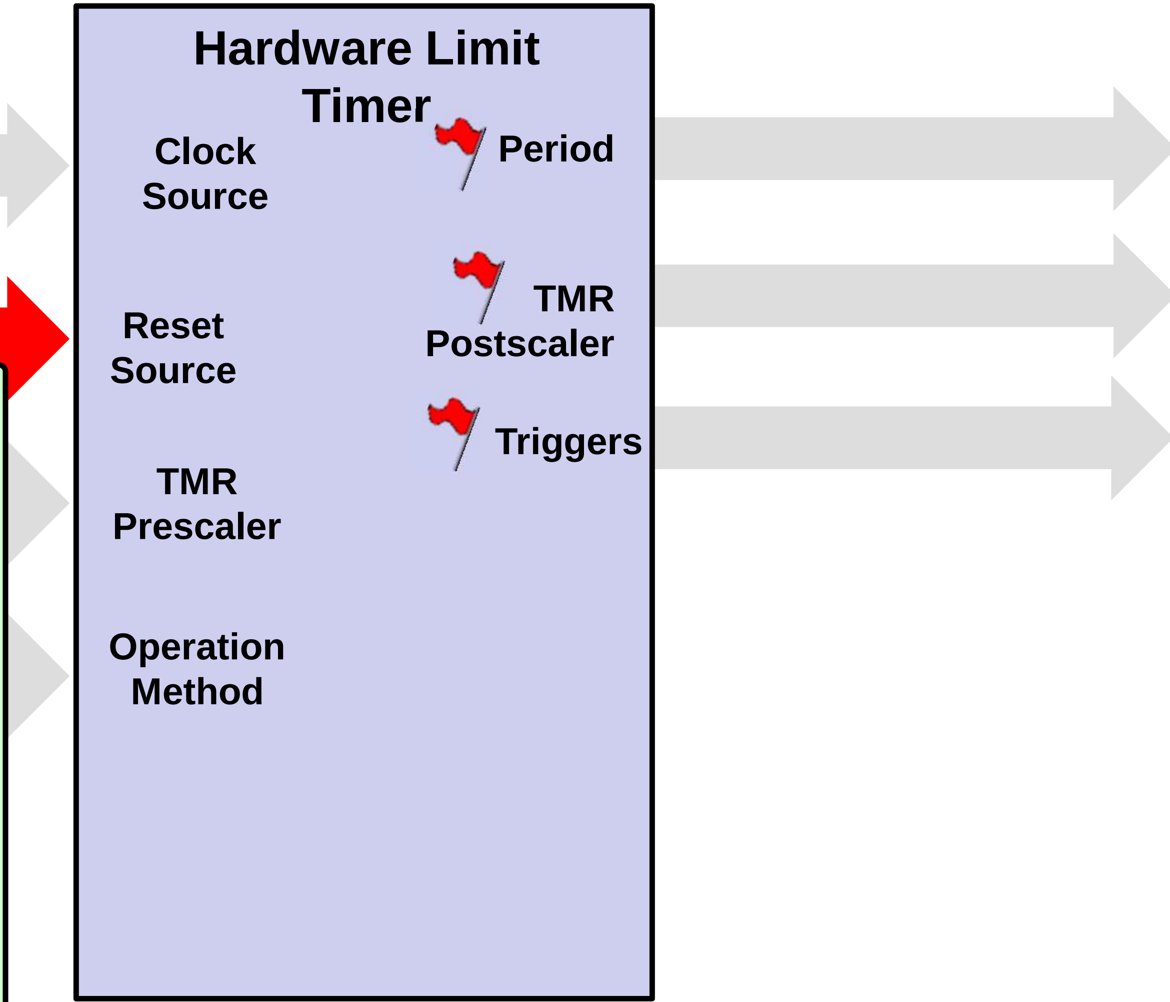
HLT 簡易方塊圖


Legacy:
• FOSC/4
Expanded:
• TxIN (2/4/6)
• SOSC
• FOSC
• HFINTOSC (16 MHz)
• MFINTOSC (31.25 kHz)
• LFINTOSC
• CLC
• (output)
• ZCD (output)
Hardware Limit Timer
Clock Source
Reset Source
TMR Prescaler
Operation Method
Period
TMR Postscaler
Triggers

HLT 簡易方塊圖

Legacy:
- N/A

Expanded:
- **TxIN** (2/4/6)
- **TMR Postscaled** (2/4/6)
- **CWG/COG**
- **CLC**
- **Comparator** (sync mode)
- **CCP/PWM**
- **ZCD**



HLT 簡易方塊圖

Hardware Limit Timer

- **Clock Source**
- **Reset Source**
- **TMR Prescaler**
- **Operation Method**

- **Period**
- **TMR Postscaler**
- **Triggers**

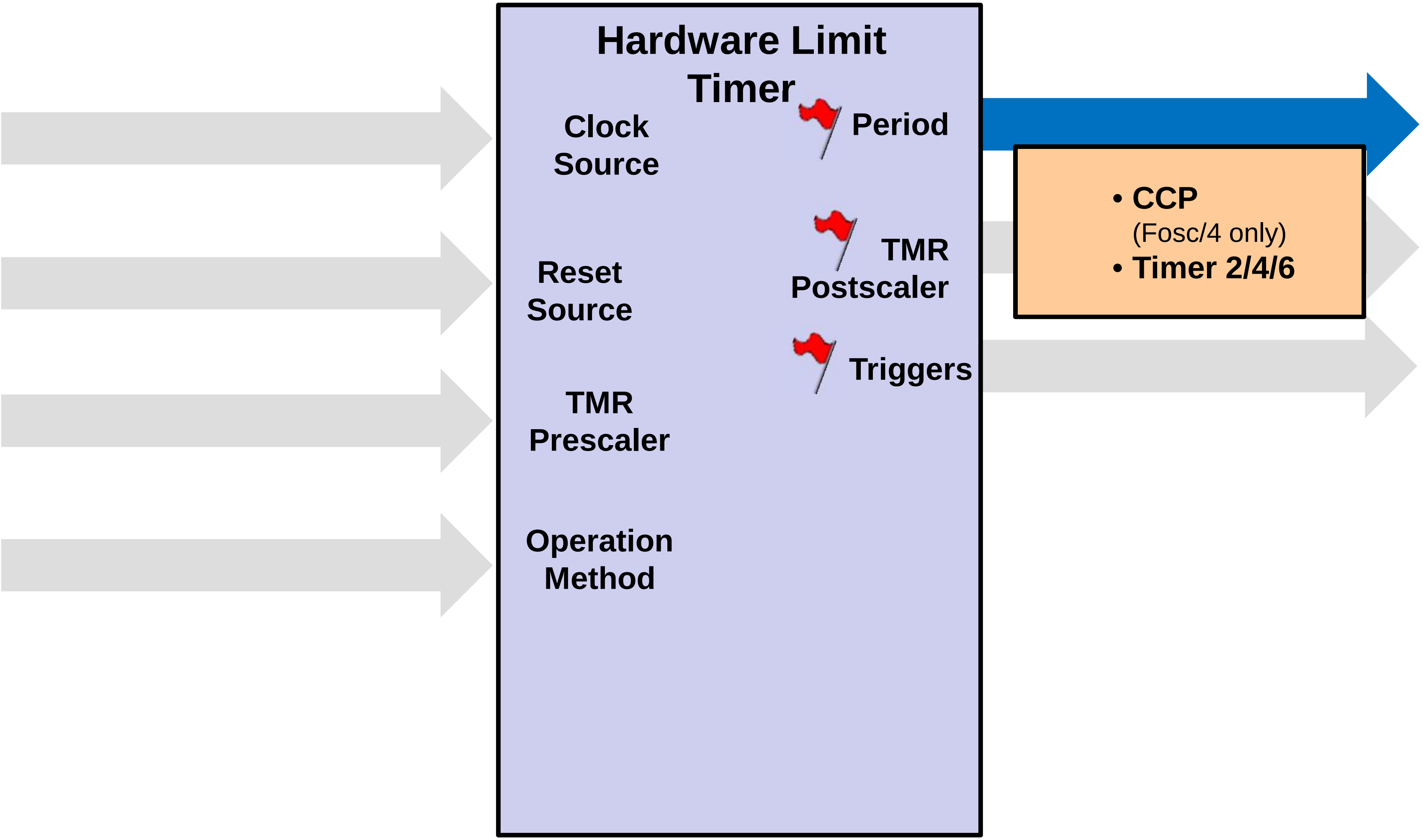
Legacy:

- **Roll-Over Pulse**
  (Repeating)

Expanded:

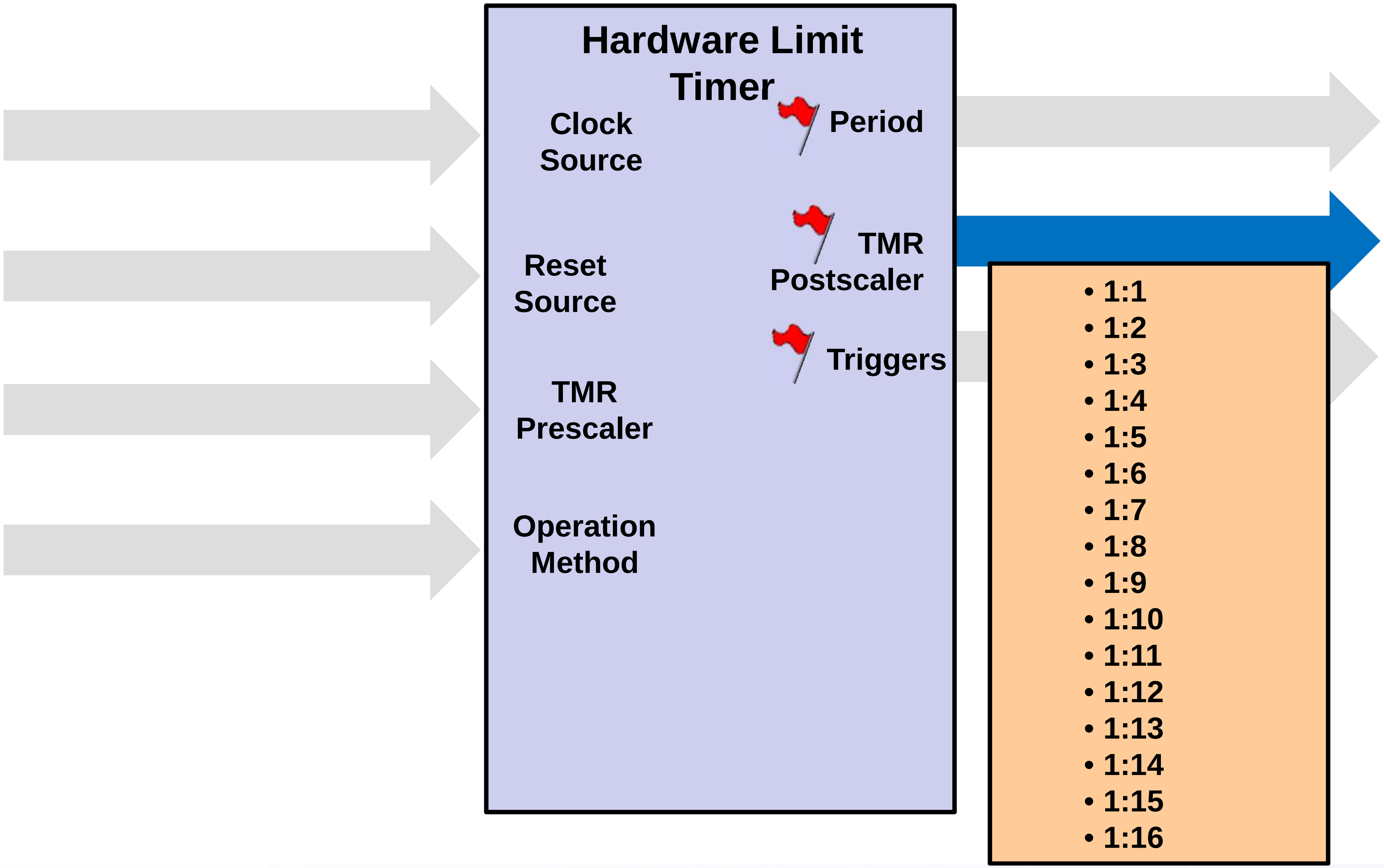
- **Monostable**
  (Reset Available)
- **One Shot**
  (Single Action)

HLT Simplified Block Diagram

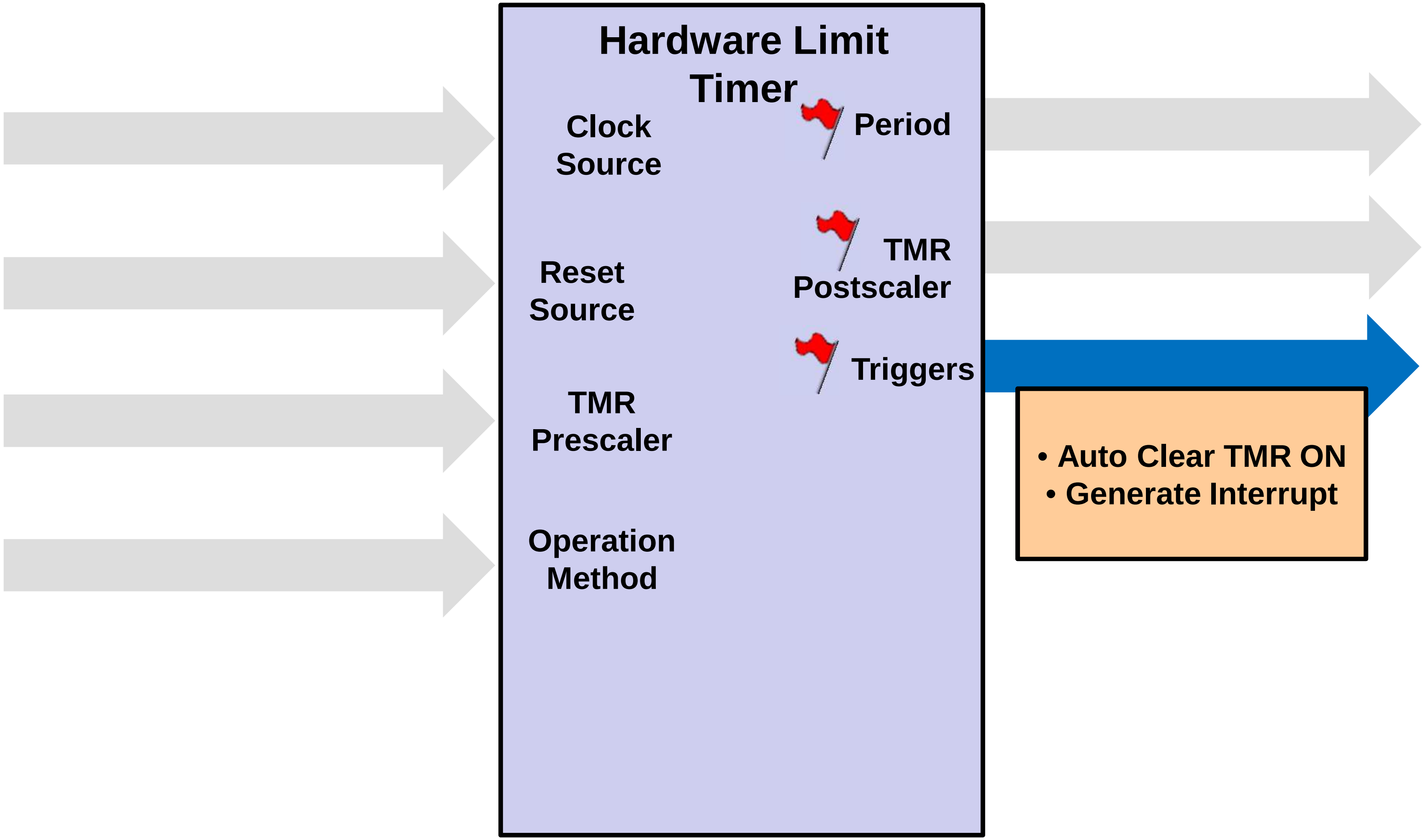


Regional Training Centers

HLT Simplified Block Diagram



HLT Simplified Block Diagram



Free Running Period

MODE	Start	Reset	Stop
Period Pulse			
000	ON = 1	---	
001	ON = 1 & RS = 1	---	ON = 0 [or] RS = 0
010	ON = 1 & RS = 0	---	ON = 0 [or] RS = 1
Period Pulse with Hardware Reset			
011	ON = 1	RS = Rising/Falling	ON = 0
100	ON = 1	RS = Rising	ON = 0
101	ON = 1	RS = Falling	ON = 0
110	ON = 1	RS = 0	ON = 0 [or] RS = 1
111	ON = 1	RS = 1	ON = 0 [or] RS = 1

PE: Period Elapse **RS:** Reset Source **ON:** TxCON <bit 7>

Monostable

每次觸發就產生固定時間的計時輸出

MODE	Start	Reset	Stop
Edge Triggered Start			
001	ON = 1 & RS = 1^{st} Rising	---	ON = 0 [or] PE *
010	ON = 1 & RS = 1^{st} Falling	---	ON = 0 [or] PE *
011	ON = 1 & RS = 1^{st} Rising/Falling		ON = 0 [or] PE *

*Next clock after PE: [TMRx = PRx]

PE: Period Elapse **RS:** Reset Source **ON:** TxCON <bit 7>

One Shot

MODE	Start	Reset	Stop
One Shot			
000	ON = 1	---	ON = 0 or PE*
Edge Triggered Start			
001	ON = 1 & RS = Rising		ON = 0 or PE*
010	ON = 1 & RS = Falling		ON = 0 or PE*
011	ON = 1 & RS = Rising/Falling		ON = 0 or PE*
Edge Triggered Start and Hardware Reset			
100	ON = 1 & RS = Rising	RS = Rising	ON = 0 or PE*
101	ON = 1 & RS = Falling	RS = Falling	ON = 0 or PE*
110	ON = 1 & RS = Rising	RS = 0	ON = 0 or PE*
111	ON = 1 & RS = Falling	RS = 1	ON = 0 or PE*
Level Triggered Start and Hardware Reset			
110	ON = 1 & RS = 1	RS = 0	ON = 0 [or] Held Reset
111	ON = 1 & RS = 0	RS = 1	ON = 0 [or] Held Reset

RS: Reset Source **ON:** TxCON <bit 7> *Next clock after PE: [TMRx = PRx]

Lab 5

利用 ZCD 偵測上升弦波交越點後，啟動 HLT/Timer2 的 Mono-Stable 控制 PWM3 輸出 1.6mS 的脈波

實驗手冊第 37 頁

Lab 6

- 利用 ZCD 偵測上升弦波交越點
- 觸發 HLT/Timer2 的 Mono-Stable 輸出單次的 1.28 mS 的脈波
- 在此 1.28mS 寬的時間框下啟動以 TMR4 的基礎的 PWM3 輸出 16 個 Duty Cycle 30% 的脈衝

實驗手冊第 **41** 頁

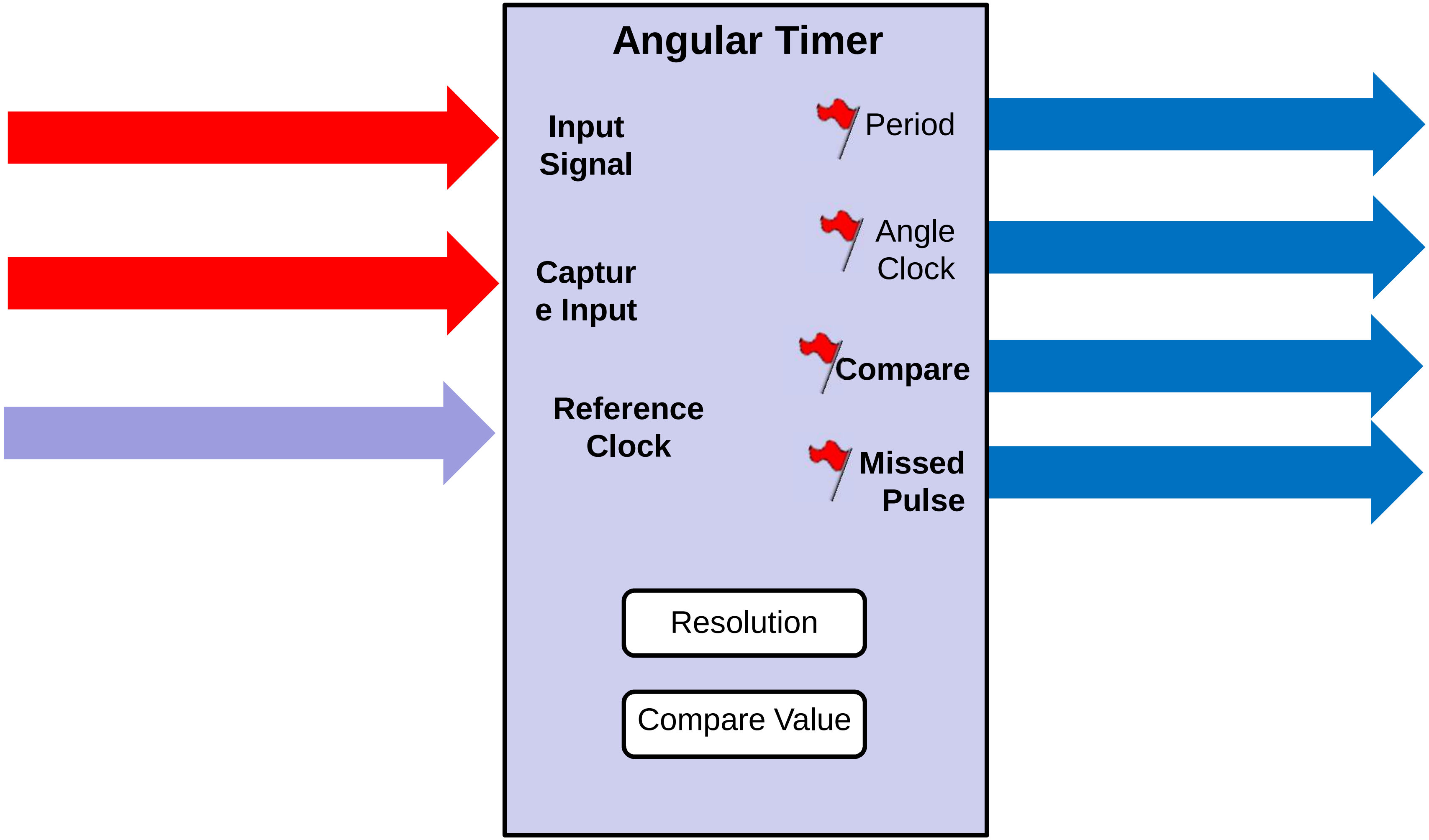
Regional Training Centers

The Angular Timer (AT)

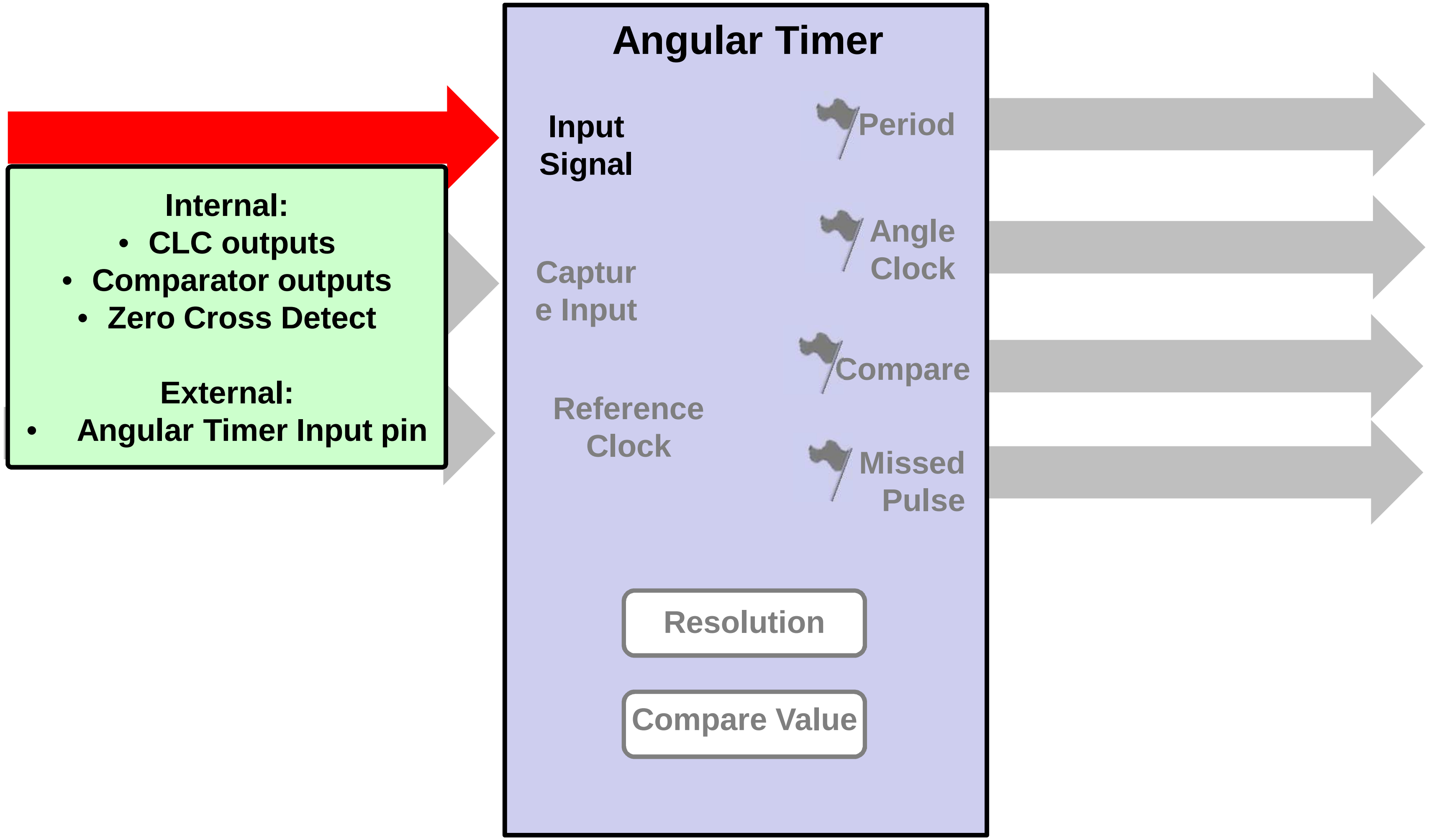
General Overview

- Multiplies an unknown time based signal into angle proportional clock
- The number of desired intervals stays constant with a change in frequency
  - Previously would be math intensive

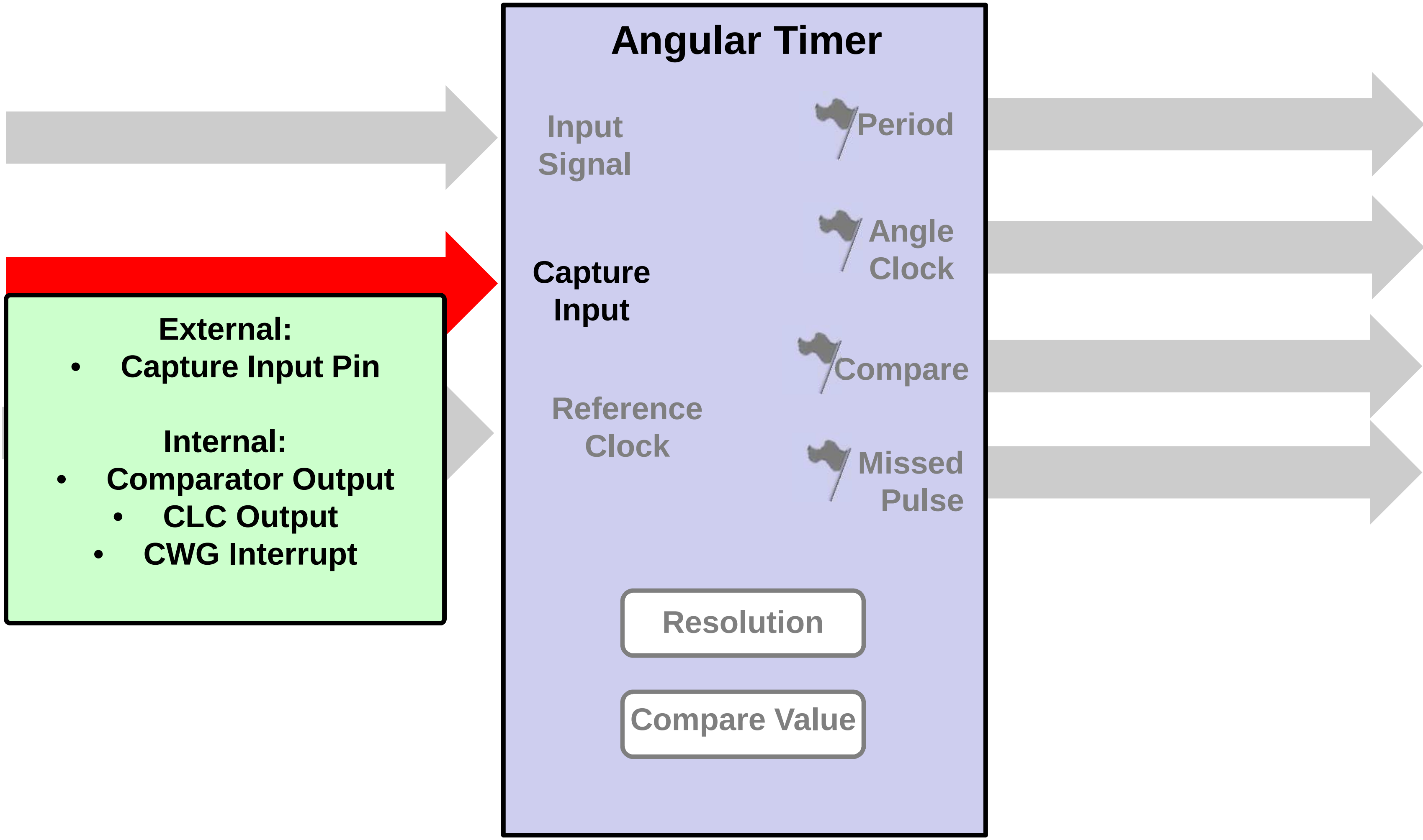
Angular Timer Simplified Block Diagram



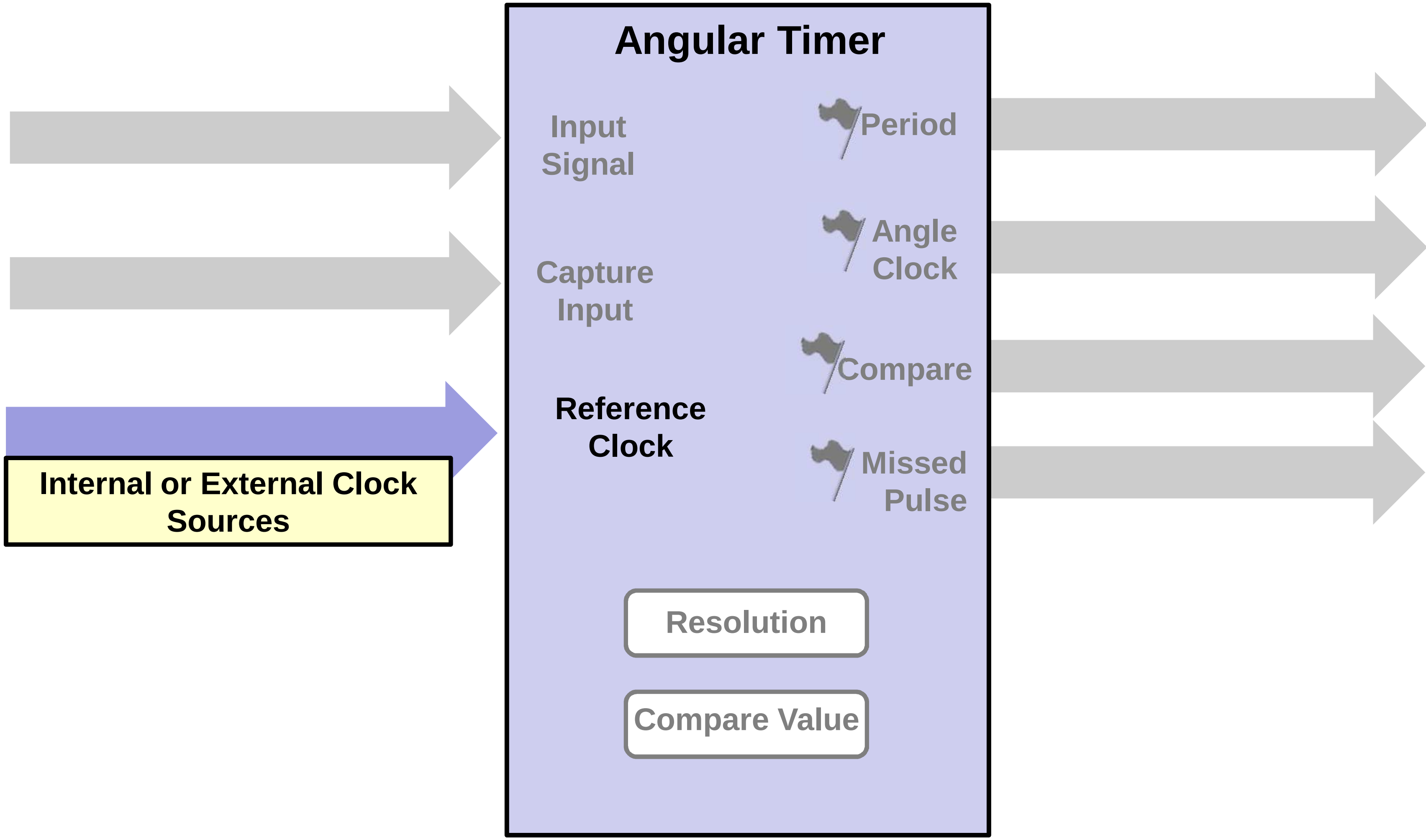
Angular Timer Simplified Block Diagram



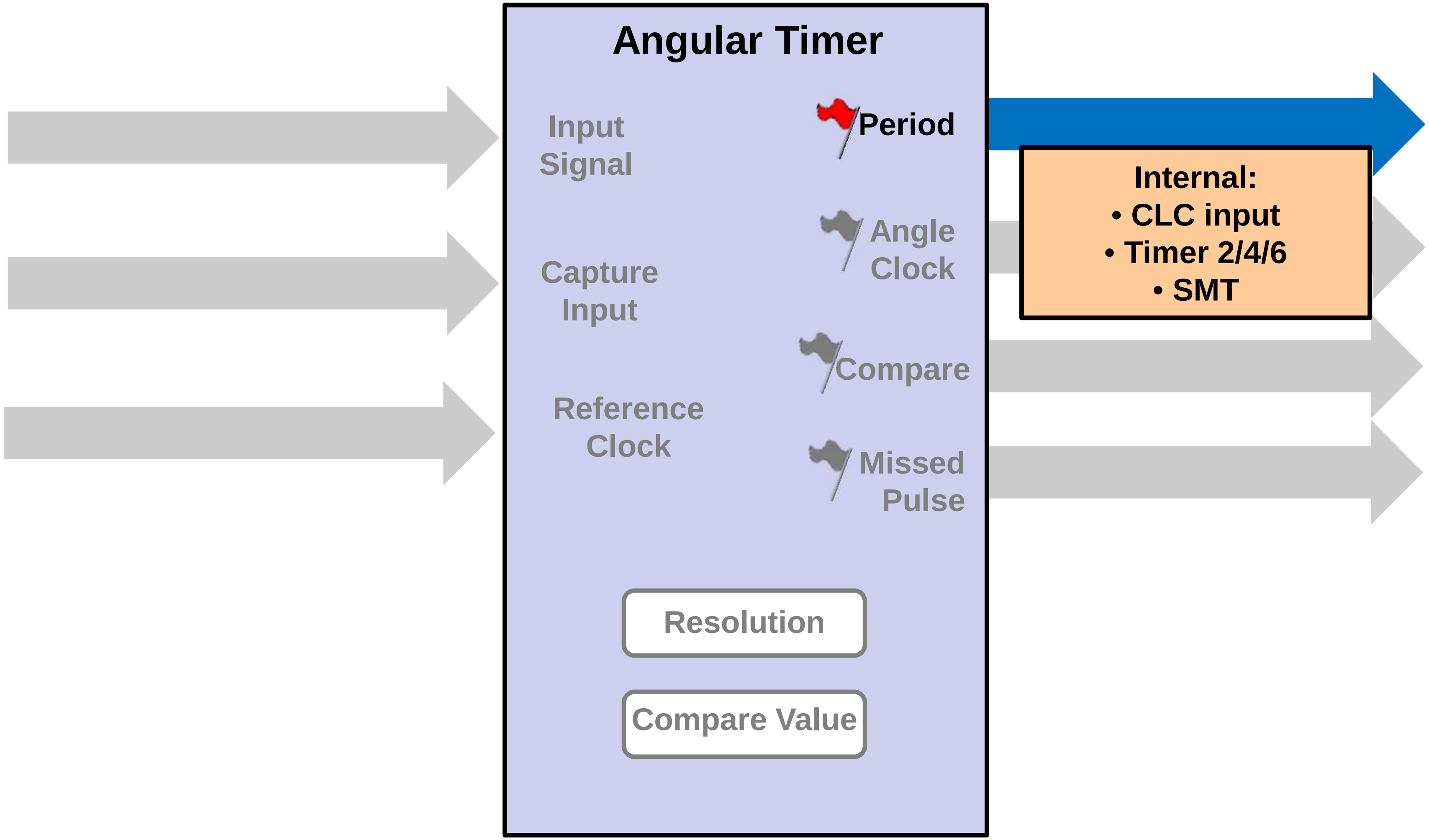
Angular Timer Simplified Block Diagram



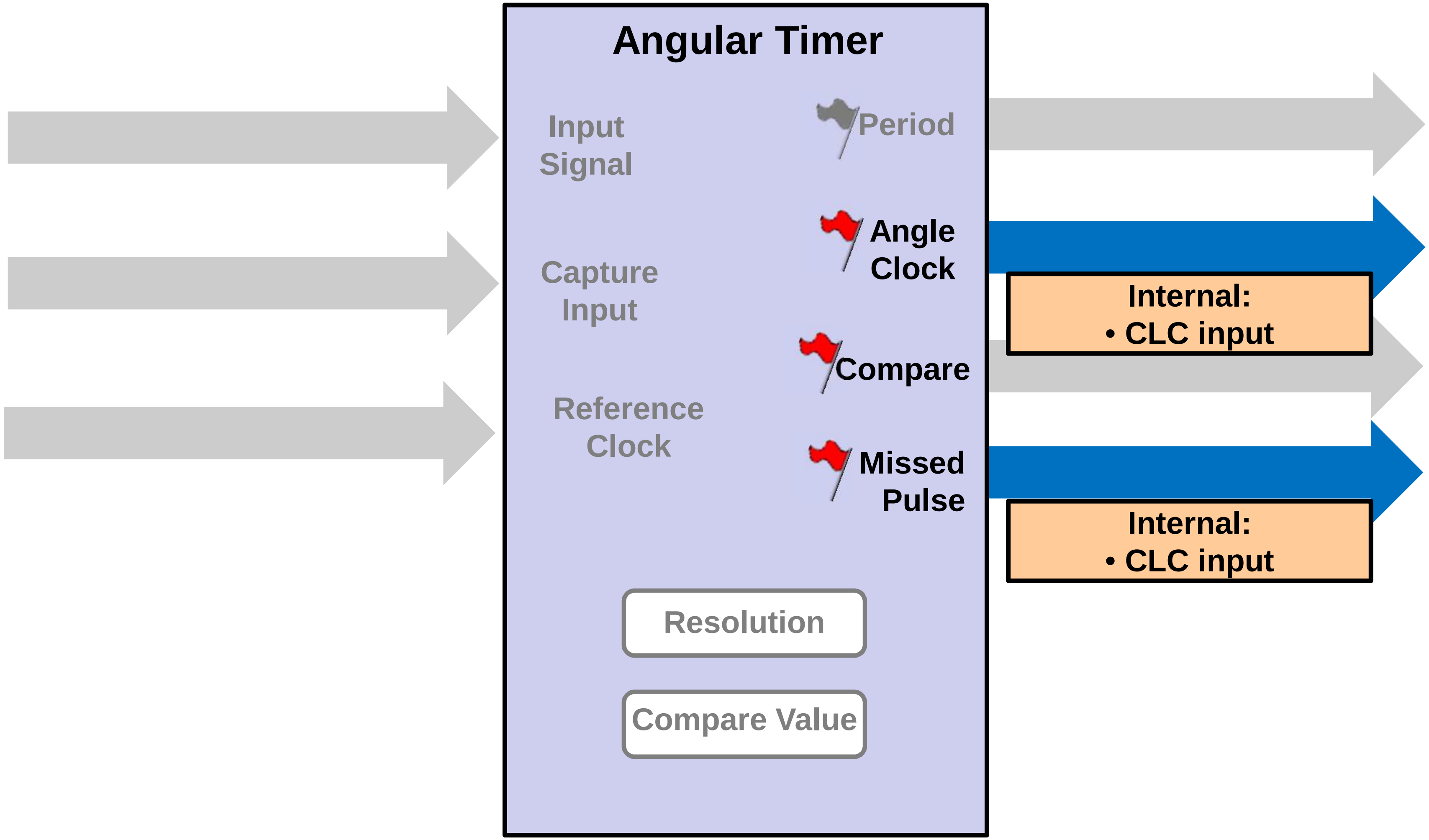
Angular Timer Simplified Block Diagram



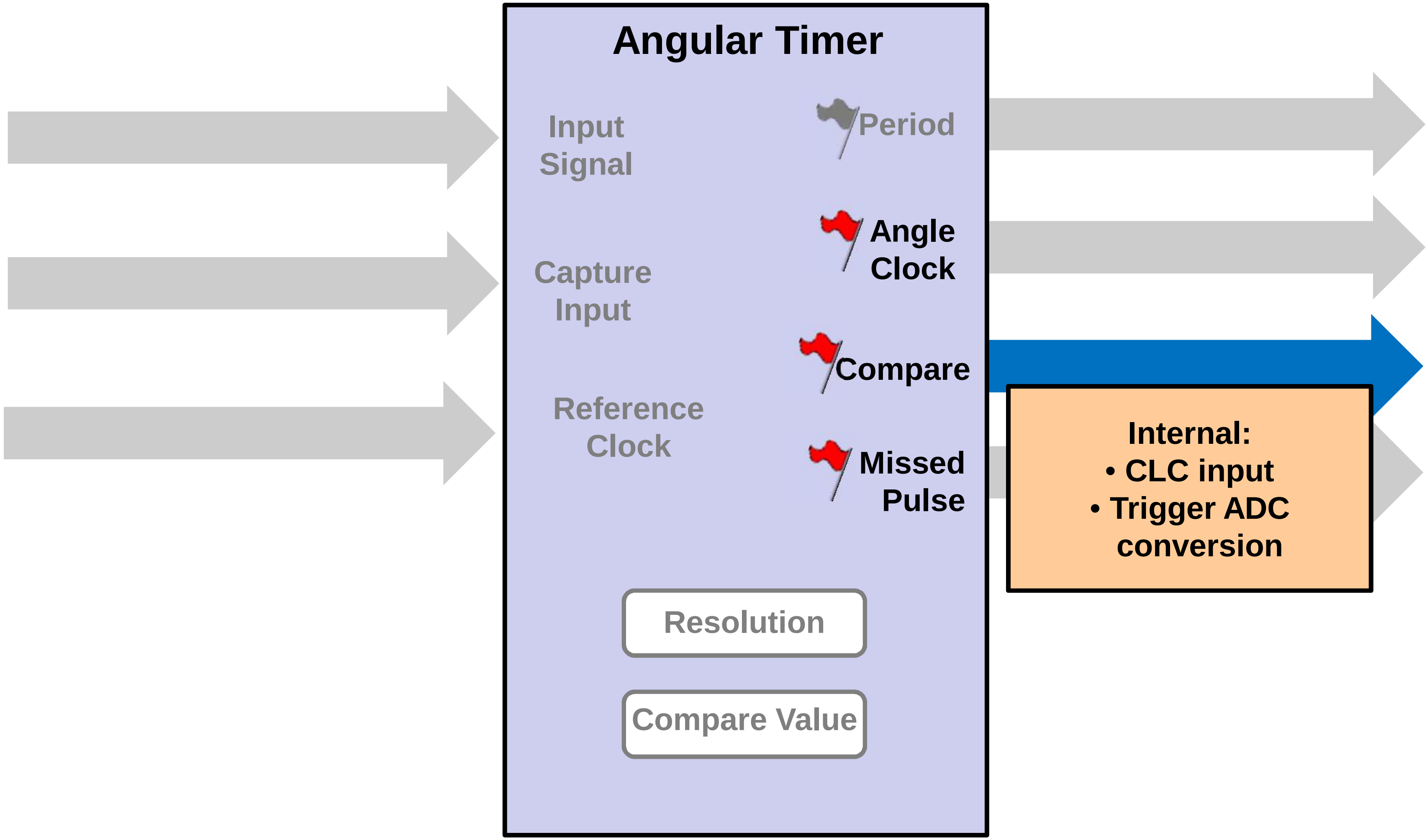
Angular Timer Simplified Block Diagram



Angular Timer Simplified Block Diagram

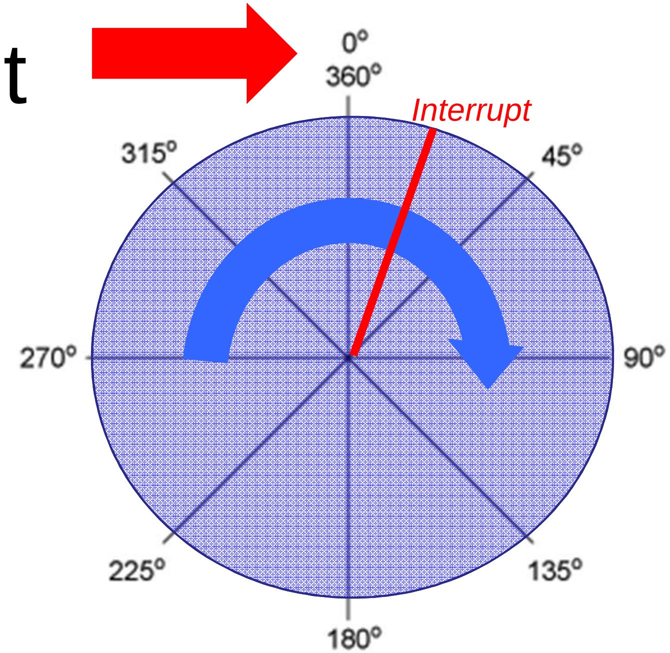


Angular Timer Simplified Block Diagram



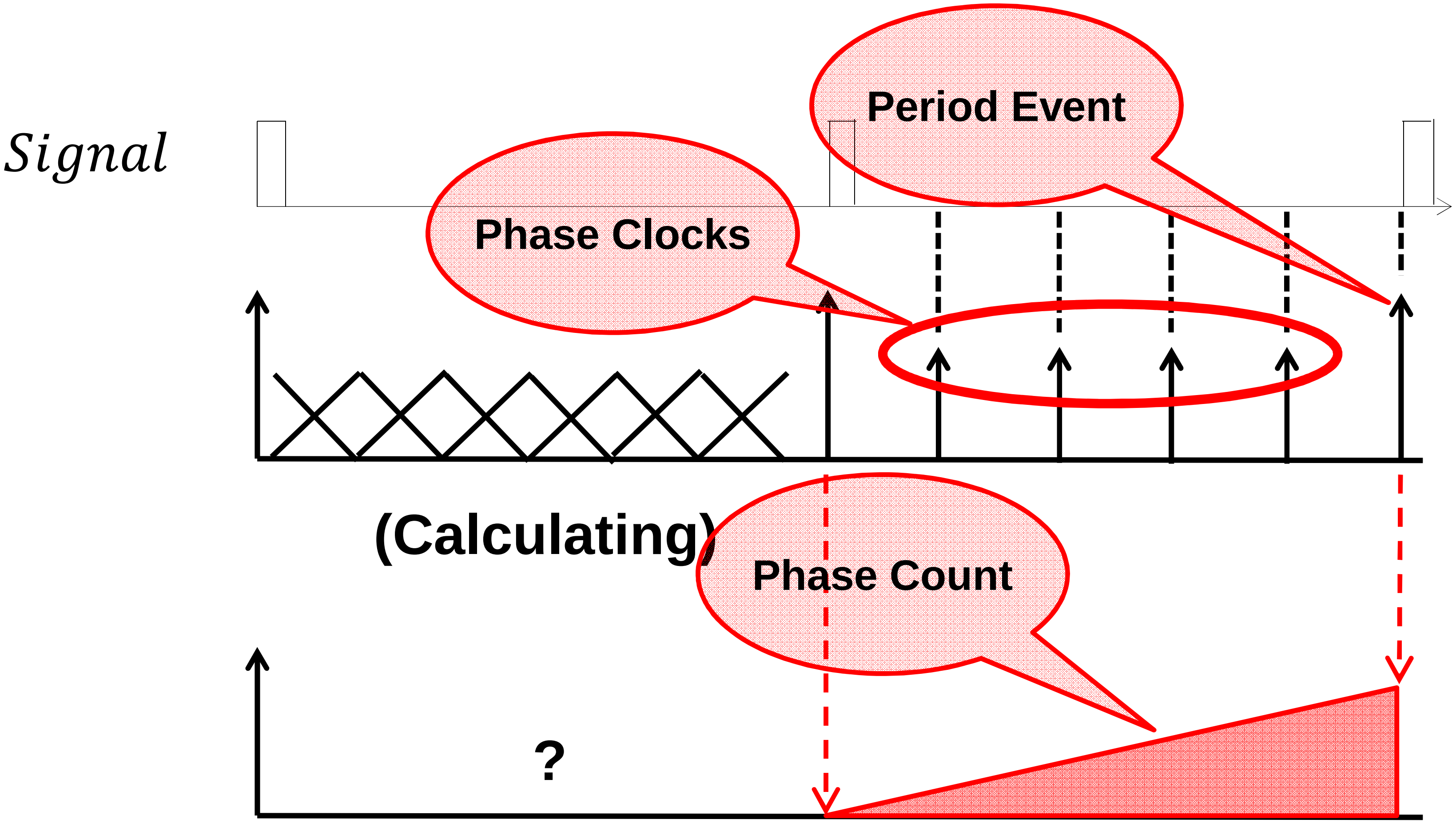
Angular Timer Operating Modes

- **Single Pulse**
  - Uses time between input pulses to determine 0°
- **Multi-Pulse**
  - Uses time between missing pulse in input stream



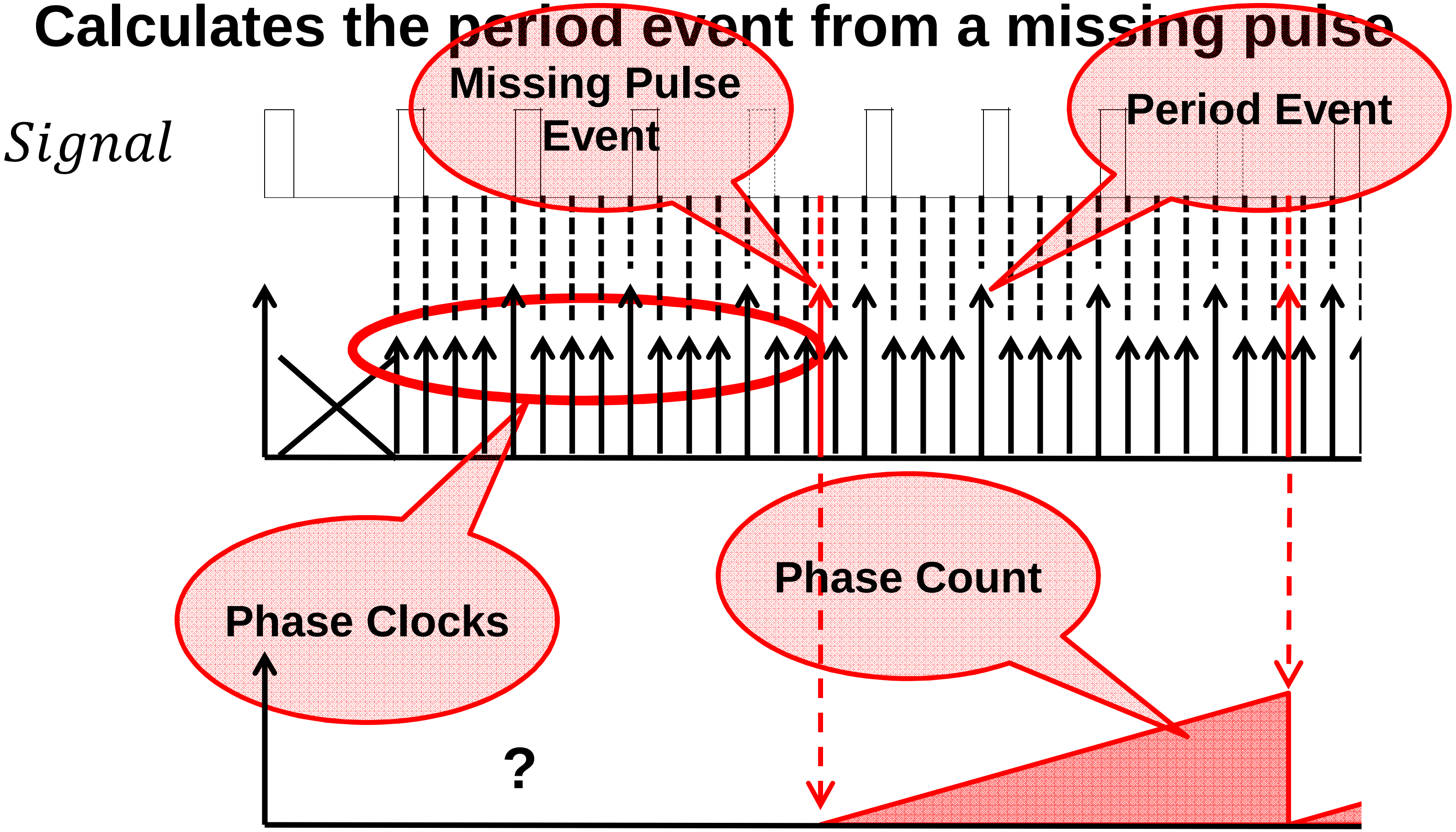
Single Pulse Mode

- Computes the phase clock from 1 pulse per period

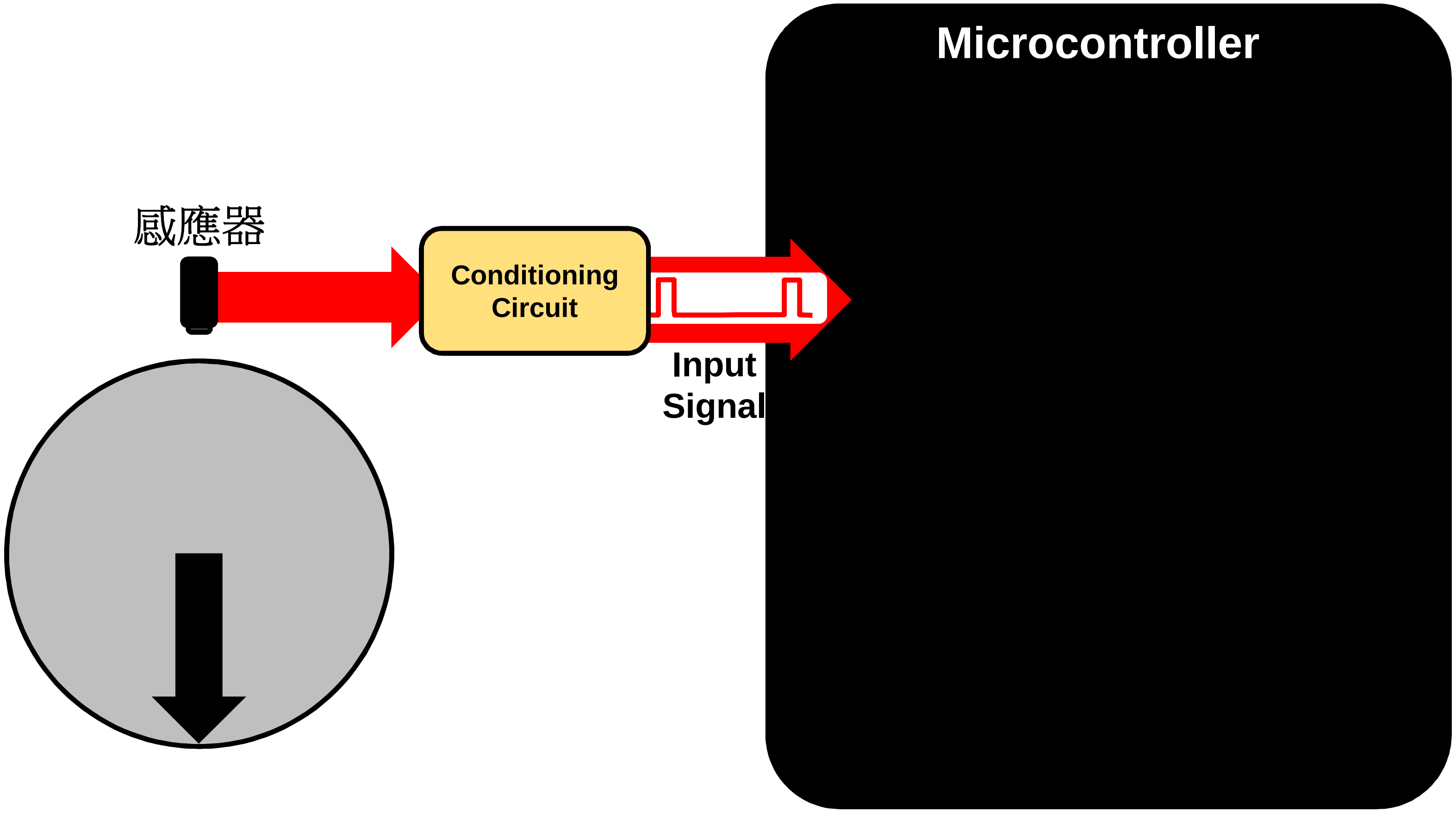


Multi Pulse Mode

- Computes the phase clock from 1 pulse per period
- Calculates the period event from a missing pulse

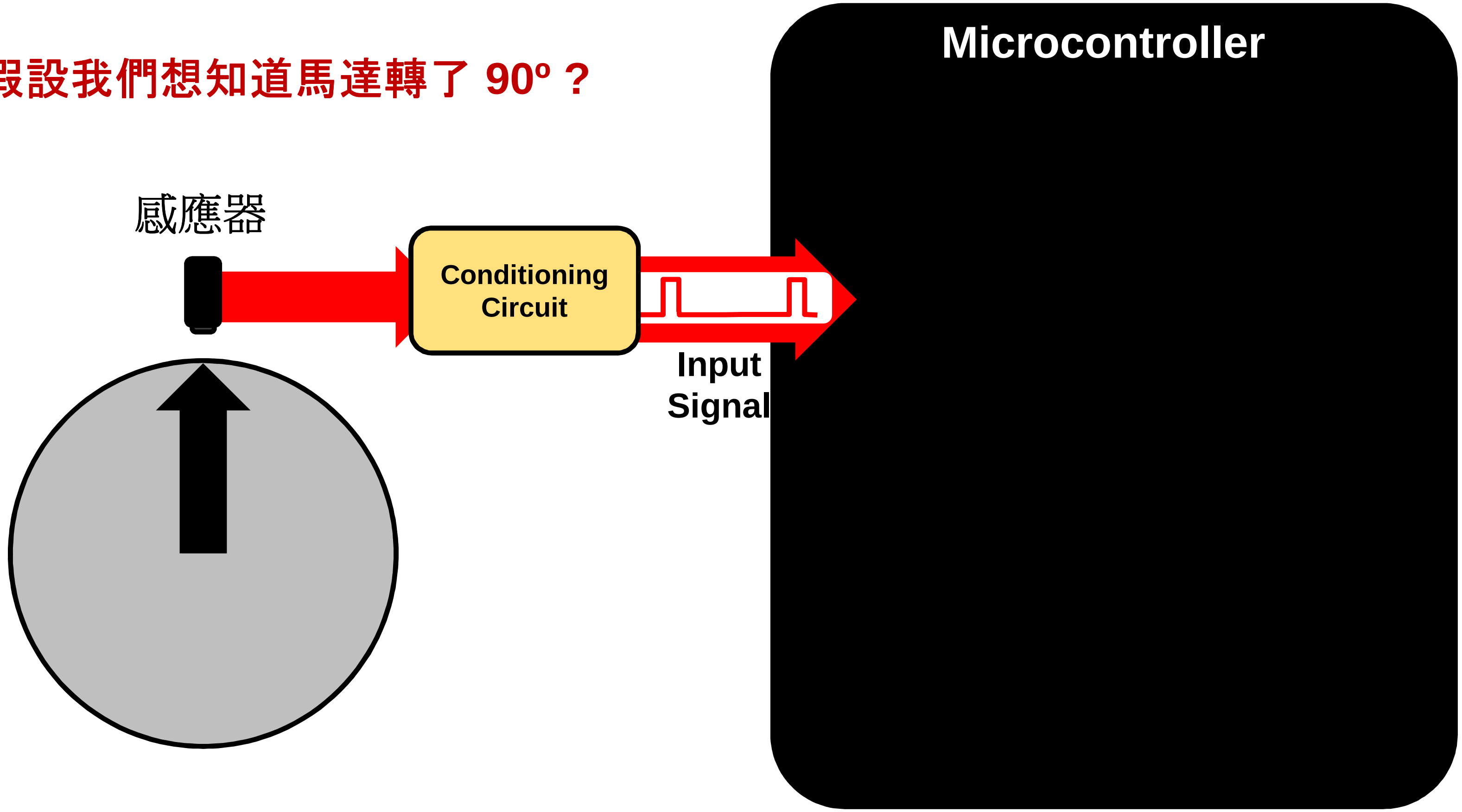


相位比較
(一般模式)



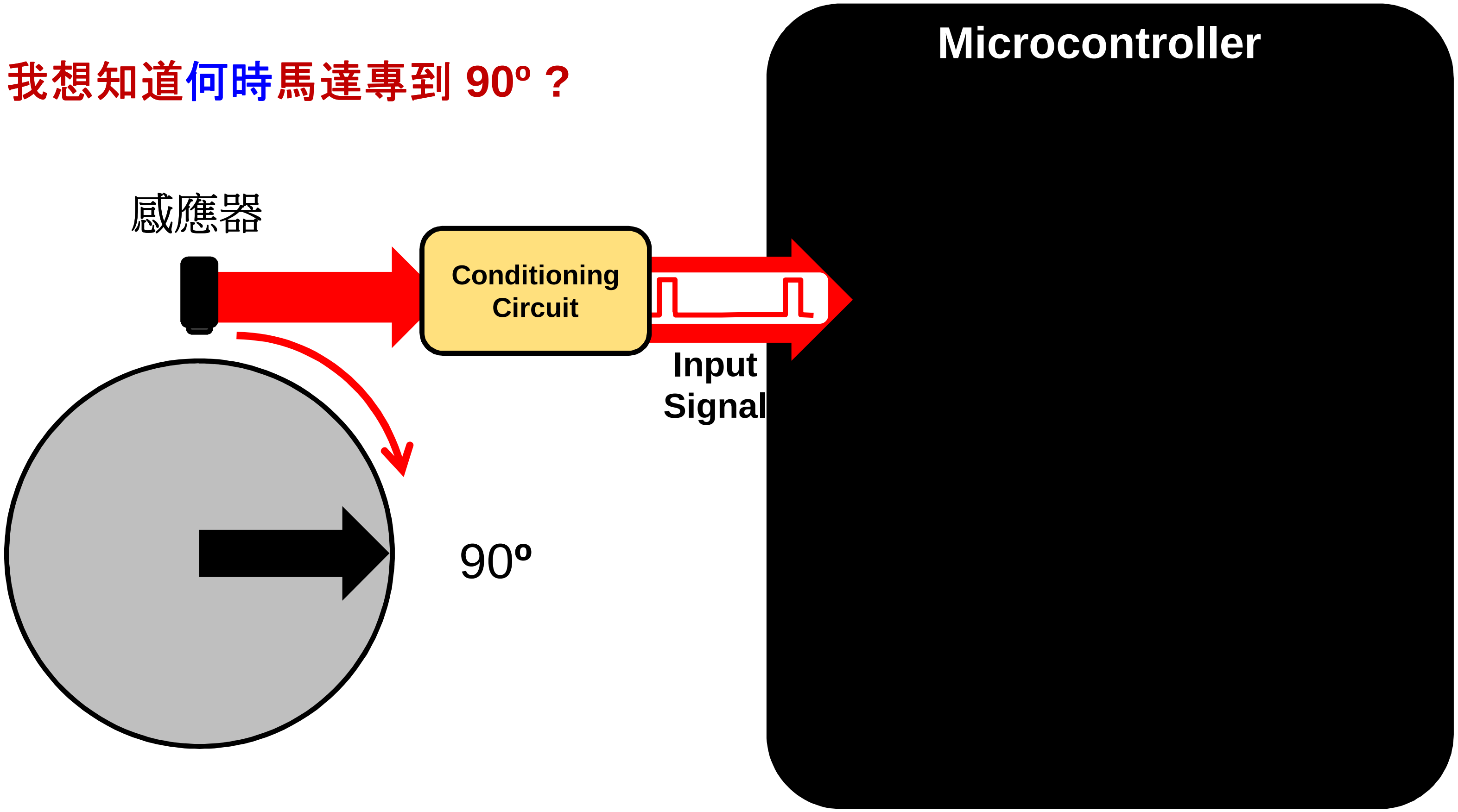
相位比較
(一般模式)

假設我們想知道馬達轉了 90º？



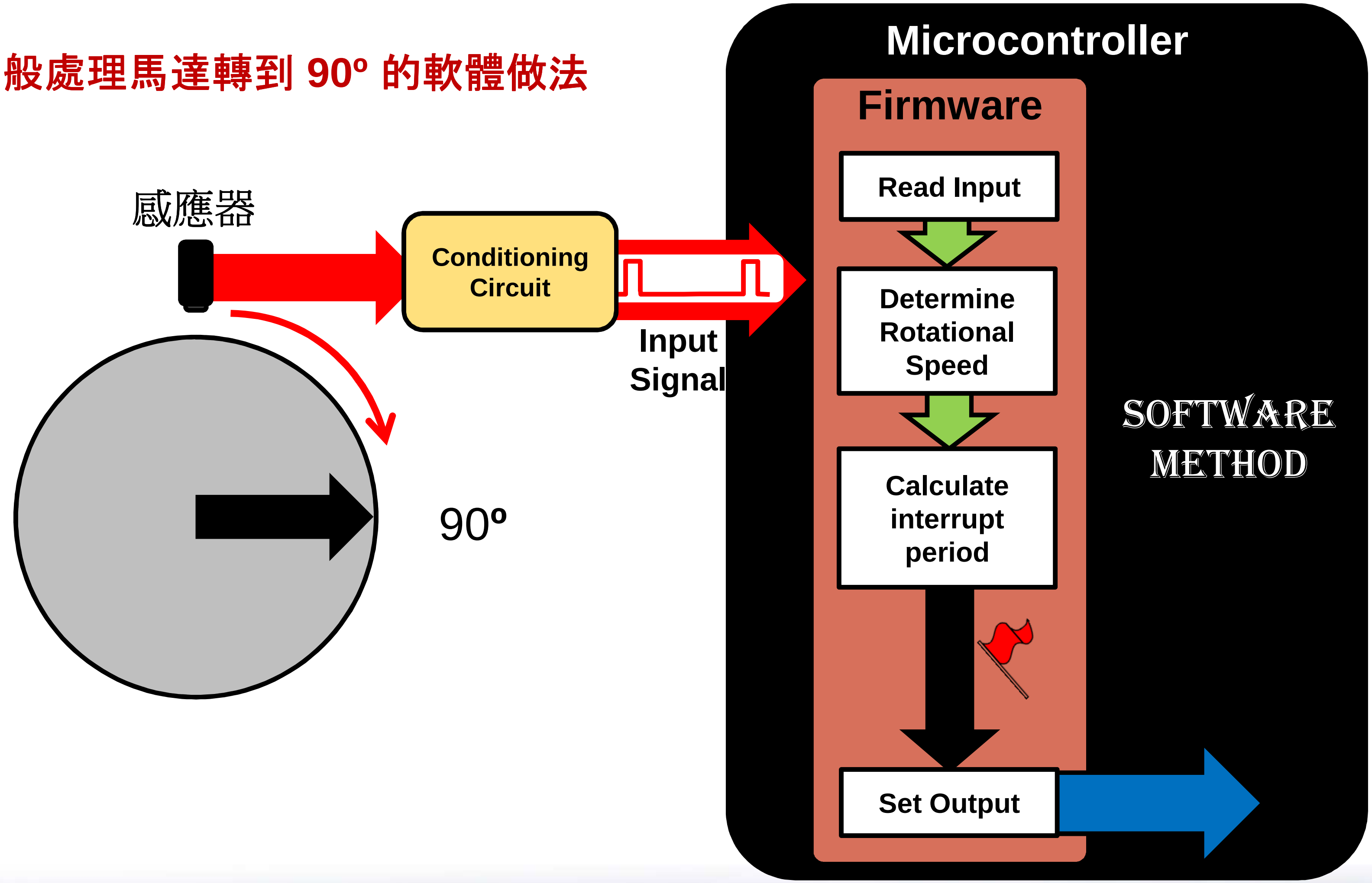
相位比較
(一般模式)

我想知道何時馬達專到 90º？

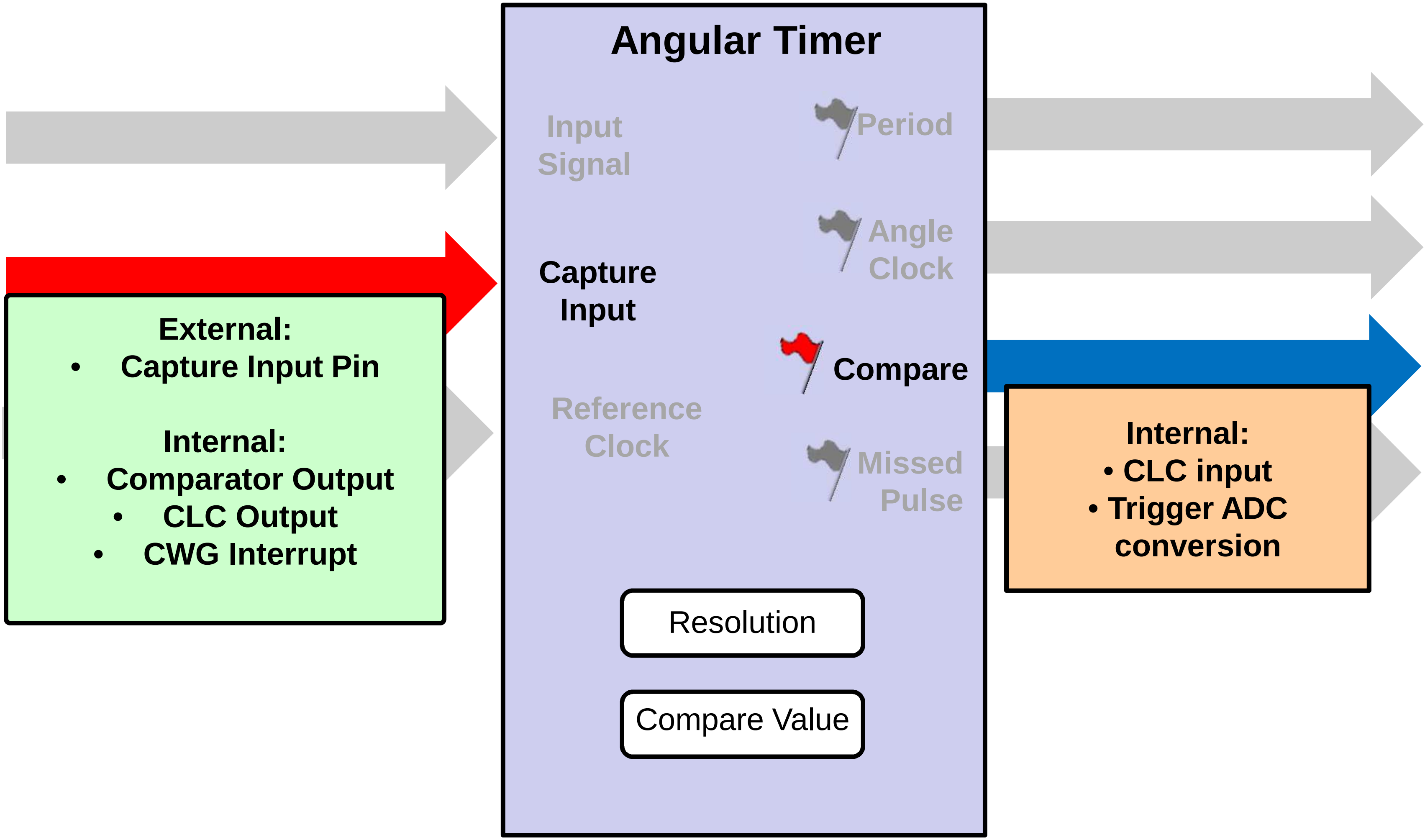


相位比較
(一般模式)

一般處理馬達轉到 90º 的軟體做法

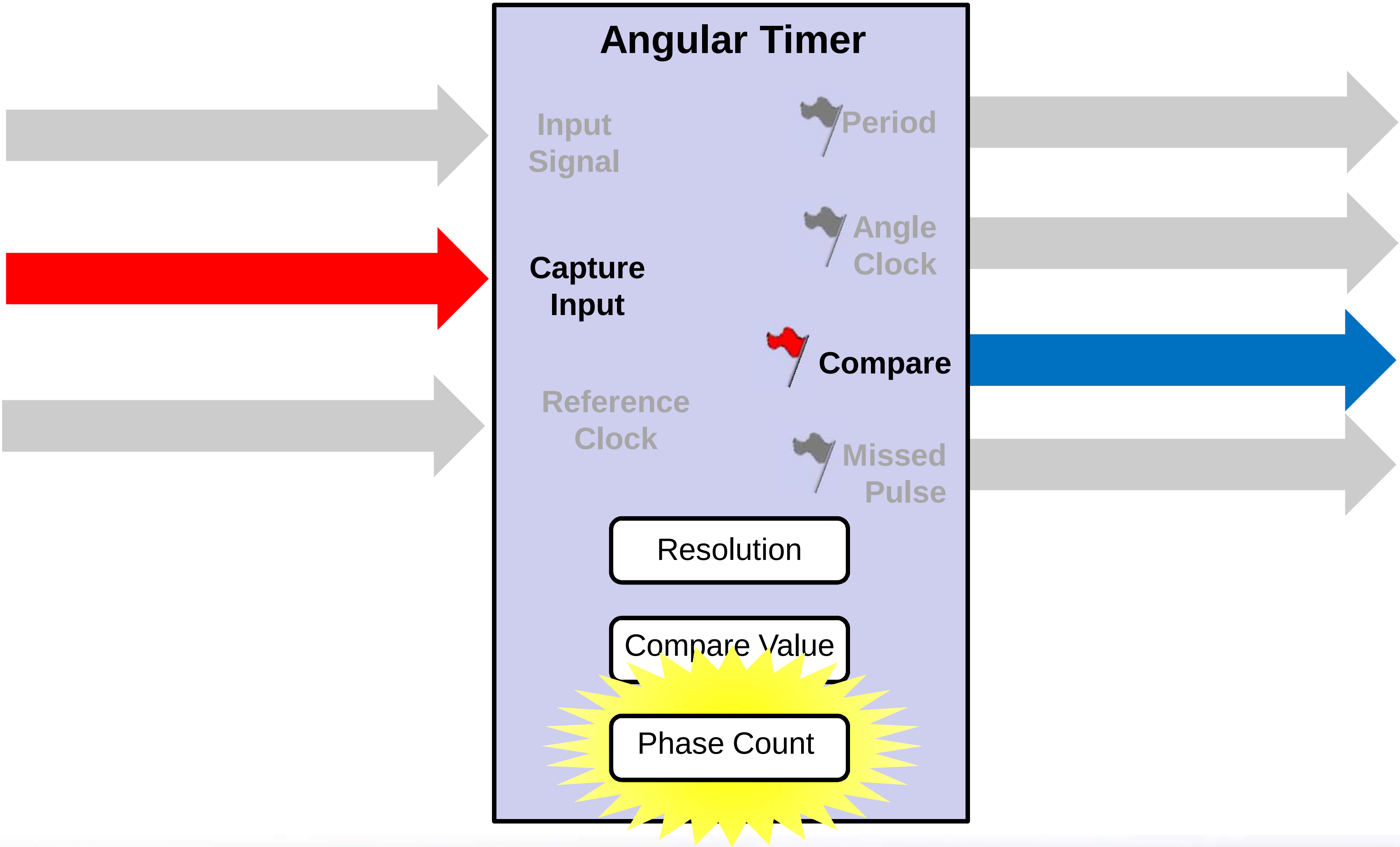


使用 AT 的方法



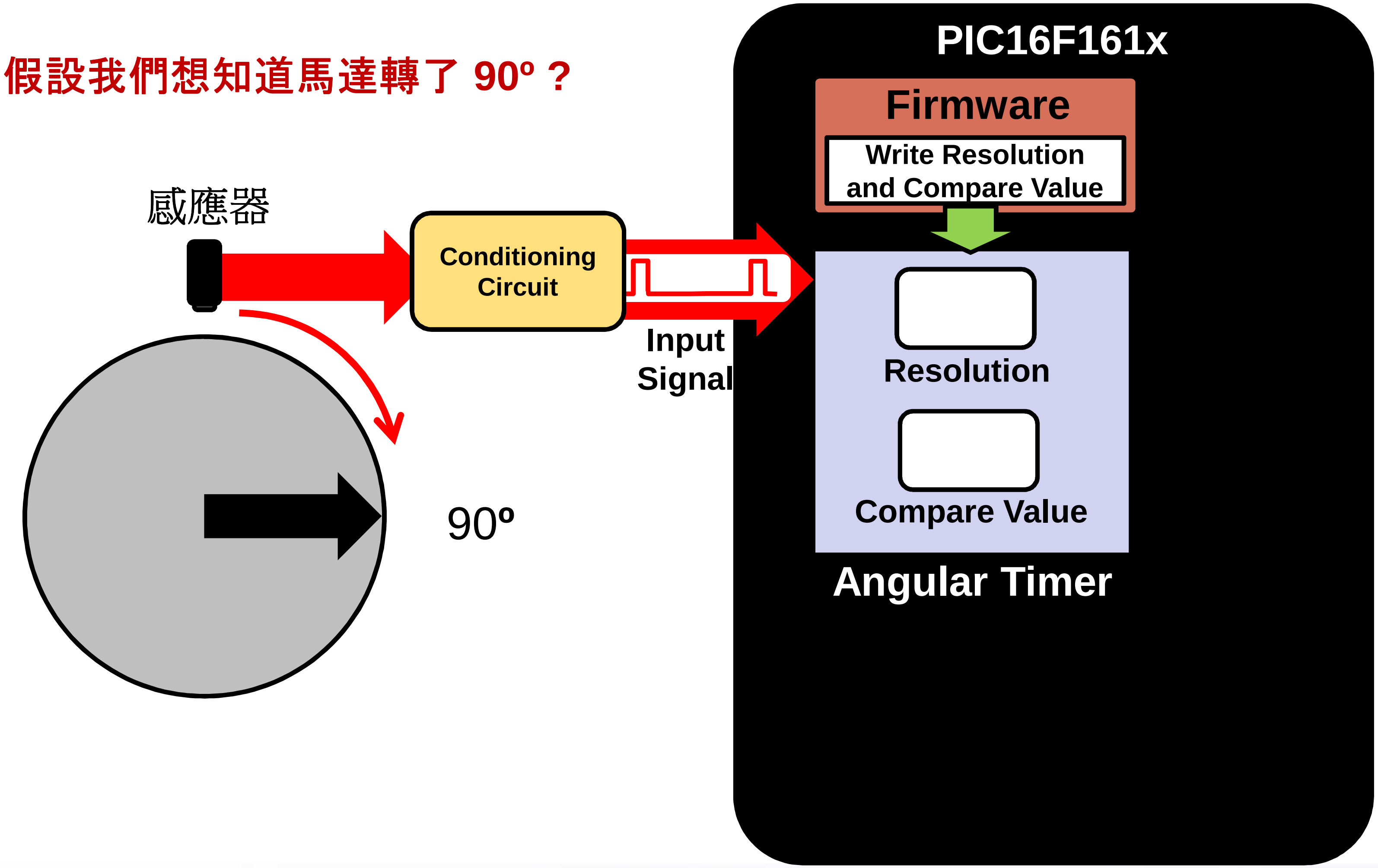
Angular Timer
Input Signal
Capture Input
Reference Clock
Period
Angle Clock
Compare
Missed Pulse
Resolution
Compare Value
External:
• Capture Input Pin
Internal:
• Comparator Output
• CLC Output
• CWG Interrupt
Internal:
• CLC input
• Trigger ADC conversion

使用 AT 的方法

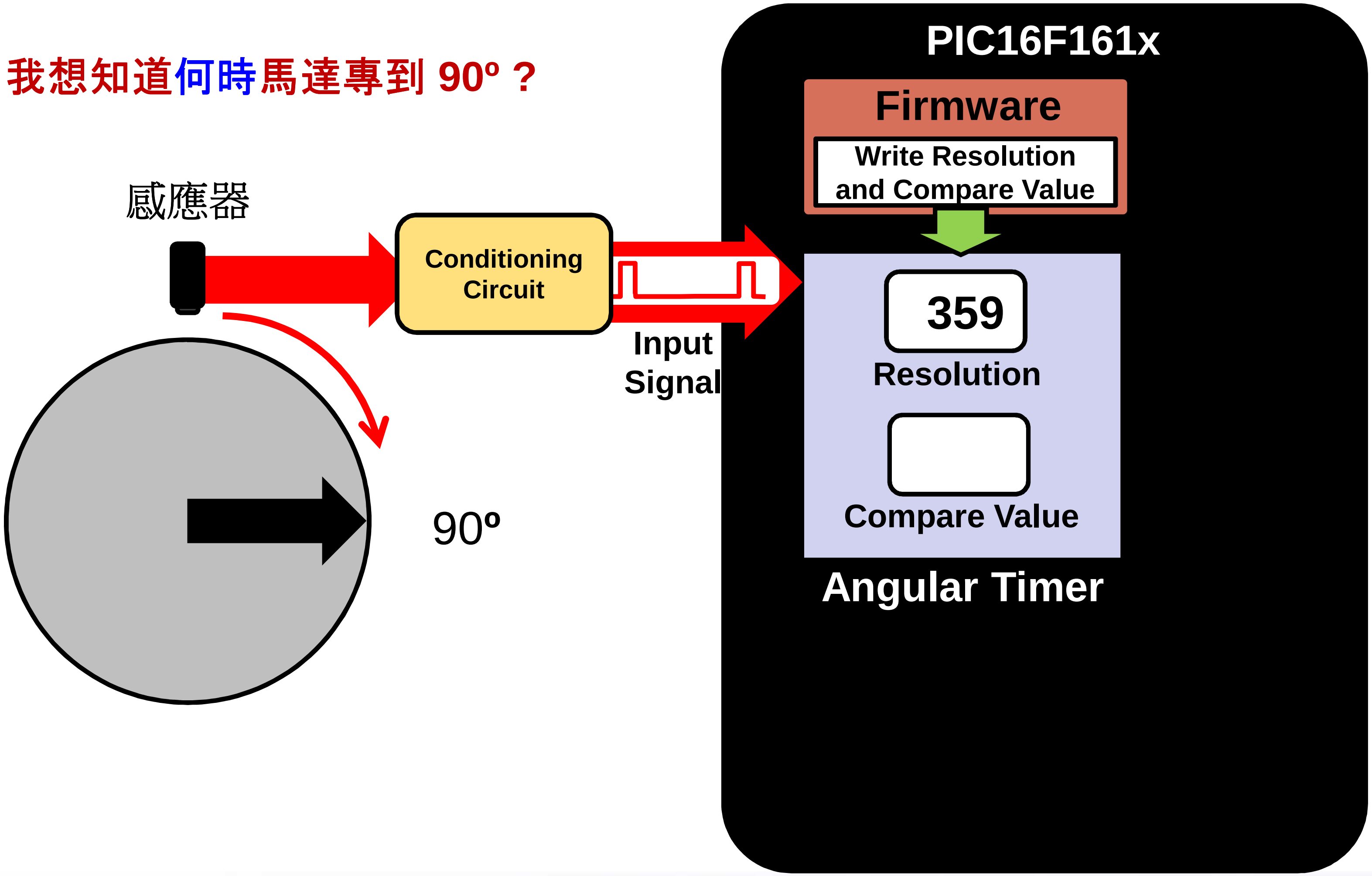

Angular Timer
Input Signal
Capture Input
Reference Clock
Period
Angle Clock
Compare
Missed Pulse
Resolution
Compare Value
Phase Count

使用 AT 的方法

假設我們想知道馬達轉了 90º？



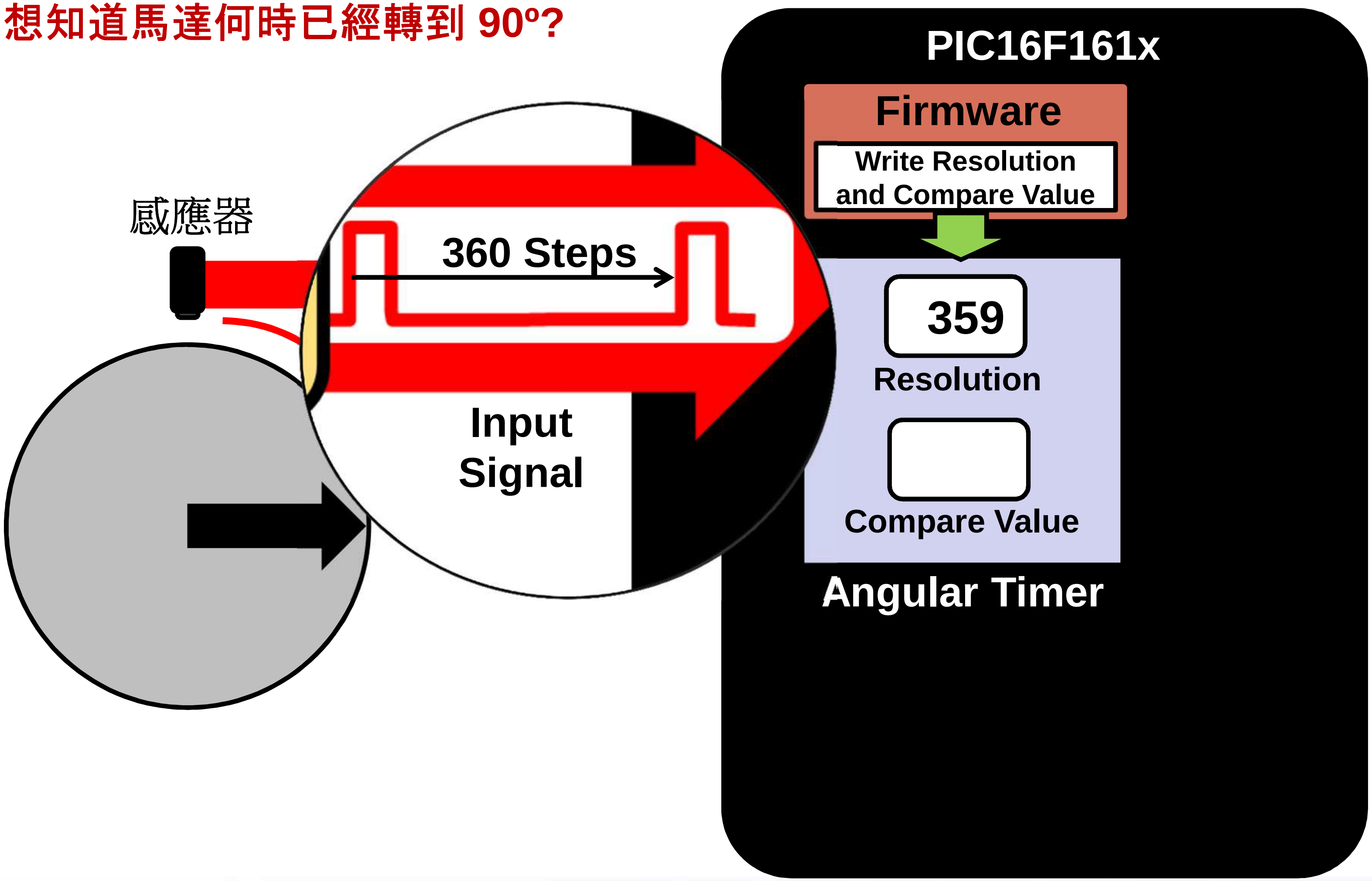
使用 AT 的方法



我想知道何時馬達專到 90º ?
感應器
Conditioning Circuit
Input Signal
90º
PIC16F161x
Firmware
Write Resolution and Compare Value
359
Resolution
Compare Value
Angular Timer

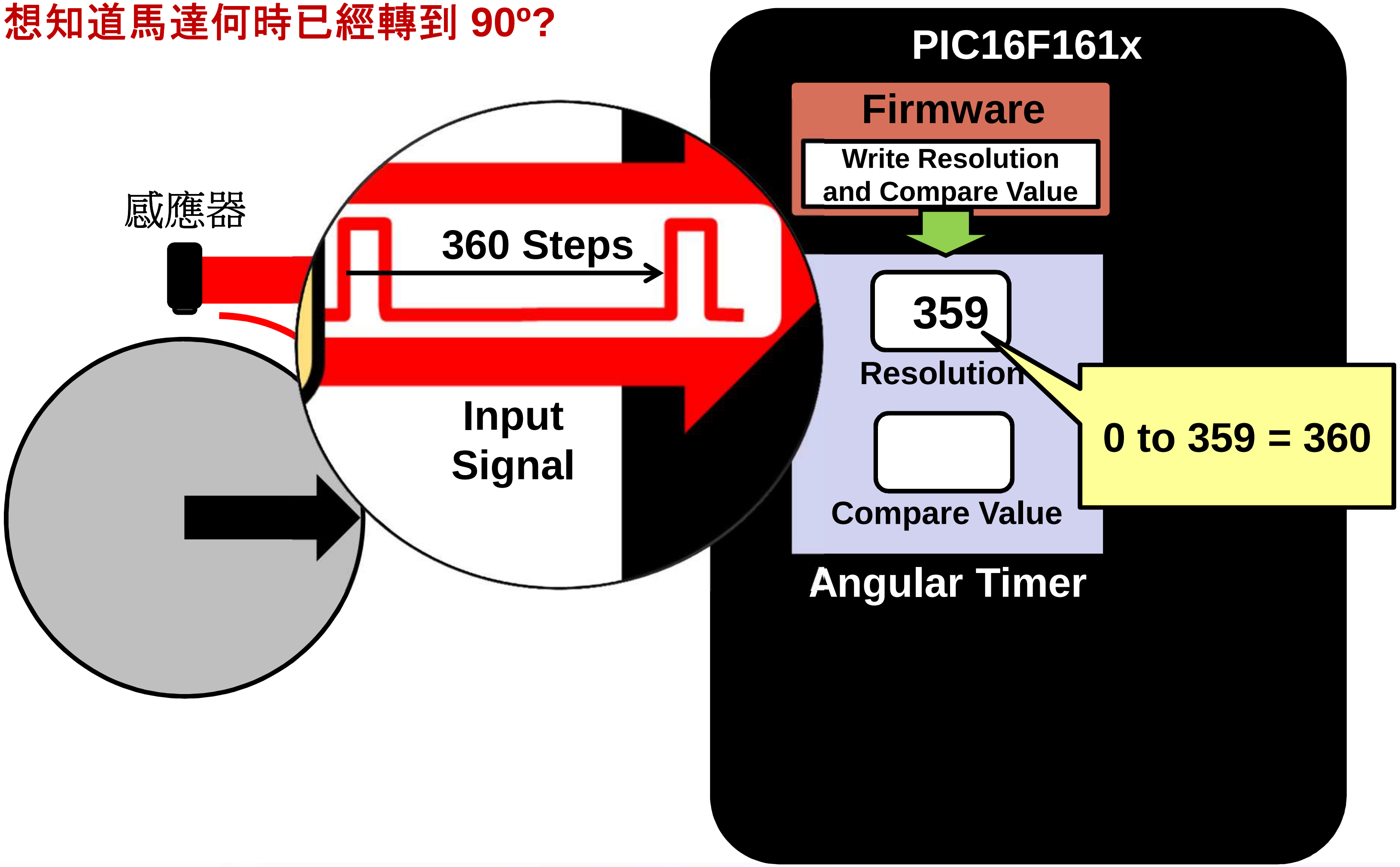
使用 AT 的方法

我想知道馬達何時已經轉到 90º?



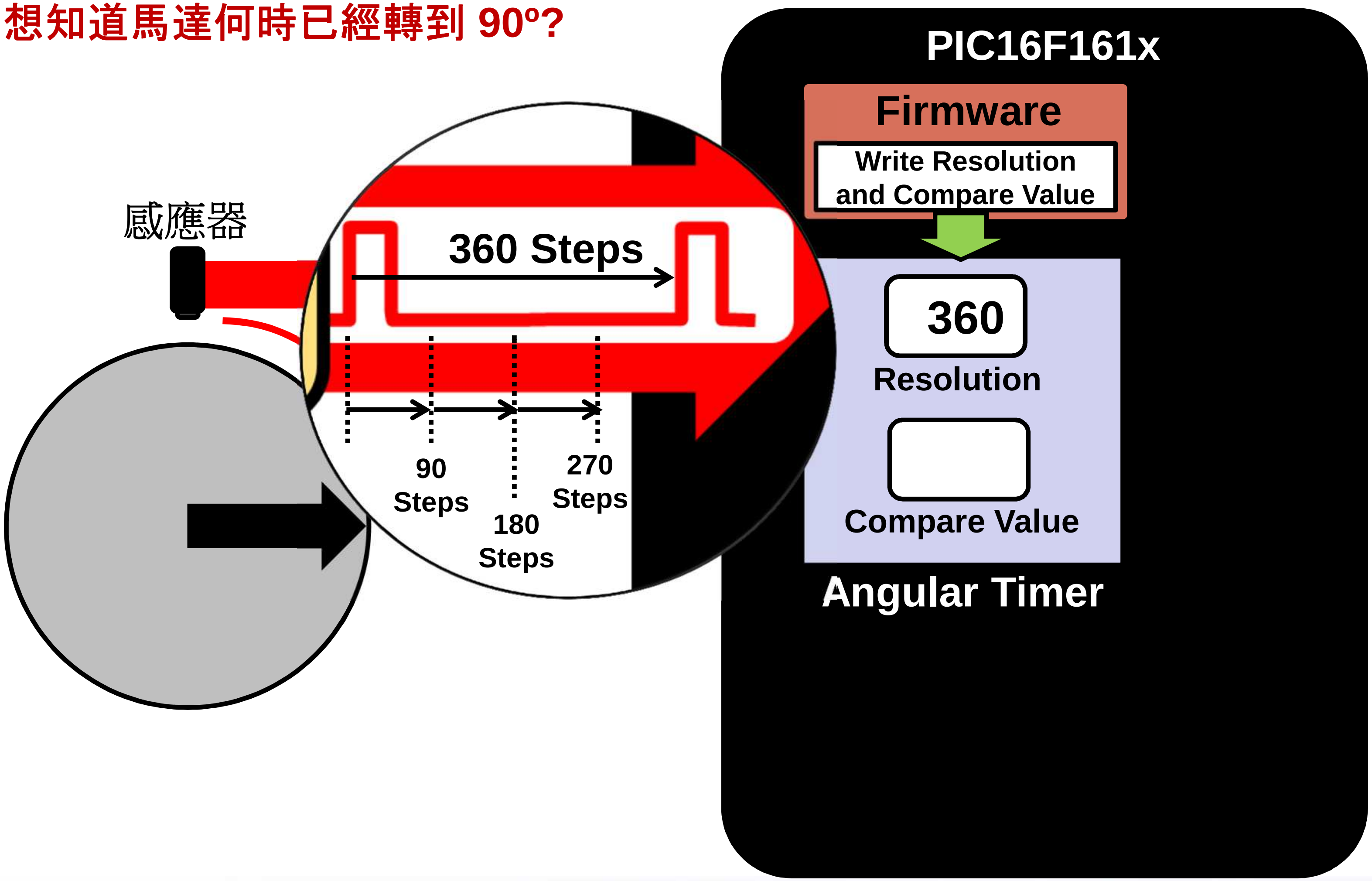
使用 AT 的方法

我想知道馬達何時已經轉到 90º?



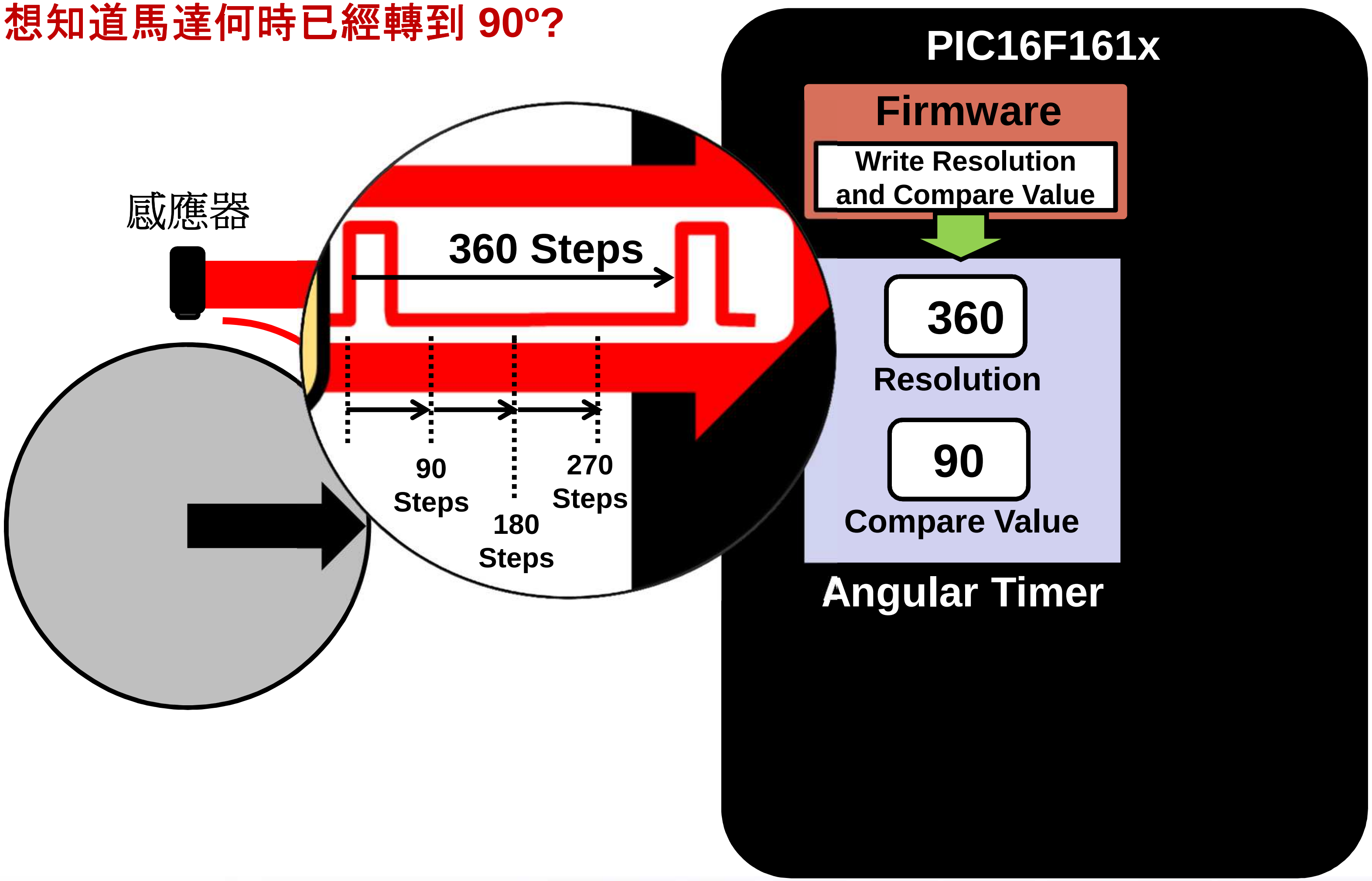
單脈衝模式

我想知道馬達何時已經轉到 90º?



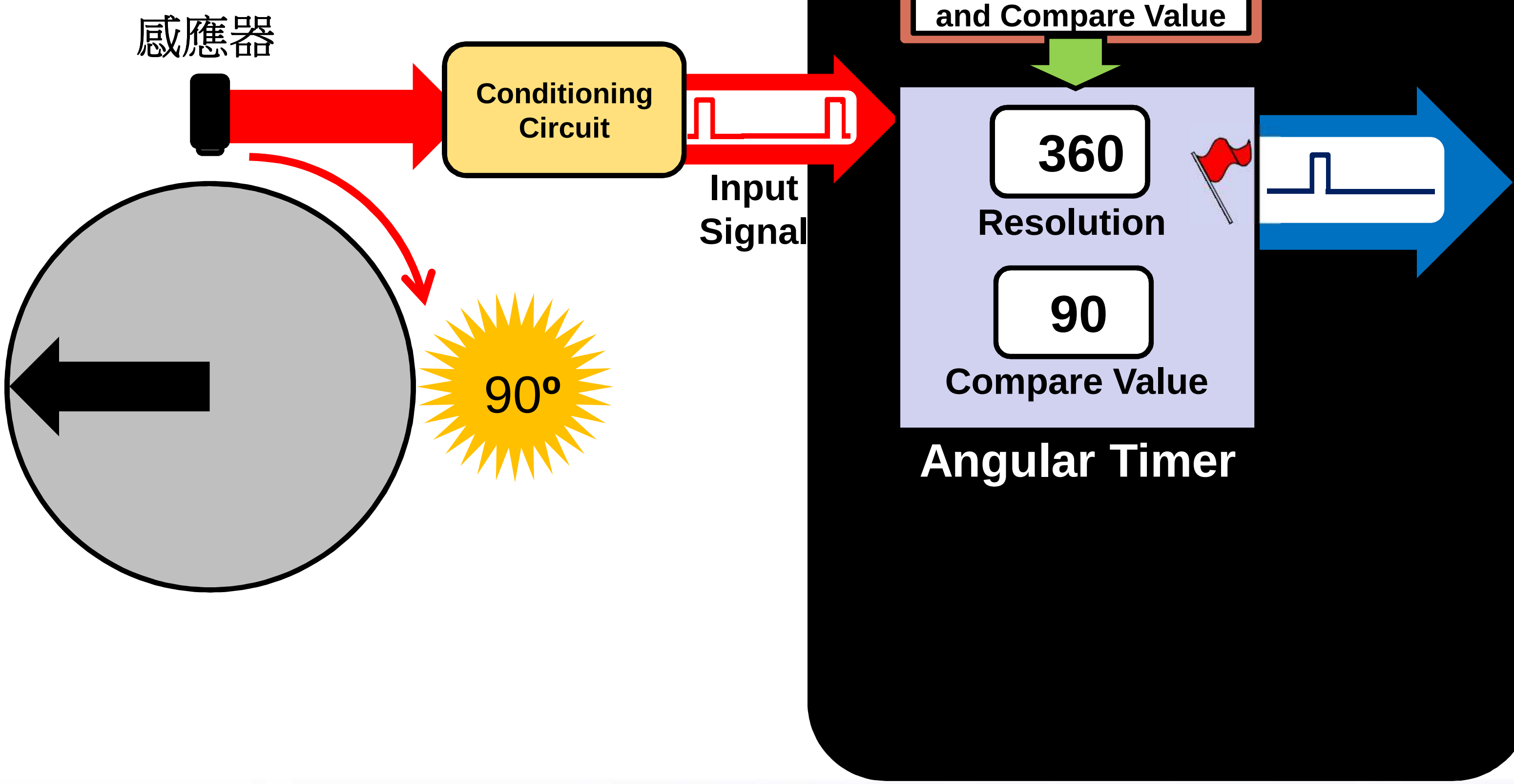
相位比較

我想知道馬達何時已經轉到 90º?

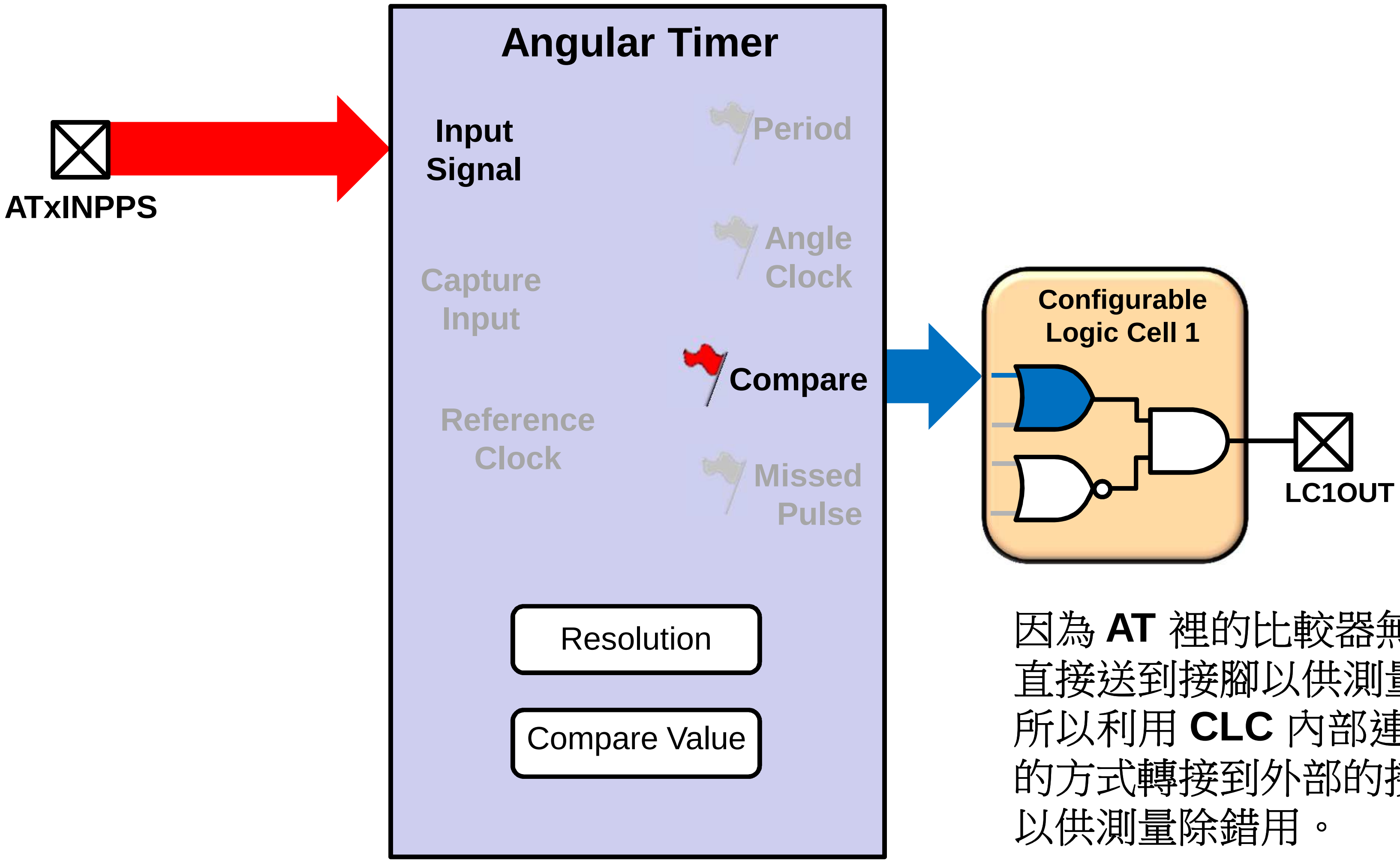


相位比較

我想知道馬達何時已經轉到 90º?



透過 CLC 觀測比較的輸出



因為 **AT** 裡的比較器無法直接送到接腳以供測量，所以利用 **CLC** 內部連線的方式轉接到外部的接腳以供測量除錯用。

Compare / Capture Mode

- **3** 個獨立的 比較**/**補捉暫存器 **(ATxCCy)**

  - 比較模式**:**
    - 比較 ATxPHS （全相位）to *ATxCCy(角度)*
    - **Trigger an ADC conversion on compare match**
  - 捕捉模式**:**
    - Initiate interrupt and latches value of ATxPHS whenever external signal occurs (rising/falling)

Real World Applications

- CDI
- TRIAC Dimming
- UPS Sine Synth
- RMS Measurement
- ANYTHING PERIODIC

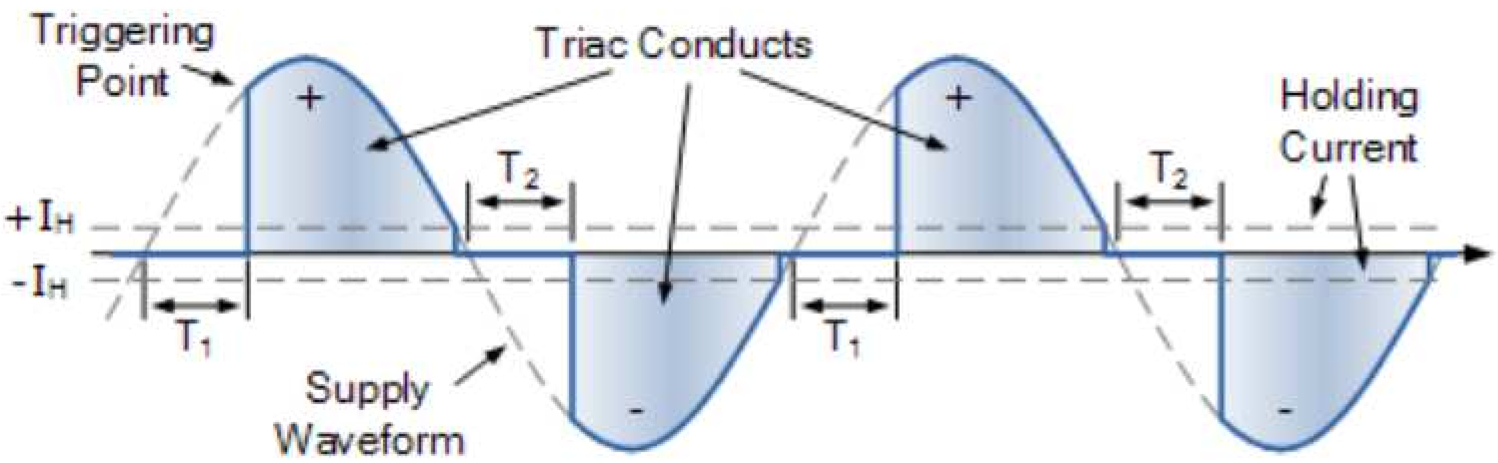
Lab 7

- 利用 ZCD 偵測上升弦波交越點
- 將 ZCD 上升緣做為 Angular Timer 的 0^{O} 起點直到下一上升緣的到來
- 設定兩比較器的比較時間 (C1 < C2)
- 將兩比較器輸出送至 CLC 內的 RS 正反器做角度(相位)的輸出

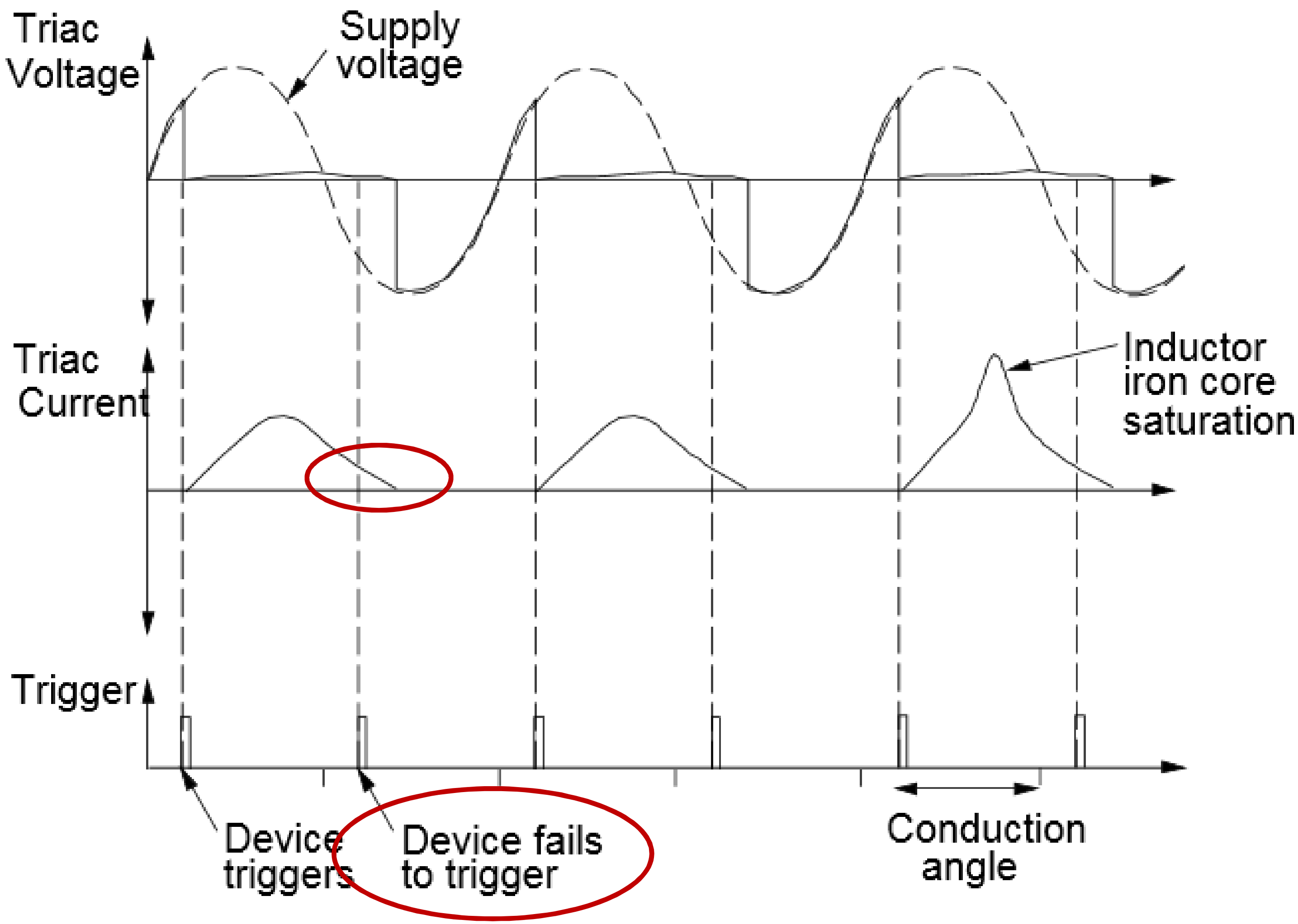
實驗手冊第 **48** 頁

TRIAC Firing

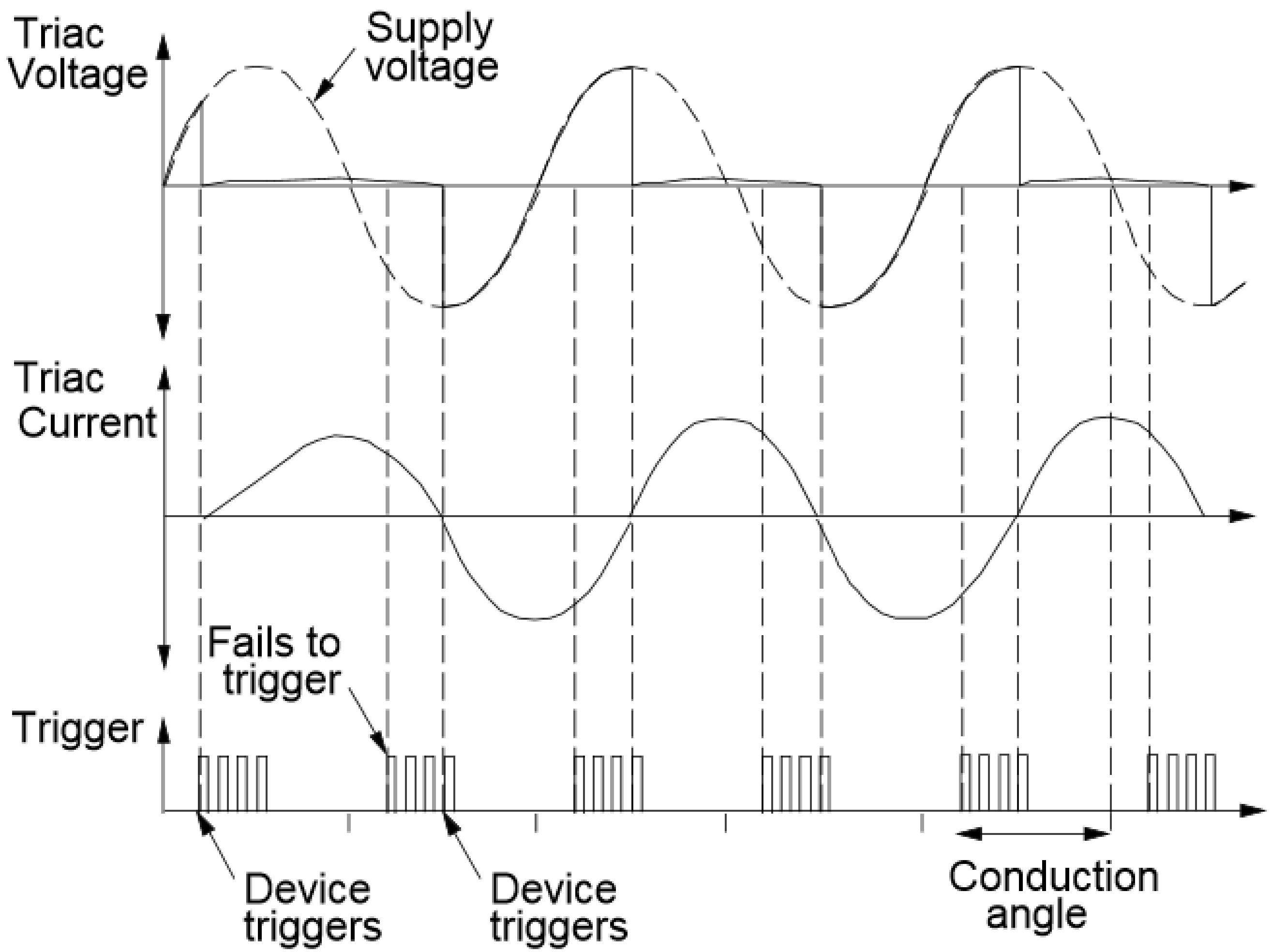
- 連續點火的觸發
  - 在一個固定角度產生五個 16uS 的脈衝區觸發 TRIAC



TRIAC Firing Pulse – 電感性負載



TRIAC Firing Train



Lab 8

- 修改前面的 Lab7 加入對 TRIAC 觸發的應用
- 將 ZCD 的上、下緣偵測做為 HLT/TMR2 的時間框
- 在時間框內做 5 個 PWM 脈衝的輸出

實驗手冊第 **54** 頁

Lab 8-1

延續 Lab8 的設定，加入 ADC 可變電阻的輸入，藉此 AD 的轉換值來調整TRIAC 觸發角度從 10 度 ~ 170 度及負半週的 190 度 ~ 350 度。

實驗手冊第 **58**頁

Regional Training Centers

Math Accelerator

附錄 A

Regional Training Centers

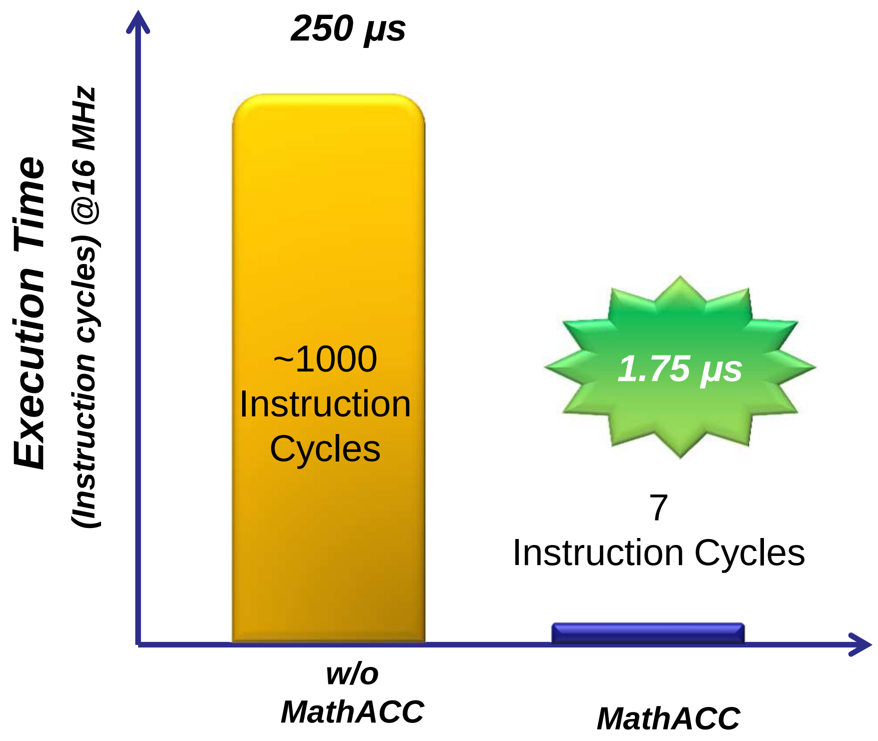
Math Accelerator (MathACC)

16-bit results in an 8-bit machine

Math Accelerator (MathACC)

- **Hardware Calculation Unit to reduce computation time of common mathematical functions**
- **Decreases PID calculation time by a factor of 150x on a PIC16**
- **Decreases multiplication time by a factor of 75x on a PIC16**

PID Loop
Calculation Time



Math Accelerator (MathACC)

16-bit math in an 8-bit machine

Benefits

- **Better performance in applications requiring closed-loop control**
  - Much faster control loop bandwidth
  - Reduces code size of closed-control calculation
  - Better interrupt performance
- **Flexibility to allow for multiple calculations to be run in parallel**
  - Ability to context switch between multiple PID loops/math algorithms
- **Saves power as calculations are completed almost instantly**
  - Run PIC® MCU at lower speed or enter SLEEP more often

Math Accelerator (MathACC)

16-bit math in an 8-bit machine

Possible Application Uses

- **Motor control**
  - Brushed DC (BDC)
  - Brushless DC motors (BLDC)
- **Digital Filters**
  - Multiply and Accumulate mode can perform a MAC in 4 instruction cycles
- **Routines which are multiply dominant**
  - **Taylor Series**
  - **Cordic**
  - **Division**
  - **FIR/FFT**

Microchip Regional Training Centers

Math Accelerator (MathACC)

16-bit math in an 8-bit machine

General Overview

- **Hardware math accelerator with two major modes:**
  - ACC += (A + B) * C
    - **Unsigned add/multiply without accumulation (**mode 000**)**
    - **Signed add/multiply with accumulation (**mode 001**)**
    - **Unsigned add/multiply without accumulation (**mode 010**)**
    - **Signed add/multiply with accumulation (**mode 011**)**
  - PID Calculation (mode 101)
    - **PID = Kp + Ki/s + Kd*s**
- **Mode control/switch by setting PIDxCONbits.MODE**

Add/Multiply Modes

Microchip Regional Training Centers

Code Example of doing a 16-bit multiply:

```
// initialize A, B and C
int16_t A = 0xFFFF;
int16_t B = 0;
int16_t C = 0xFFFF;

PID1SET = B;          // load B to PIDxIN
PID1K1 = C;           // load C to PIDxK1
PID1INH = A >> 8;     // load the higher byte of A to PIDxINH
PID1INL = A;          // load the lower byte of A to PIDxINL

while(PID1CONbits.BUSY);    // wait for the calculation to be done
```

- **Note: A write to PIDxINL triggers the calculation**

Add/Multiply Modes

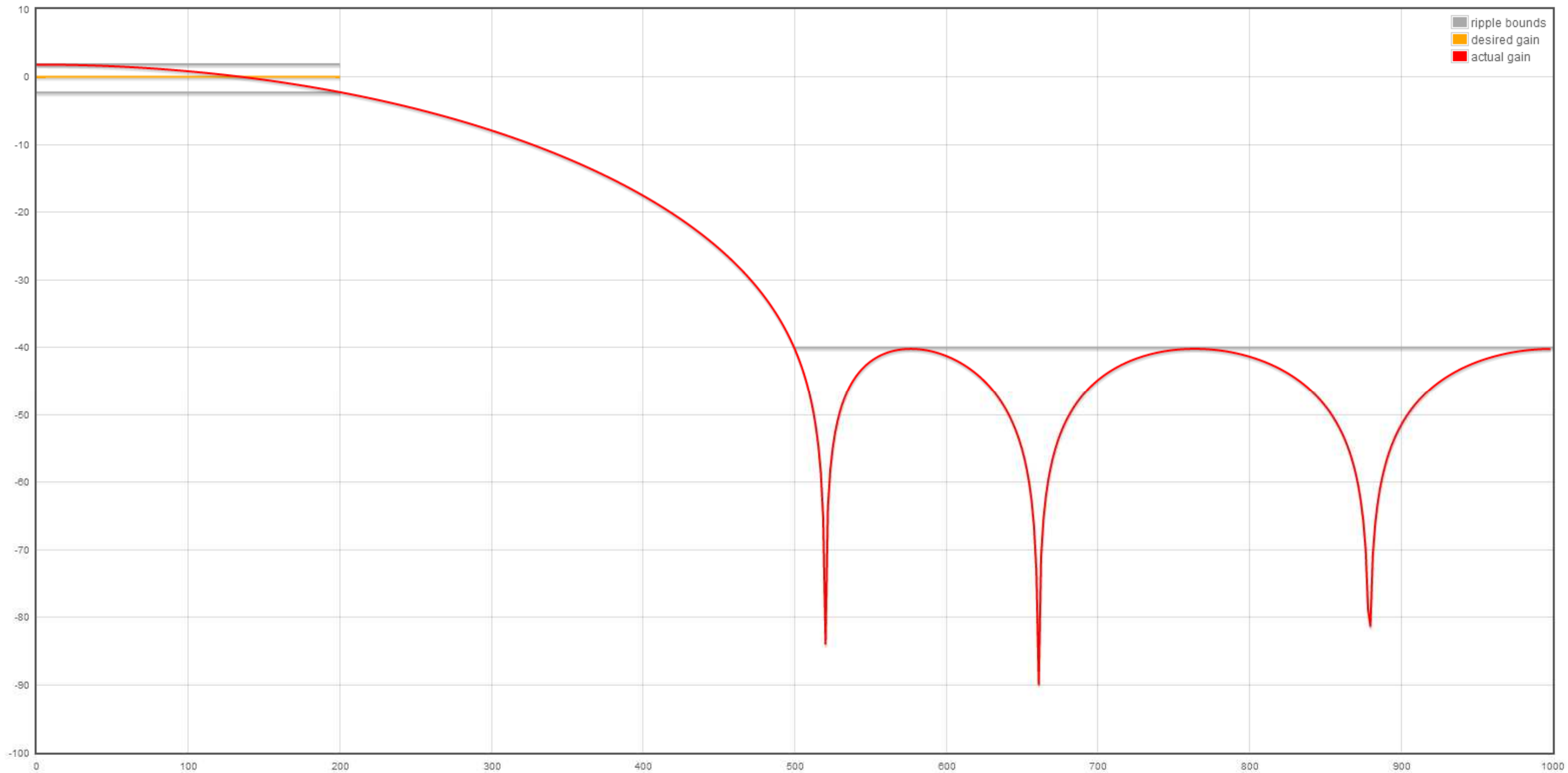
Benchmark:

- **Instruction cycles of doing a 16-bit multiply:**

Digital Filter Applications

Finite Impulse Response (FIR) Filter:

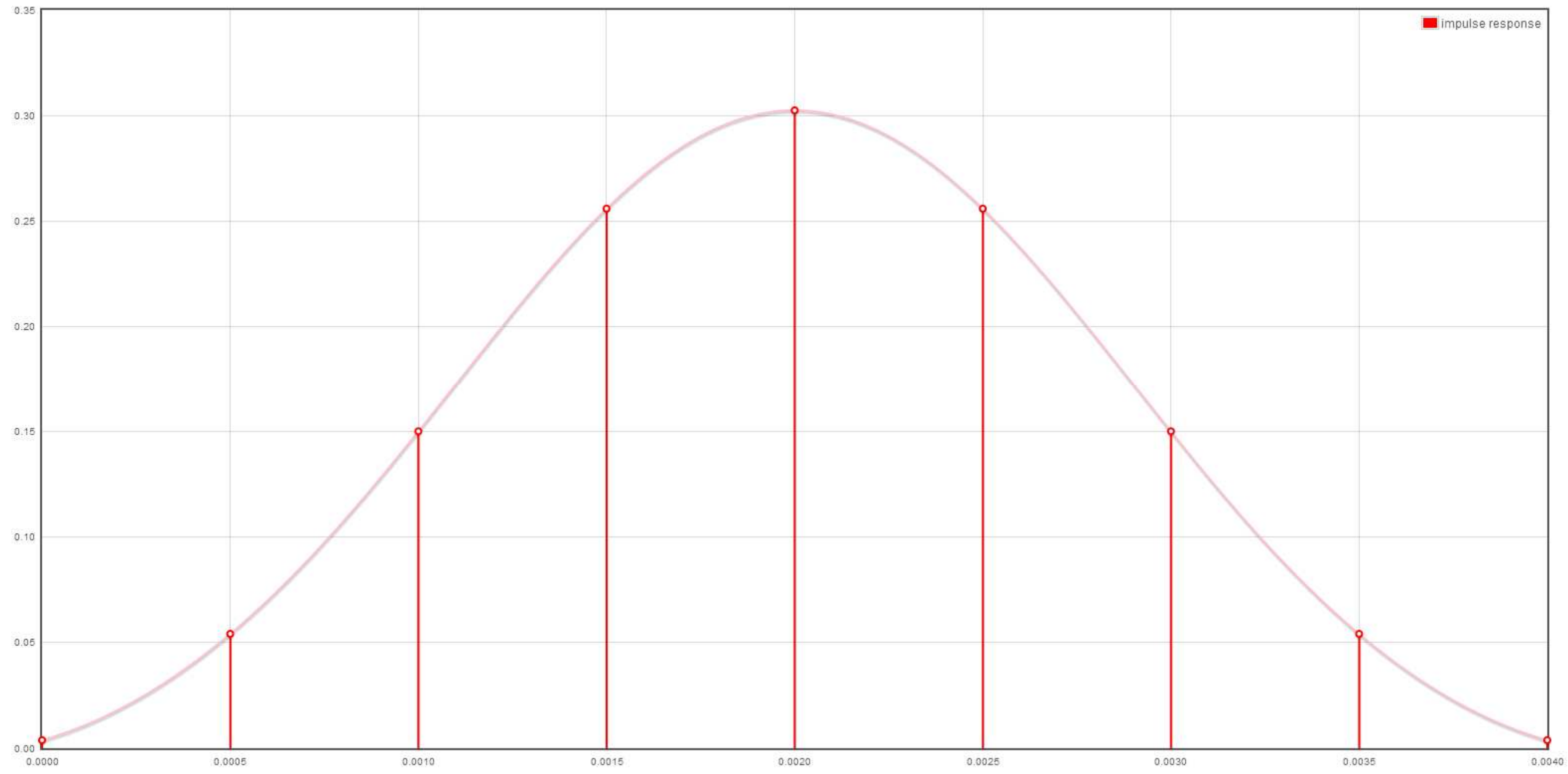
- **Impulse Response**
- **Frequency Response**



Digital Filter Applications

FIR Filter Design:

- $y[n] = b_0*x[n] + b_1*x[n-1] + \ldots + b_k*x[n-k]$ (ACC += A * B)



- **Too many multiplies and adds?**

- Why not use the MathACC!!!

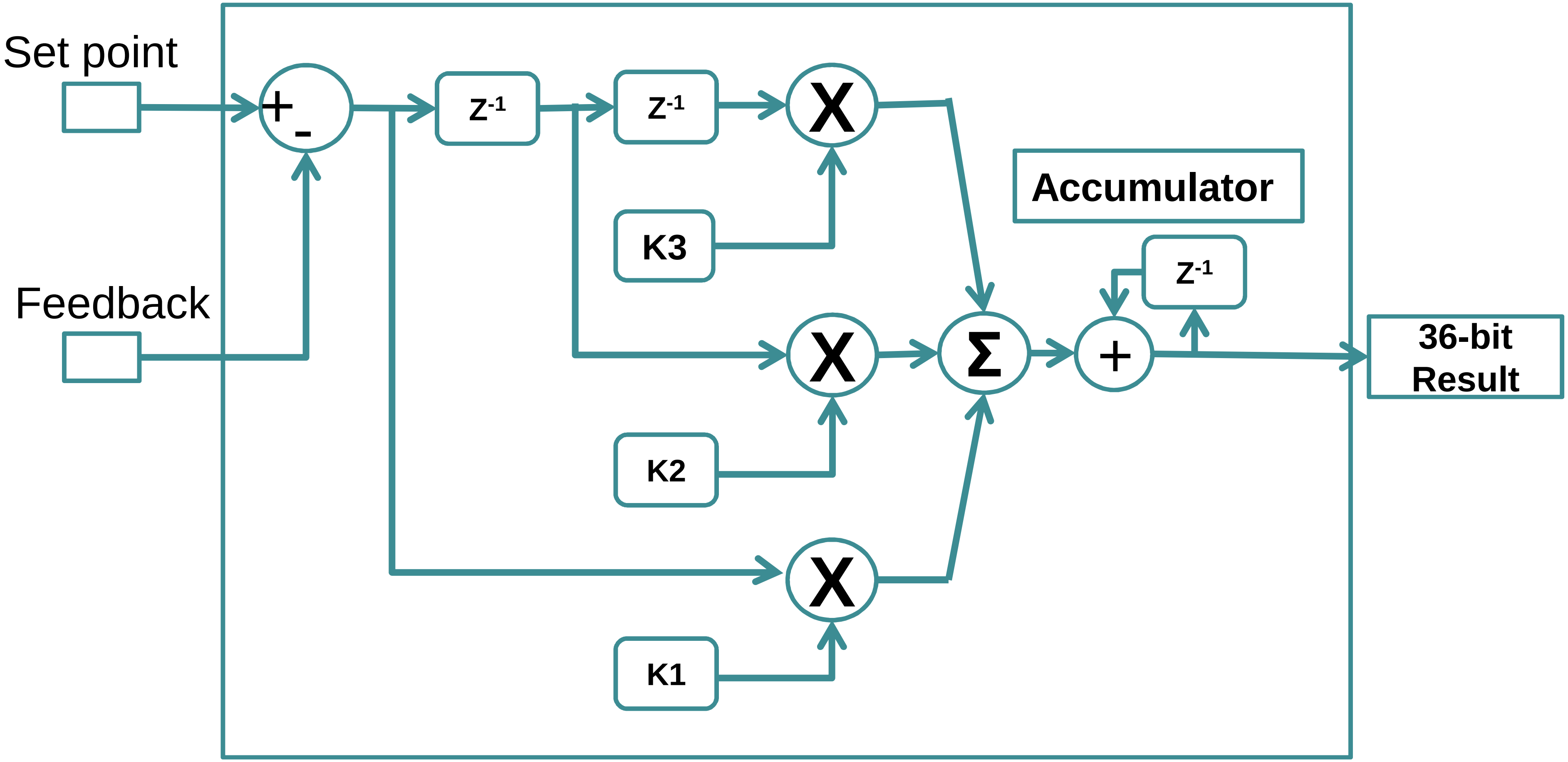
Digital Filter Applications

FIR Filter Benchmark:

- **Takes 100us to do 8 coefficients(8 multiplies + 7 adds) each time**
  - **RECALL: it takes over 200us to do 1 multiply in software!**
    - **Real data measured by BODE 100:**

PID Mode

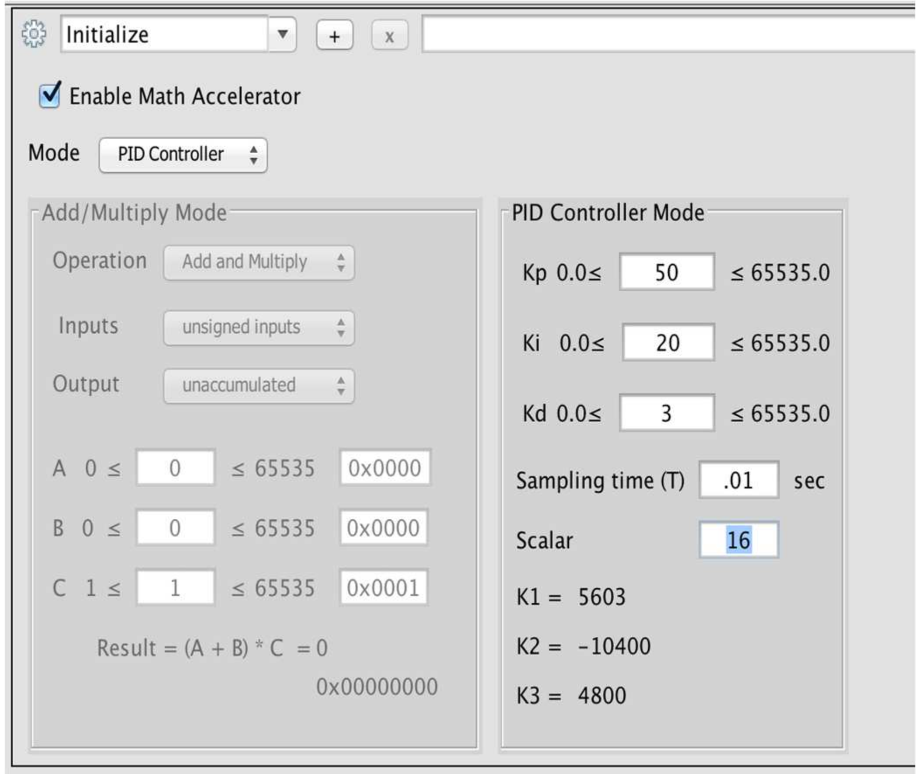
Constant Inputs: $K1 = Kp + Ki * T + \frac{Kd}{T}$ | $K2 = -(Kp + 2*\frac{Kd}{T})$ | $K3 = \frac{Kd}{T}$



PID Mode

PID mode MCC setup:

- The user only needs to determine Kp, Ki, Kd and T!



Note: K1, K2 and K3 will be rounded by MCC (may need to scale Kp, Ki and Kd up)

Regional Training Centers

PID Mode

PID setup Notes:

- **Sampling Period T:**
  - Must be the sampling period of the system inputs
  - Must be constant

- **Scaler:**
  - Scalers should be implemented by the user to make K1, K2 and K3 integers
  - Make the scaler a power of 2 to keep performance high.

PID Mode

Microchip Regional Training Centers

Comparison of Results – Full PID Conversion

- **Software only**

```
errorPrev = error;
error = setpoint - adc_result;
errorAcc += error;
errorDelta = error - errorPrev;
pid_result += error * Kp + errorAcc * Ki + errorDelta * Kd;
```

~839 instruction cycles for a 32-bit result

- **Using MathACC**

```
pid_result = MATHACC_PIDController(setpoint, adc_result);
```

5 to load input
7 to complete the calculation
6 to save the result

~18 instruction cycles for a 36-bit result
WHY SO FEW???

PID Mode

Accumulator Notes:

- **Integral windup**
- **Cannot solve all loops (slow response system)**

Implementation:

- **Handle the overflow: clamp the accumulator when it's about to overflow.**

```
//Interrupt service routine
void interrupt ISR(void)

  IF (PIR5BITS.PID1EIF==1&&PIE5BITS.PID1EIE==1)
  {
     //saturate the PID1OUT registers
     PID1OUTHH=0xFF;
     PID1OUTHL=0xFF;
     PID1OUTLH=0xFF;
     PID1OUTLL=0xFF;
     PID1OUTHH=0xFF;
     //clear the interrupt flag
     PIR5bits.PID1EIF=0;
  }
```

Microchip Regional Training Centers

LEGAL NOTICE

SOFTWARE:

You may use Microchip software exclusively with Microchip products. Further, use of Microchip software is subject to the copyright notices, disclaimers, and any license terms accompanying such software, whether set forth at the install of each program or posted in a header or text file.

Notwithstanding the above, certain components of software offered by Microchip and 3rd parties may be covered by "open source" software licenses – which include licenses that require that the distributor make the software available in source code format. To the extent required by such open source software licenses, the terms of such license will govern.

NOTICE & DISCLAIMER:

These materials and accompanying information (including, for example, any software, and references to 3rd party companies and 3rd party websites) are for informational purposes only and provided "AS IS." Microchip assumes no responsibility for statements made by 3rd party companies, or materials or information that such 3rd parties may provide.

MICROCHIP DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING ANY IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY DIRECT OR INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND RELATED TO THESE MATERIALS OR ACCOMPANYING INFORMATION PROVIDED TO YOU BY MICROCHIP OR OTHER THIRD PARTIES, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBLITY OF SUCH DAMAGES OR THE DAMAGES ARE FORESEEABLE. PLEASE BE AWARE THAT IMPLEMENTATION OF INTELLECTUAL PROPERTY PRESENTED HERE MAY REQUIRE A LICENSE FROM THIRD PARTIES.

TRADEMARKS:

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KleerNet, KleerNet logo, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.