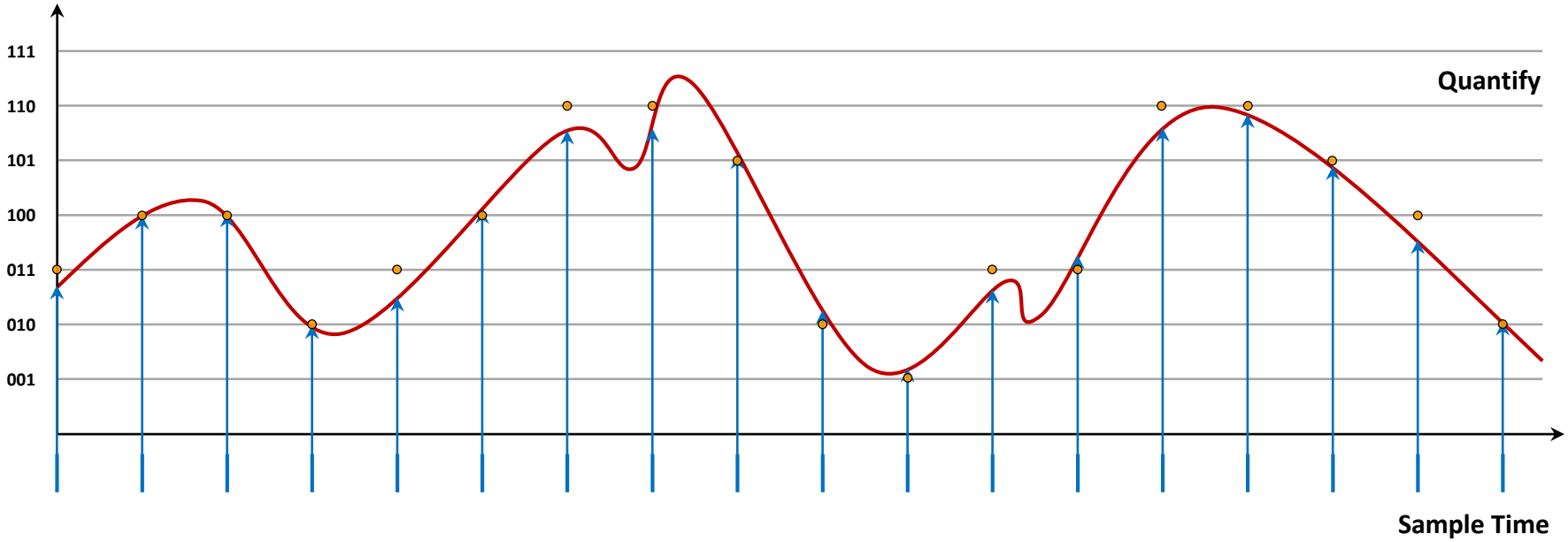
MICROCHIP

Regional Training Centers

Section 10
High Speed 12-Bits ADC Architecture

What's ADC ?

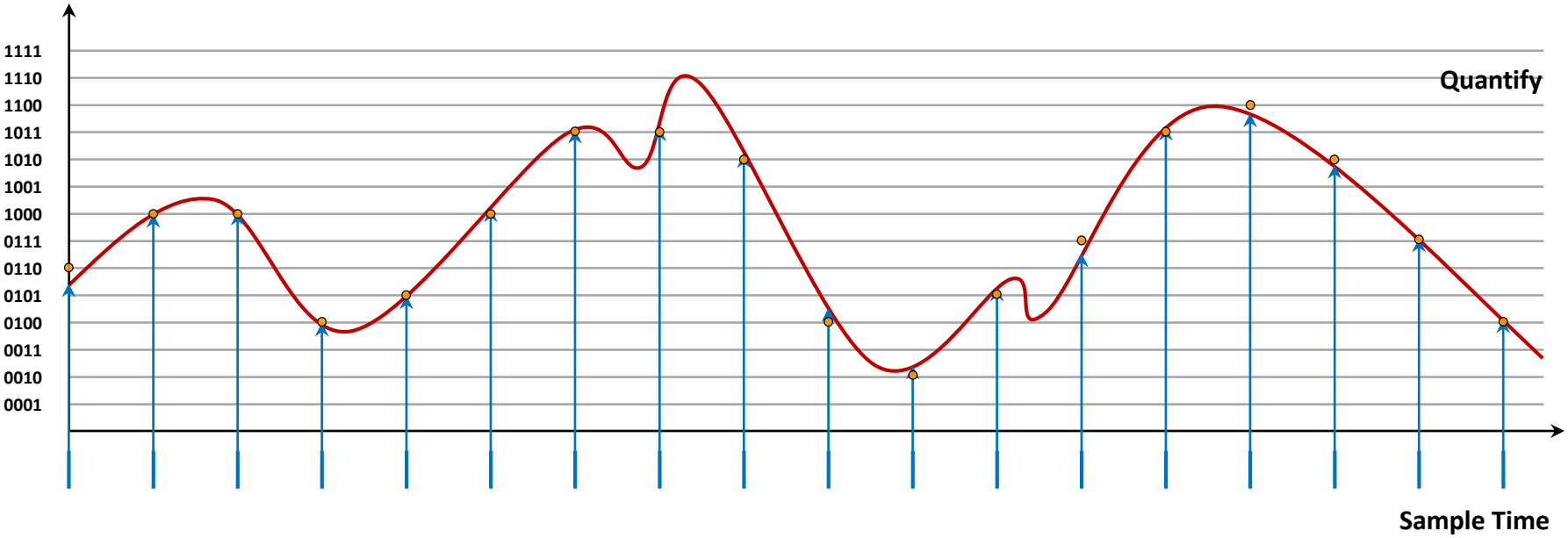
- **The Analog-to-Digital Converter(ADC) converts a signal from the time continuous/amplitude continuous domain into a time discrete/amplitude discrete domain.**



© 2019 Microchip Technology Incorporated. All Rights Reserved.

Resolution

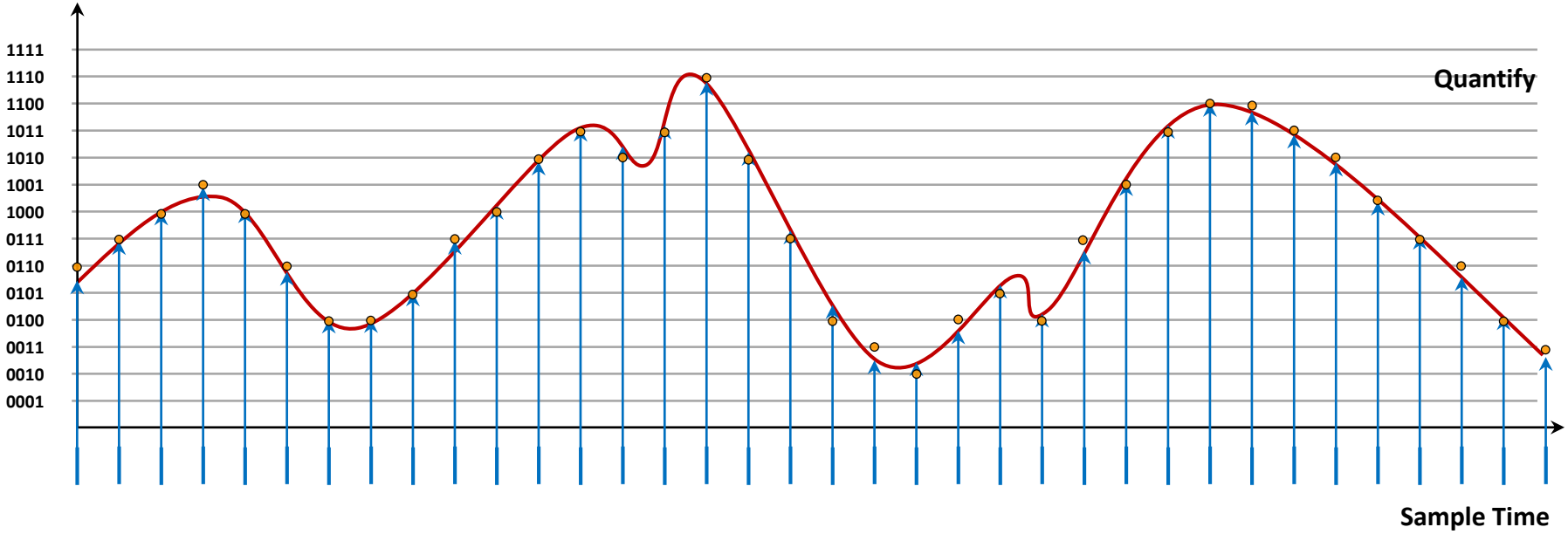
- The resolution of the converter indicates the number of discrete values. The resolution determines the magnitude of the quantization error. (wiki)



© 2019 Microchip Technology Incorporated. All Rights Reserved.

Sample Rate

- The analog signal is required to define the rate at which new digital values are sampled from the analog signal.
- This faithful reproduction is only possible if the sampling rate is higher than twice the highest frequency of the signal. (wiki)

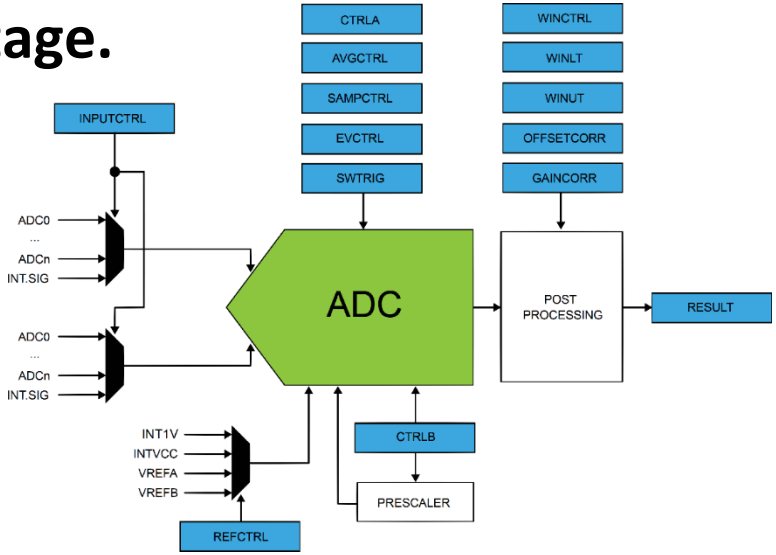


© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Block Diagram

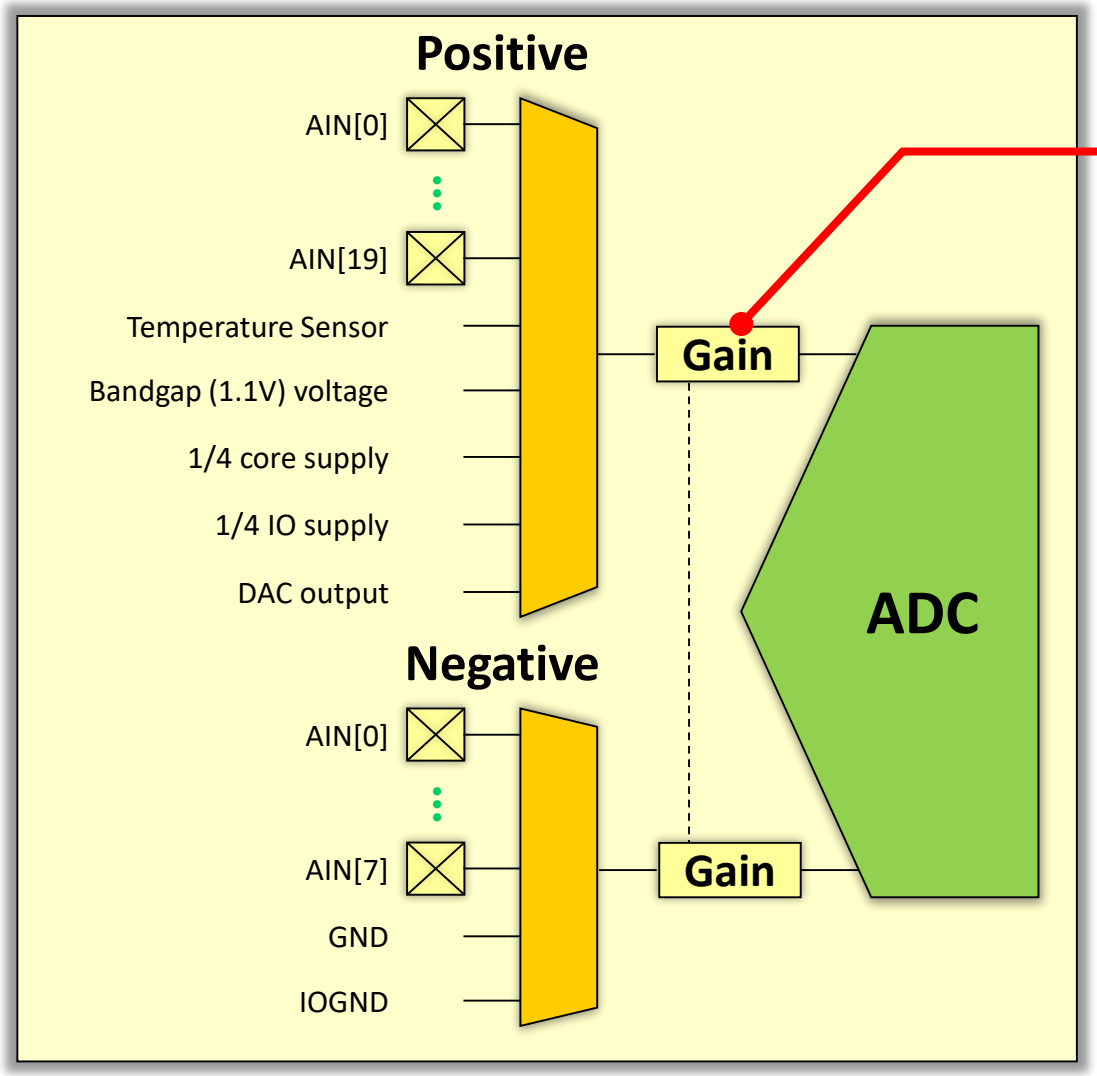
- Up to 32 analog input, 8, 10 or 12 bits resolution.
- Differential and single-ended inputs.
- 1/2x to 16x gain. Up to 350ksps.
- Support Event-triggered and pin scan.
- Support averaging and oversampling.
- External or Internal reference voltage.
- Oversample accumulation average and window monitor feature.

$$Reusult_{AD} = \left(\frac{AN_{+} - V_{Ref-}}{V_{Ref+} - V_{Ref-}} \times 2^{n} \right) - 1$$



© 2019 Microchip Technology Incorporated. All Rights Reserved.

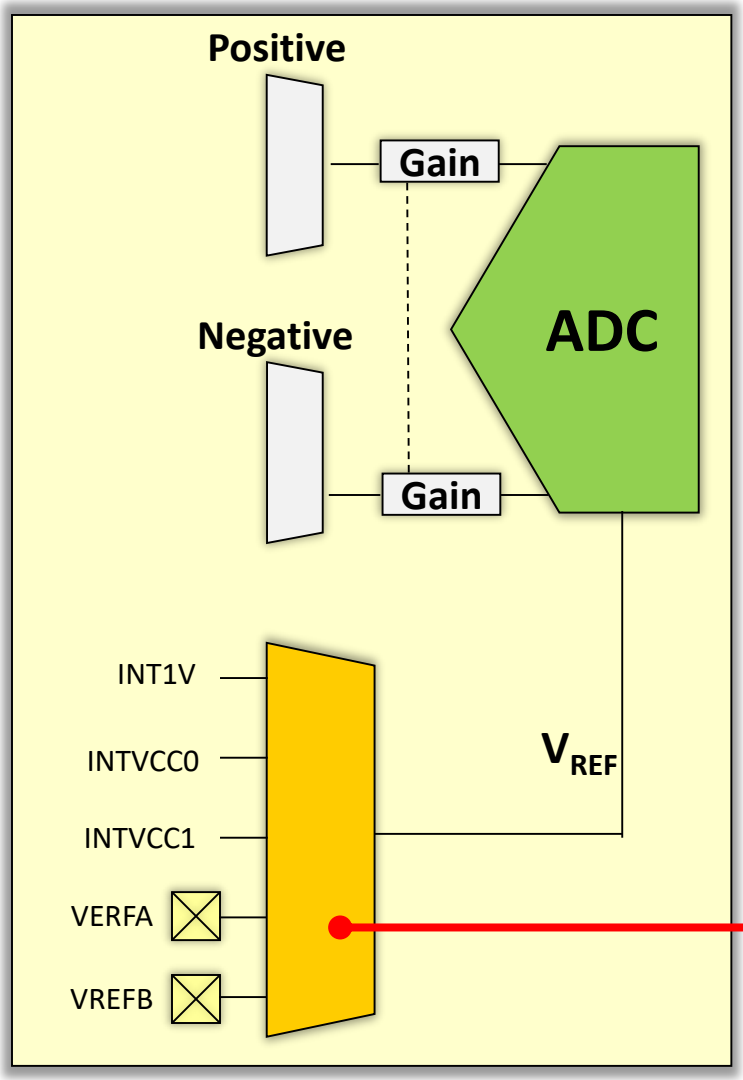
ADC Channel Selection & Gain



GAIN[3:0]	Name
0x0	1X
0x1	2X
0x2	4X
0x3	8X
0x4	16X
0x5-0xE	Reserved
0xF	1/2x

© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Reference Voltage Selection



VREFA and VREFB range

$1.0V \sim (V_{DDANA} - 0.6V)$

Conversion range:

– Single-ended mode

$0V \sim V_{REF}/Gain$

– Differential mode

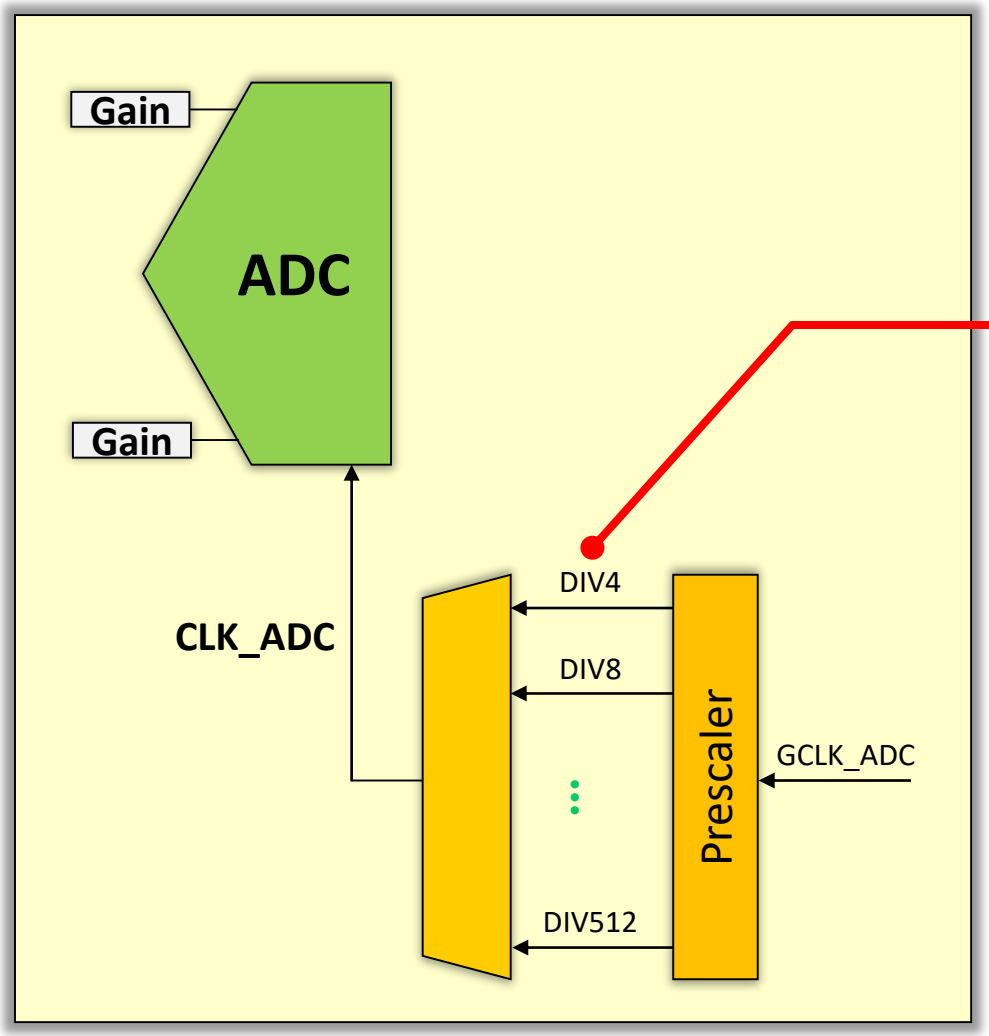
$-V_{REF}/Gain \sim V_{REF}/Gain$

REFSEL[3:0]	Name	Description
0x0	INT1V	1.0V voltage reference
0x1	INTVCC0	1/1.48 VDDANA
0x2	INTVCC1	1/2 VDDANA (only for VDDANA > 2.0V)
0x3	VREFA	External reference
0x4	VREFB	External reference
0x5-0xF		Reserved

VDDANA = 3.3V
½ VDDANA = 1.65V

© 2019 Microchip Technology Incorporated. All Rights Reserved.

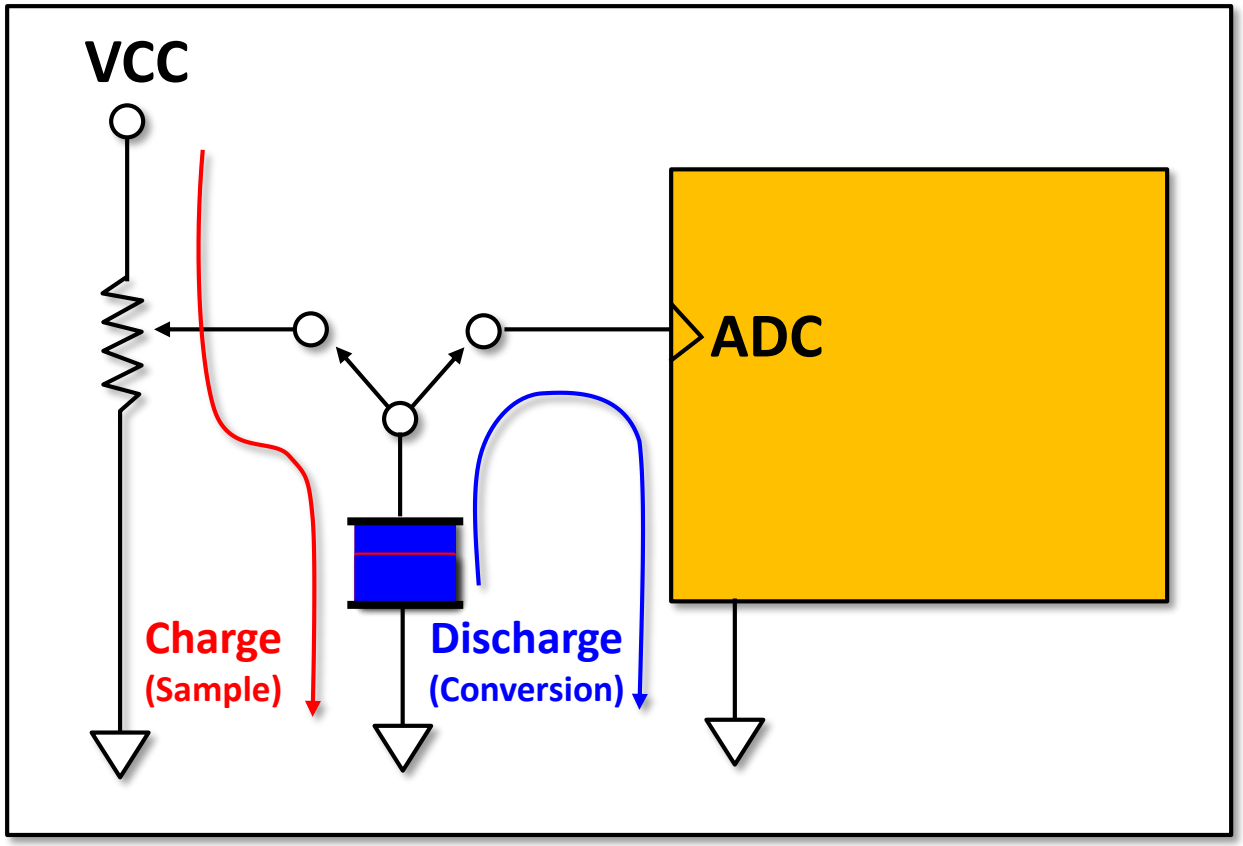
ADC Prescaler



PRESCALER[2:0]	Name
0x0	DIV4
0x1	DIV8
0x2	DIV16
0x3	DIV32
0x4	DIV64
0x5	DIV128
0x6	DIV256
0x7	DIV512

© 2019 Microchip Technology Incorporated. All Rights Reserved.

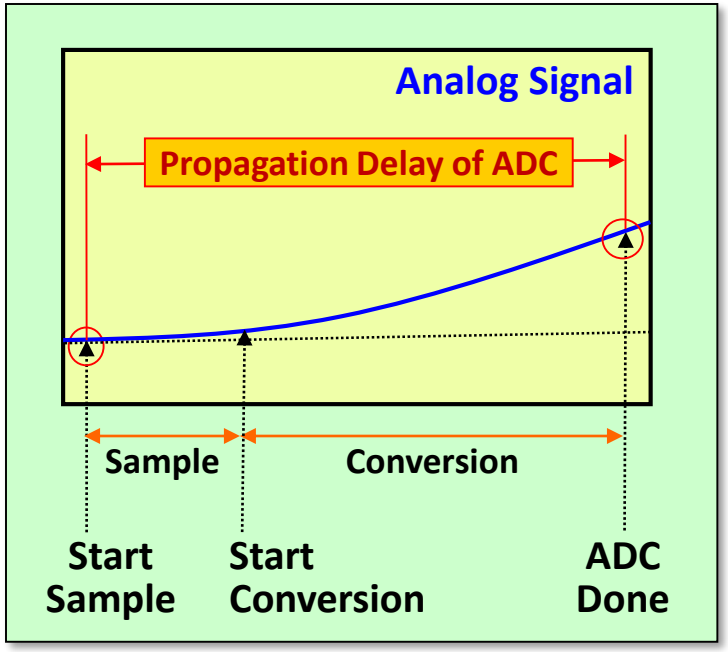
Sample and Conversion Sequence



© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Characteristics

- **Maximum Sampling Rate :**
  Single-shot : 2.1M/Round(6+0.5) = up to 300k samples/second.
  Free-running : 2.1M/6 = up to 350k samples/second.



Symbol	Parameter	Conditions	Min.	Max.	Units
RES	Resolution		8	12	bits
f_{CLK_ADC}	ADC Clock frequency		30	2100	kHz
	Sample rate	Single shot	5	300	ksps
		Free running	5	350	ksps
	Sampling time		0.5	-	cycles
	Conversion time	1x Gain	6	-	cycles

© 2019 Microchip Technology Incorporated. All Rights Reserved.

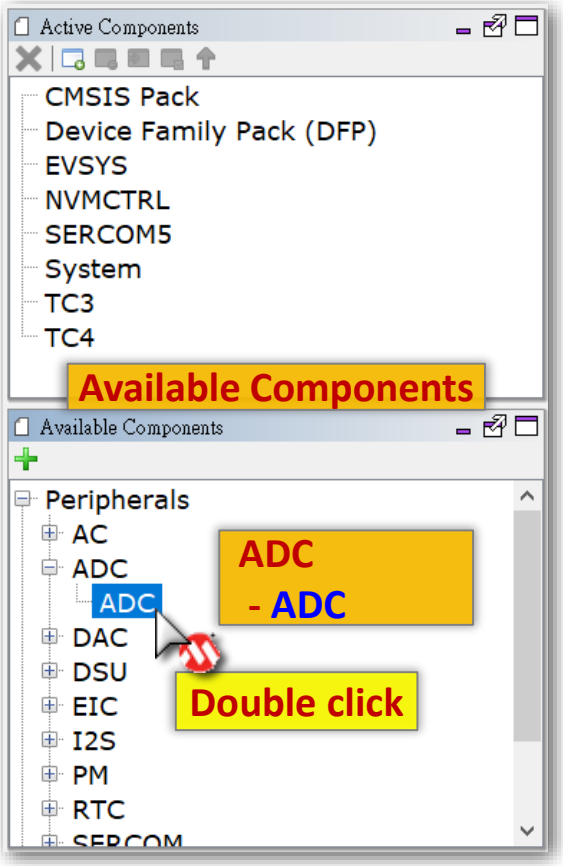
ADC Trigger Source

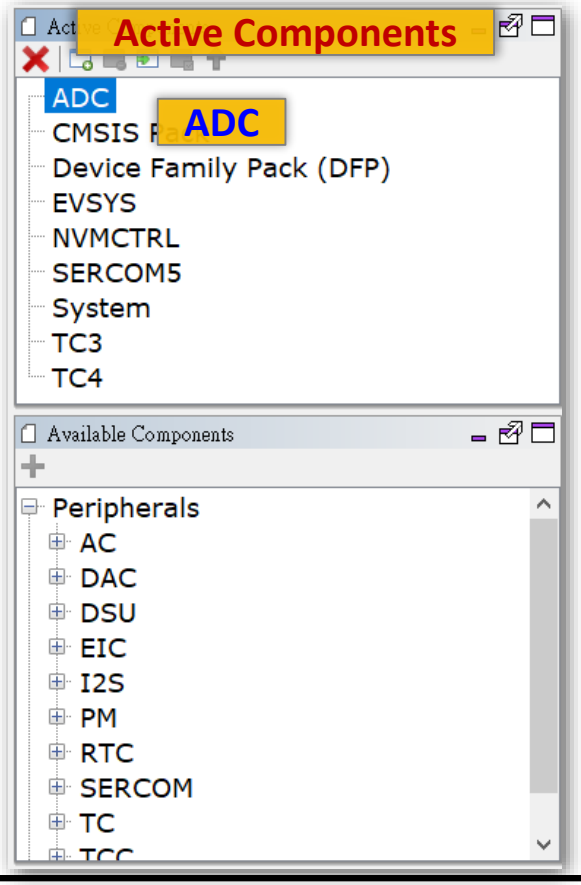
- **There are three way to be started data conversion.**
  - **Software Trigger**
    One shot conversion control via software.
  - **Free Run**
    A free-running mode can be used to continuously convert an input channel. When using free-running mode the first conversion must be started, while subsequent conversions will start automatically at the end of previous conversions.
  - **HW Event Trigger**
    Through Event System peripheral to automatically conversion.

© 2019 Microchip Technology Incorporated. All Rights Reserved.

Add ADC Function using MHC

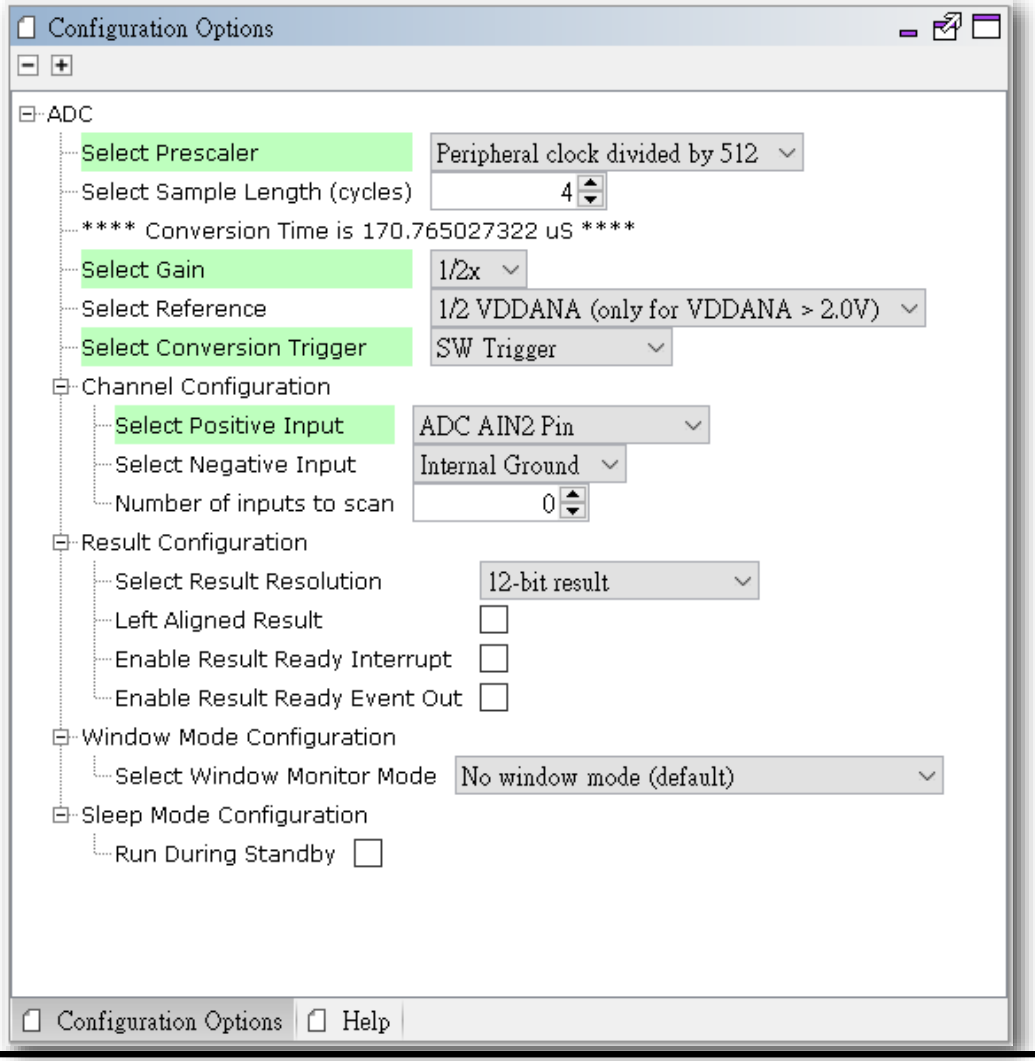
- Find **ADC** component in Available Components window.
- Double click **ADC** to add to Active Components window.





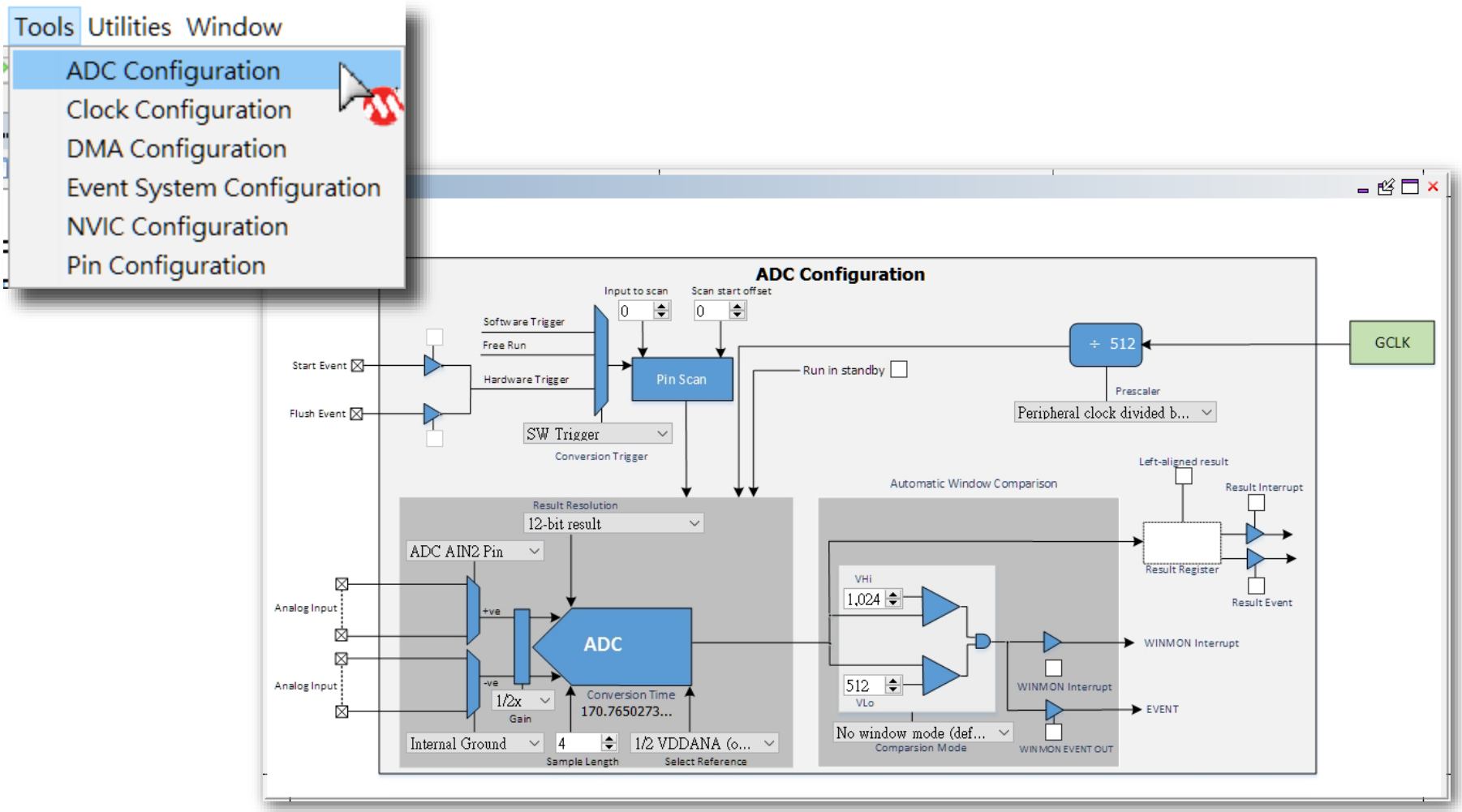
© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Polling
Configuration Options Example



© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Polling
Easy View Configuration Example



© 2019 Microchip Technology Incorporated. All Rights Reserved.

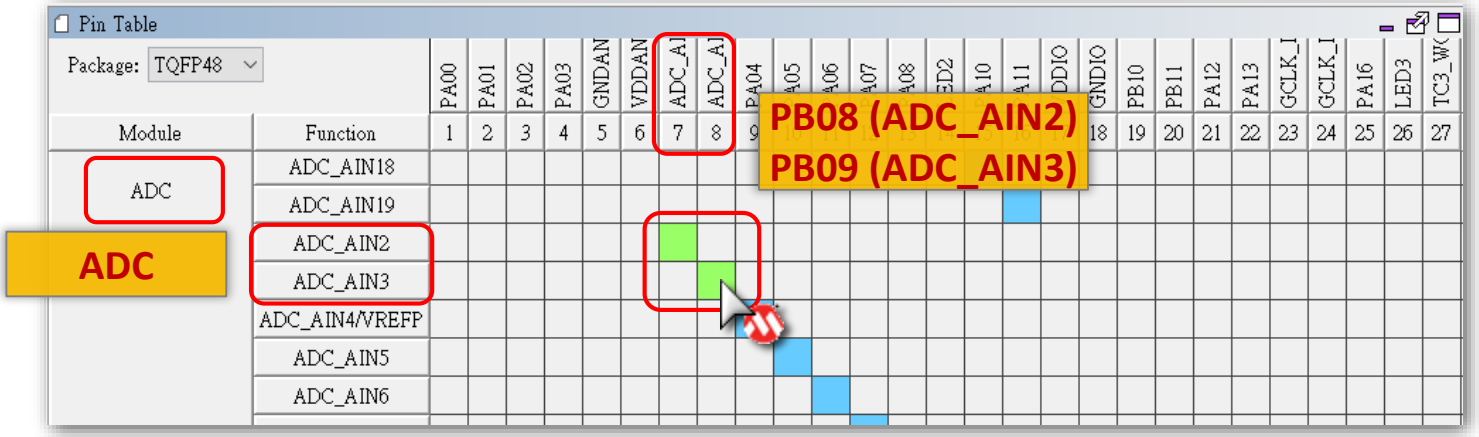
ADC PINMUX Configuration Example

- **Analog pins** will all belong to **PINMUX function group B**.
- For example : **PB08** as **AIN[2]** (Analog input Channel 2)
  PB09 as **AIN[3]** (Analog input Channel 3)

Pin(1)			I/O Pin	Supply	A	B(2)(3)					C	D	E	F	G	H
SAMD21E	SAMD21G	SAMD21J			EIC	REF	ADC	AC	PTC	DAC	SERCOM(2)(3)	SERCOM-ALT	TC(4) /TCC	TCC	COM	AC/ GCLK
	7	11	PB08	VDDANA	EXTINT[8]		AIN[2]		Y[14]			SERCOM4/ PAD[0]	TC4/WO[0]	TCC3/ WO[6]		
	8		PB09	VDDANA	EXTINT[9]		AIN[3]					SERCOM4/ PAD[1]	TC4/WO[1]	TCC3/ WO[7]		

PB08
PB09

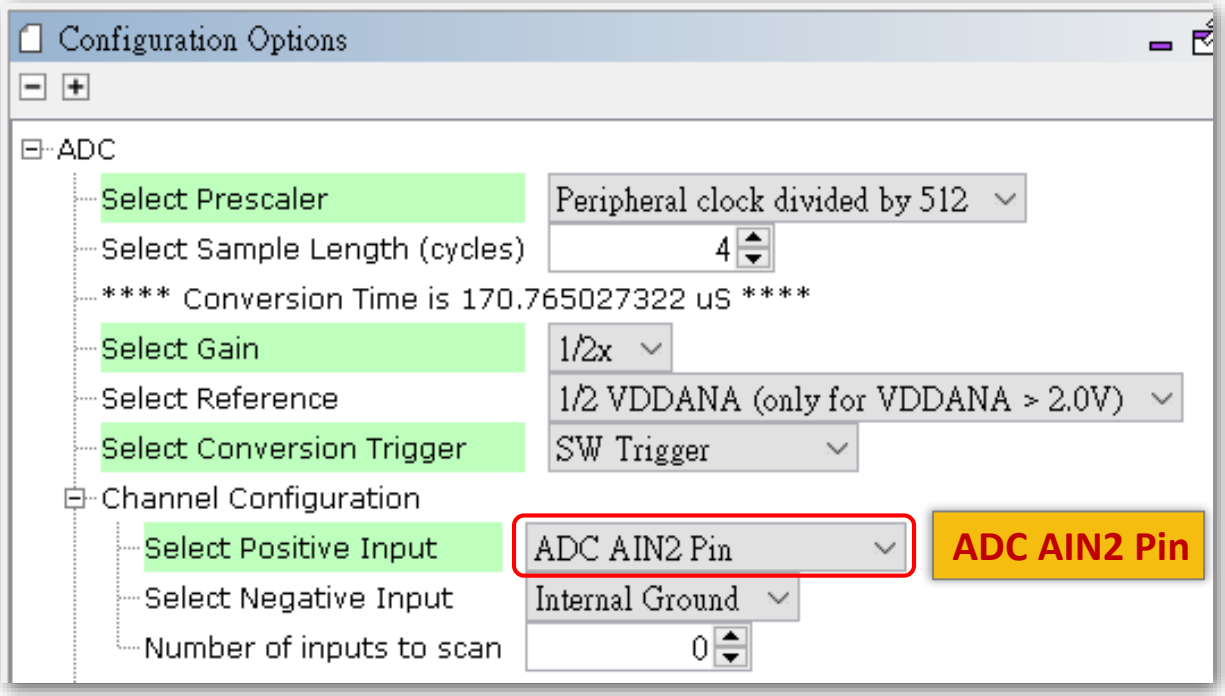
AIN[2]
AIN[3]



© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC PINMUX Configuration Example

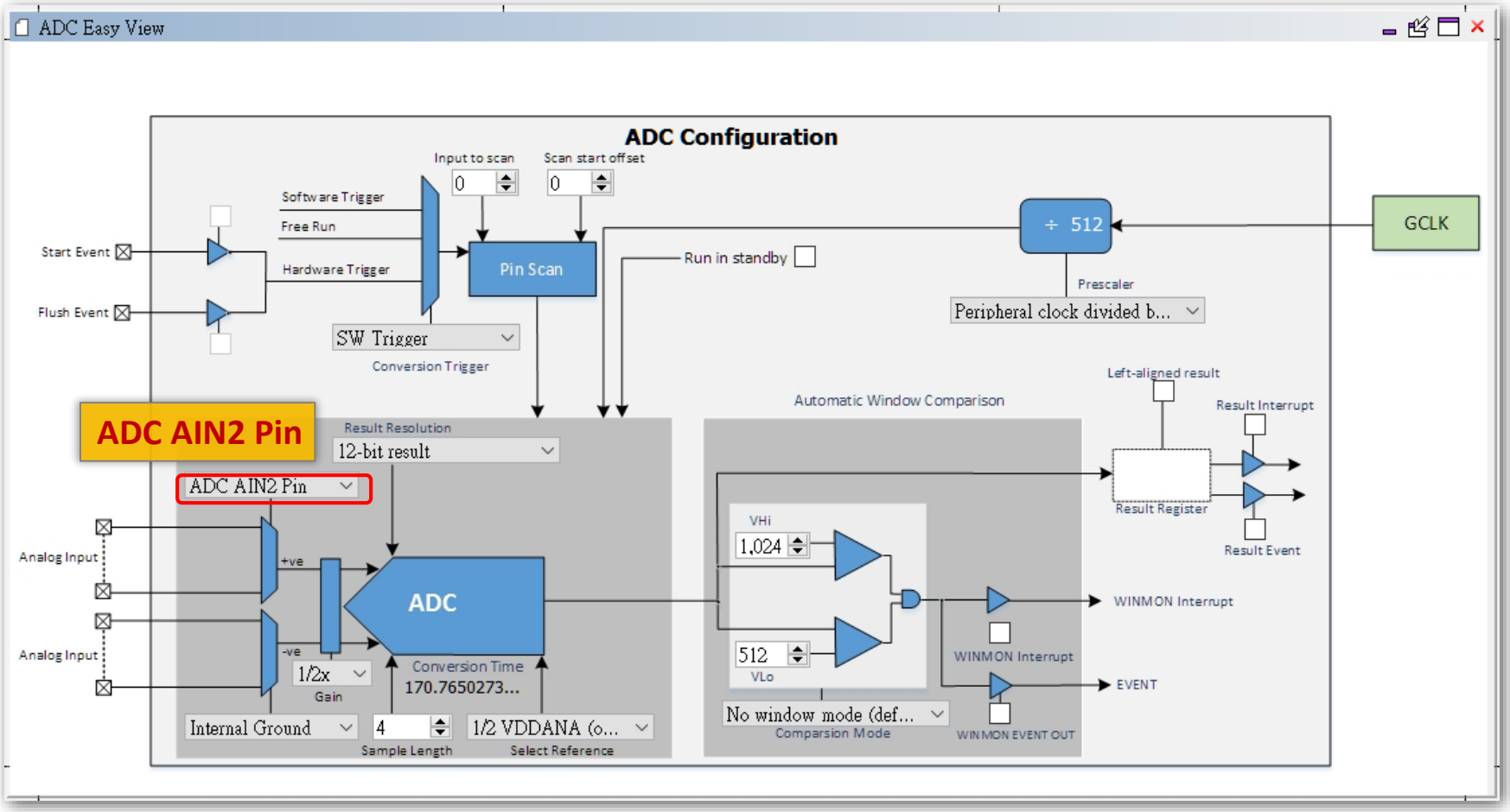
- Remember assign **Positive Input** pin in ADC module.
- For example : **ADC AIN2 Pin** as Positive Input Pin (**PB08**)



© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC PINMUX

Easy View Configuration Example



© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Polling Interface Routines

```
/*** ADC Interface Routines***/
void ADC_Initialize( void );
void ADC_Enable( void );
void ADC_Disable( void );
void ADC_ChannelSelect( ADC_POSINPUT positiveInput, ADC_NEGINPUT negativeInput );
void ADC_ConversionStart( void );
uint16_t ADC_ConversionResultGet( void );
bool ADC_ConversionStatusGet( void );
void ADC_ComparisonWindowSet(uint16_t low_threshold, uint16_t high_threshold);
void ADC_WindowModeSet(ADC_WINMODE mode);
bool ADC_ConversionStatusGet( void );
```

© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Polling Software Trigger Code Example

```
 …

int main (void)
{
   SYS_Initialize ( NULL );
   …
   ADC_Enable();
   while(1)
   {
      …
      ADC_ConversionStart();
      while( !ADC_ConversionStatusGet() );
      printf( "%d\r\n", ADC_ConversionResultGet() );
   }
}
```

© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab12 ADC Polling Software Trigger

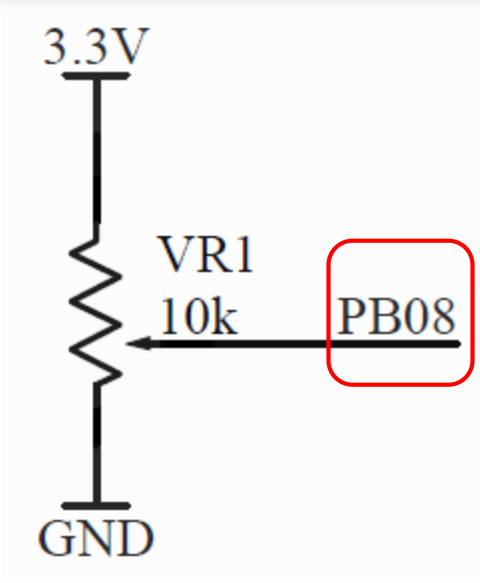
- Try to add ADC function to project.
- Initial ADC and get the conversion value from VR1 (AIN[2]) with one second period, then use UART to output to PC.
- ADC set to Software Trigger, 12 bits resolution, Single-End mode.

- **Let's go!**

© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab12 ADC Polling Software Trigger

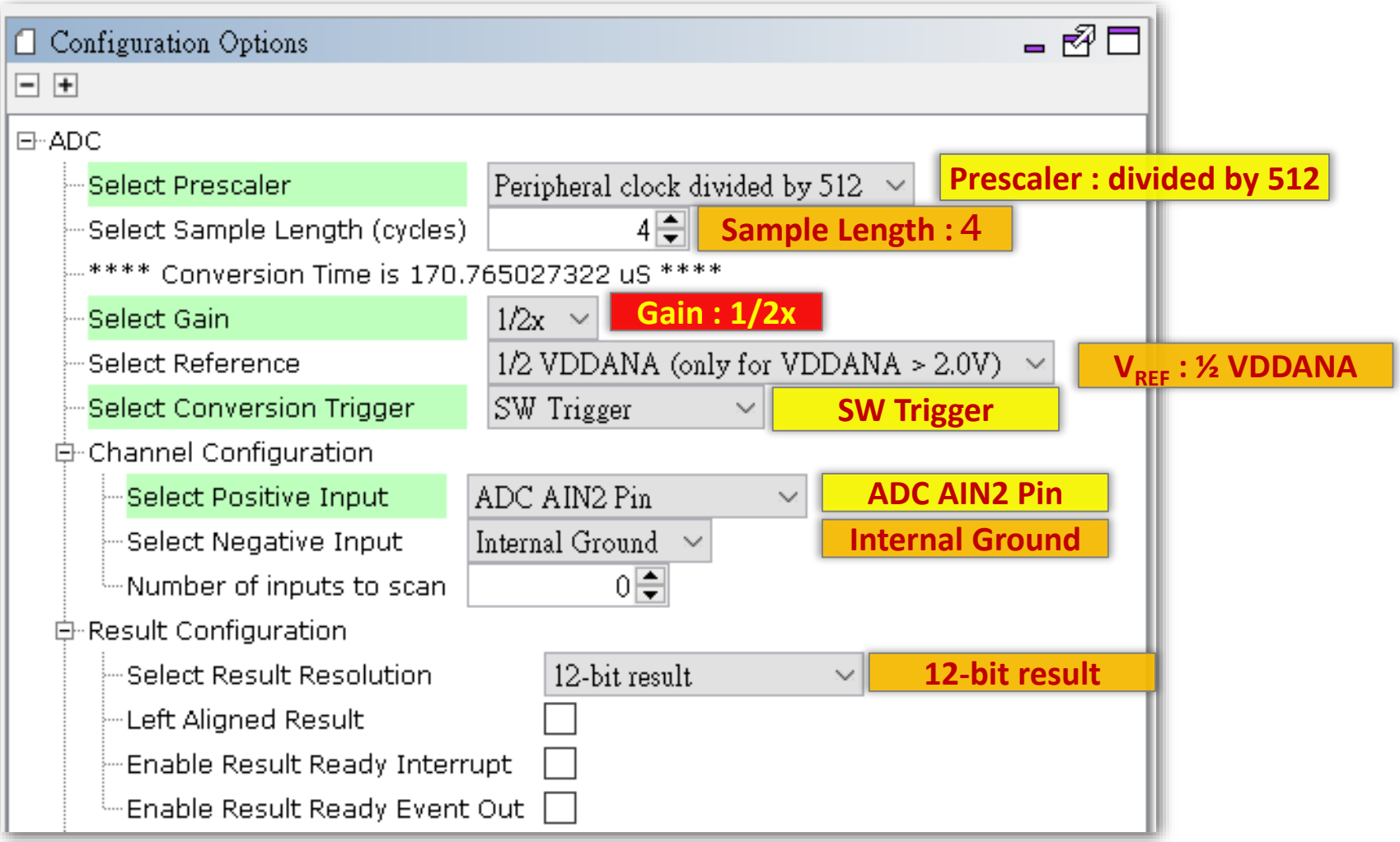
Pin(1)			I/O Pin	Supply	A	B(2)(3)				
SAMD21E	SAMD21G	SAMD21J			EIC	REF	ADC	AC	PTC	DAC
	7	11	PB08	VDDANA	EXTINT[8]		AIN[2]		Y[14]	
	8	12	PB09	VDDANA	EXTINT[9]		AIN[3]		Y[15]	



© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab12 ADC Polling Software Trigger

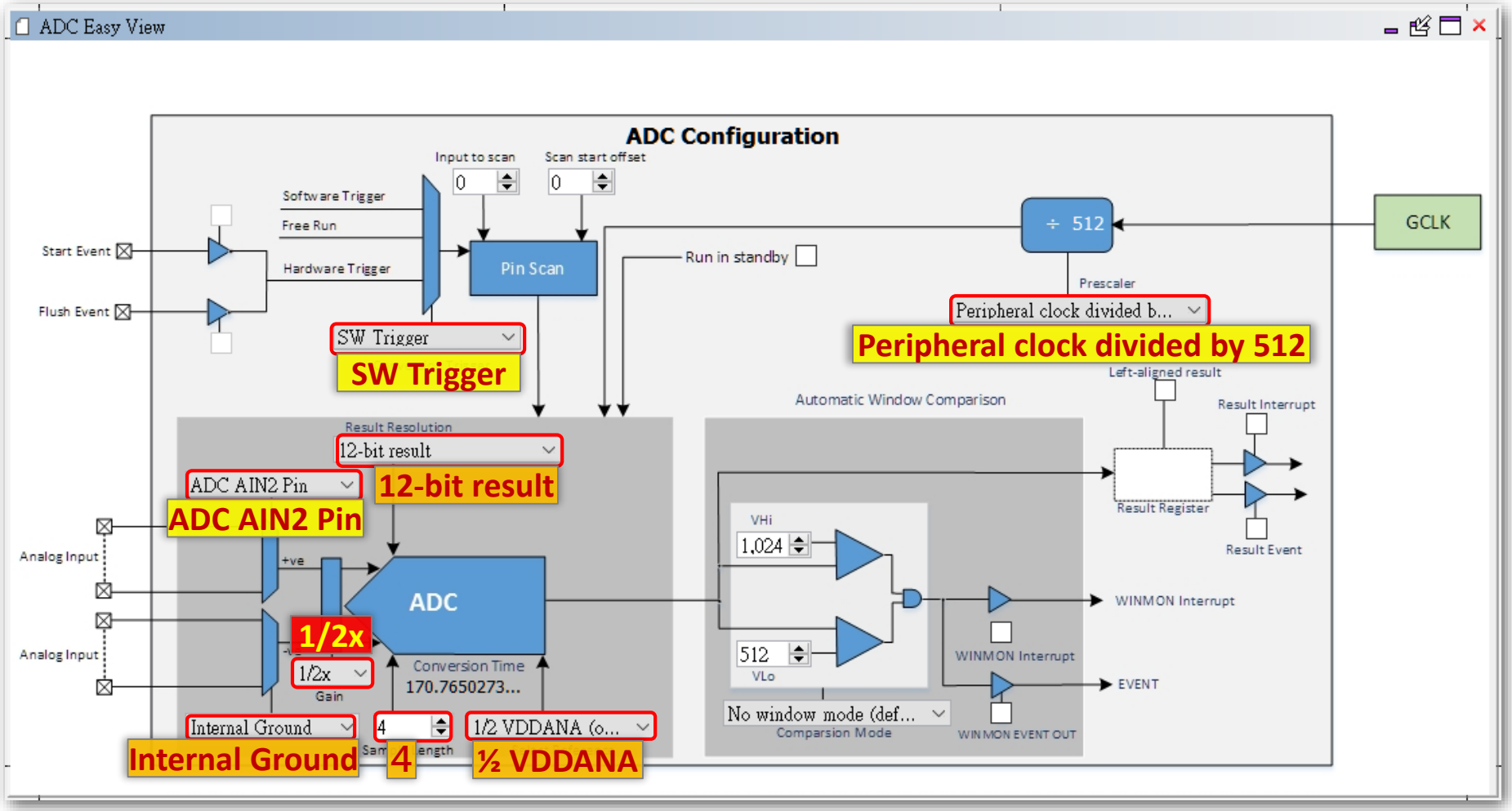
Step 1 (Use Configuration Options)



© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab12 ADC Polling Software Trigger

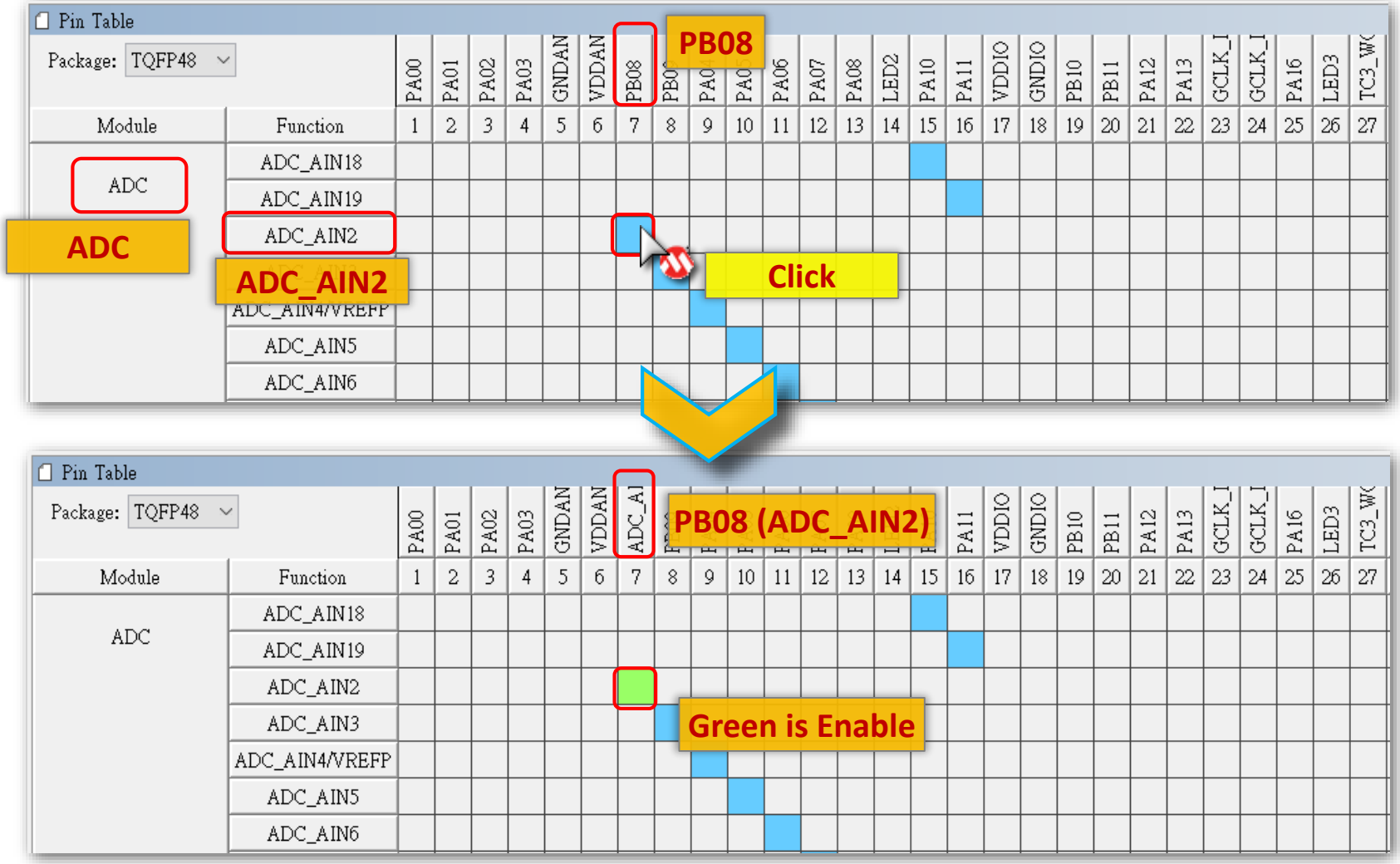
Step 1 (Use Easy View)



© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab12 ADC Polling Software Trigger

Step 2

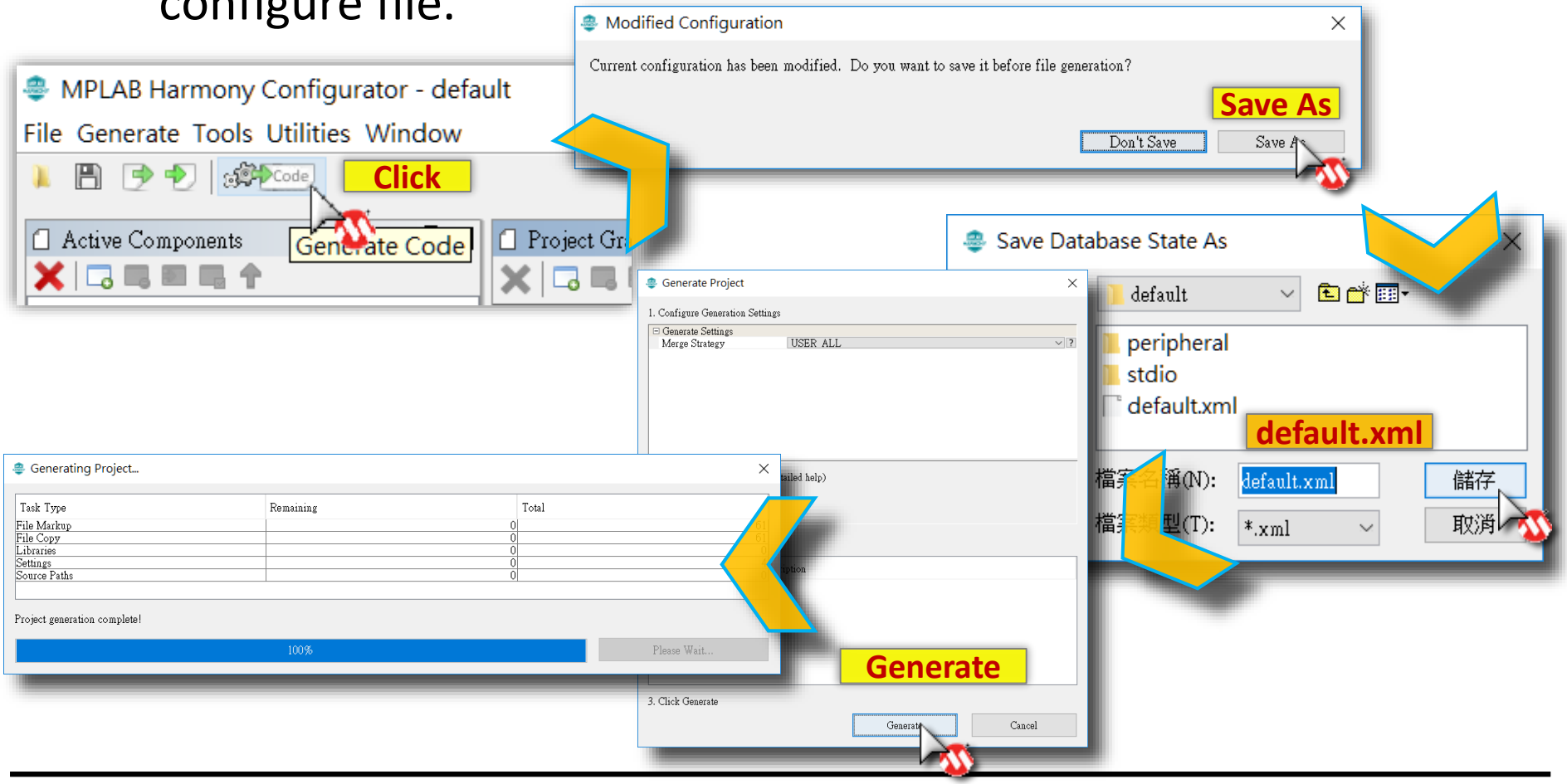


© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab12 ADC Polling Software Trigger

Step 3

- Click to Generate Code and save changes to MHC configure file.



© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab12 ADC Polling Software Trigger

Step 4

a **Add code segment to your main.c**

```
// TODO 12.01
uint16_t ADC_Result;

int main (void)
{  …
   // TODO 12.02
   ADC_Enable();

   while(1)
   {  …
      if( TC3_IsOverflow )
      {
         // TODO 12.03
         ADC_ConversionStart();
         while( !ADC_ConversionStatusGet() );
         ADC_Result = ADC_ConversionResultGet();
         printf( "ADC Data : %d\r\n", ADC_Result );
      }
      …
   }
 }
```

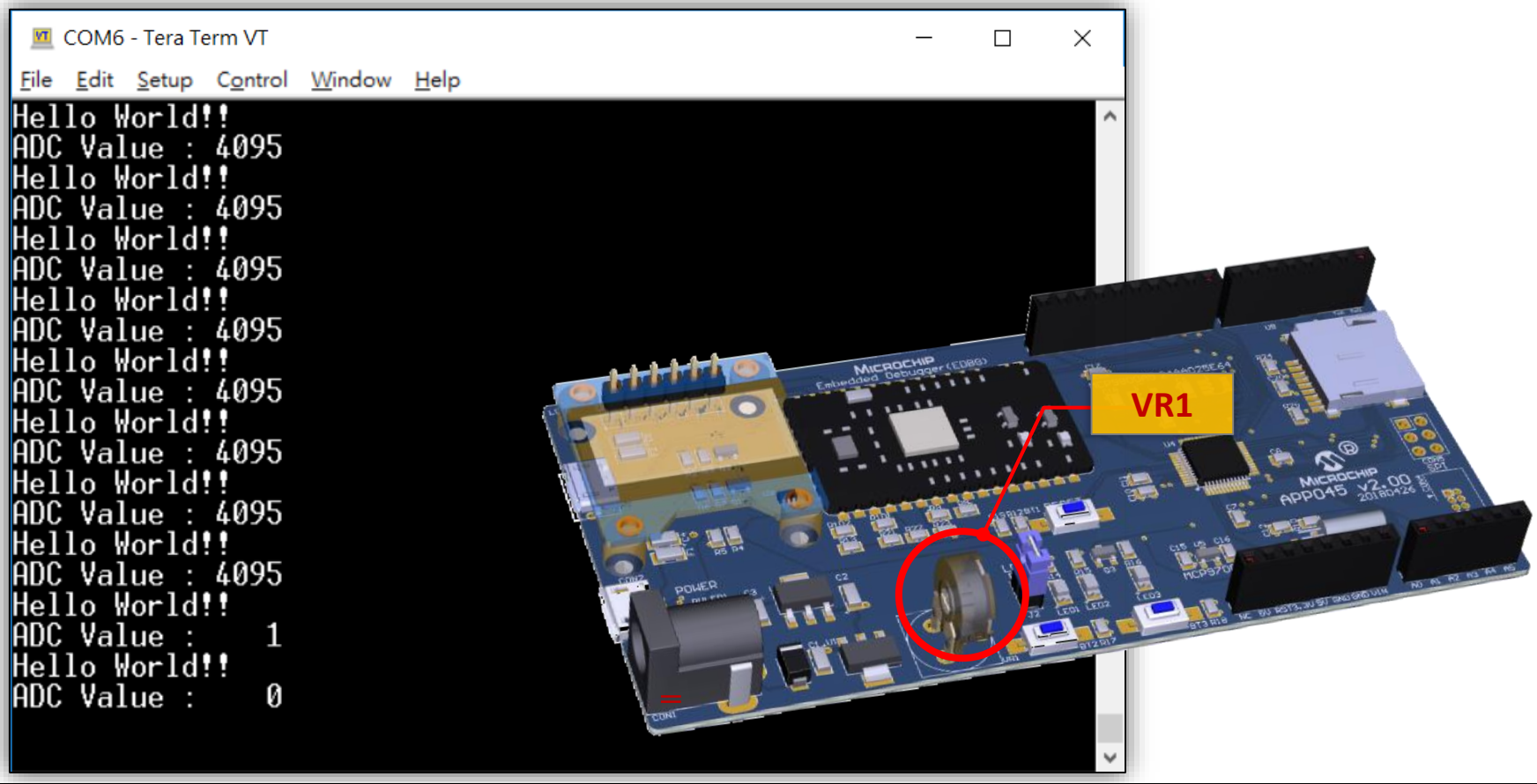
b **Program firmware to target board then observe result.**

© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab12 ADC Polling Software Trigger

Result

- "Hello World!!" string and VR1 ADC value will output to terminal with one second period.



© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Pin Scan

- The ADC support analog channel switch call Pin Scan function.
- Pin scan switch analog channel sequential after a conversion automatically.
- Start Pin : Positive Input + Scan Start Offset
- End Pin : Start Pin + Number of Inputs to Scan

For Example : To do ADC Pin Scan from AIN[2] to AIN[3].

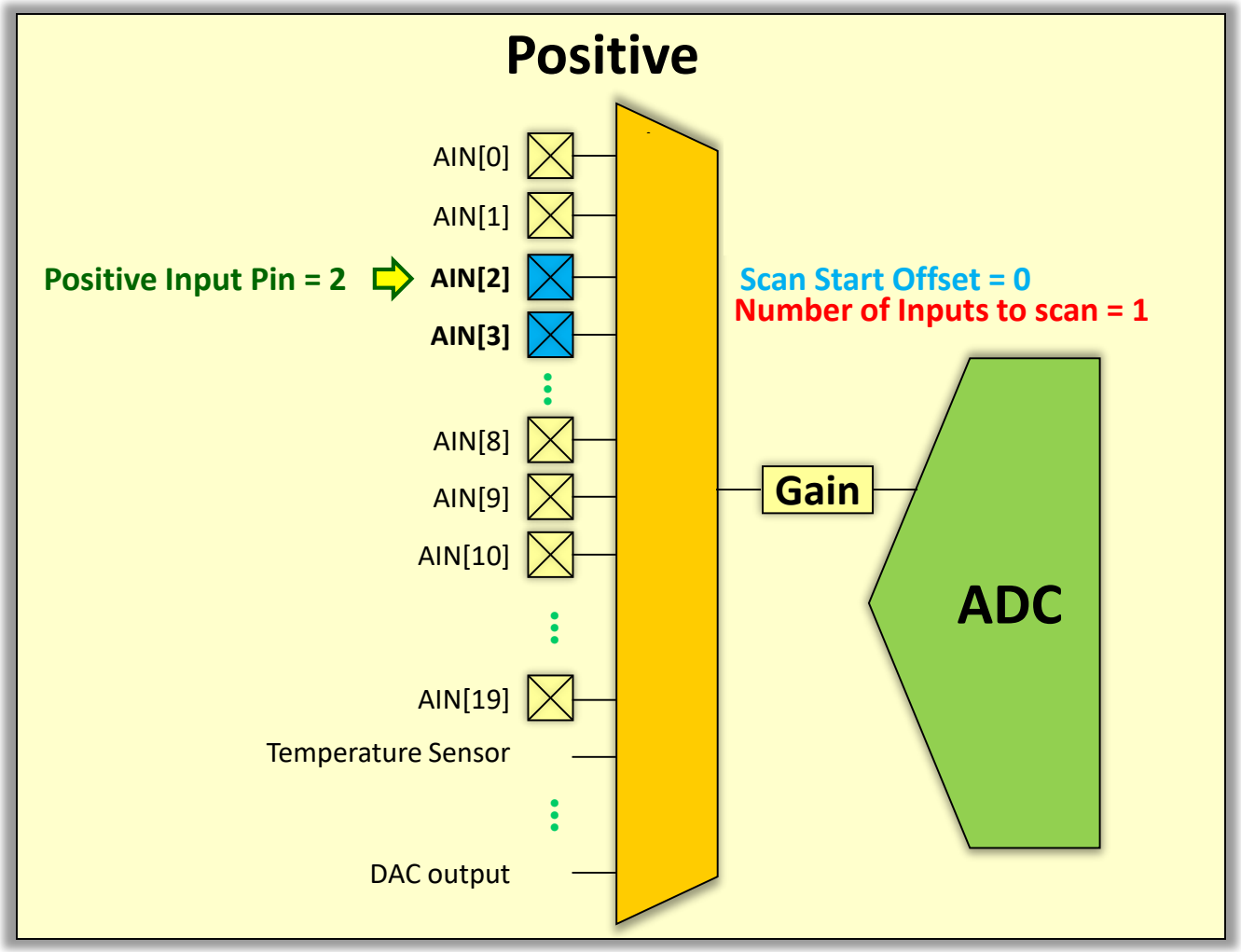
Positive Input Pin = AIN[0] Scan Start Offset = 2 Start Pin = AIN[0+2] = AIN[2]	OR	Positive Input Pin = AIN[2] Scan Start Offset = 0 Start Pin = AIN[2+0] = AIN[2]

Number of Inputs to scan = 1

End Pin = AIN[2+1] = AIN[3]

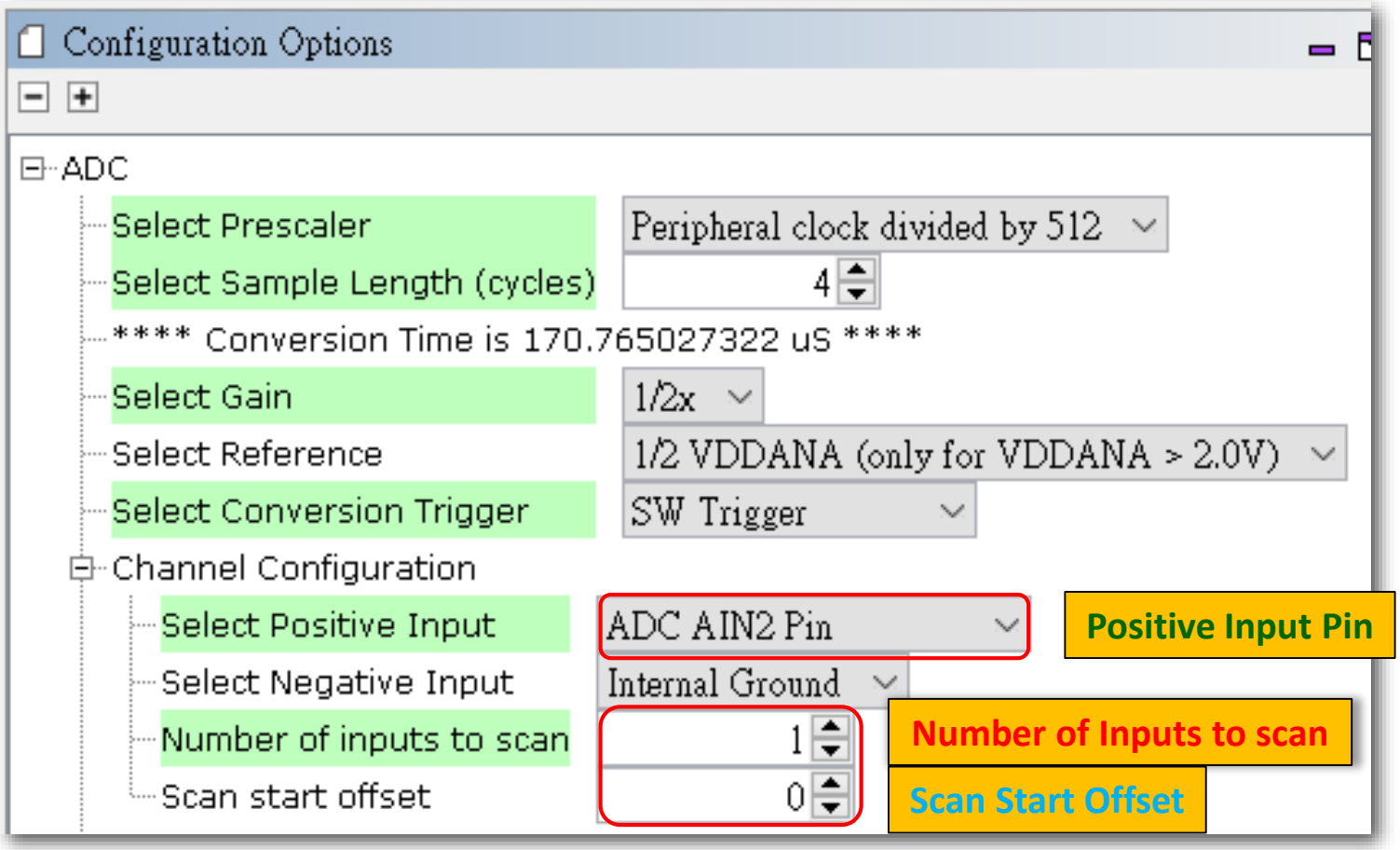
© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Pin Scan



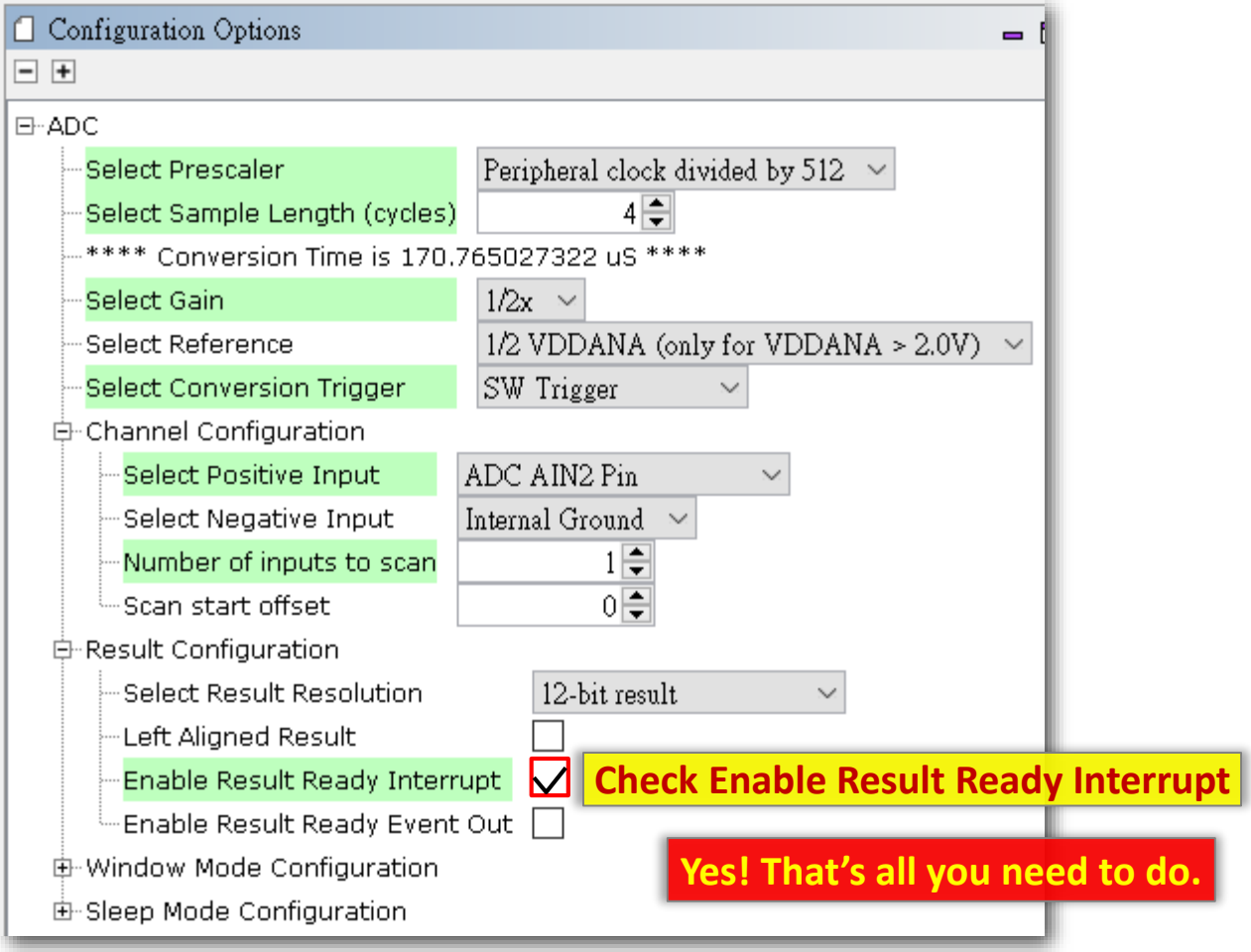
© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Pin Scan Configuration Example



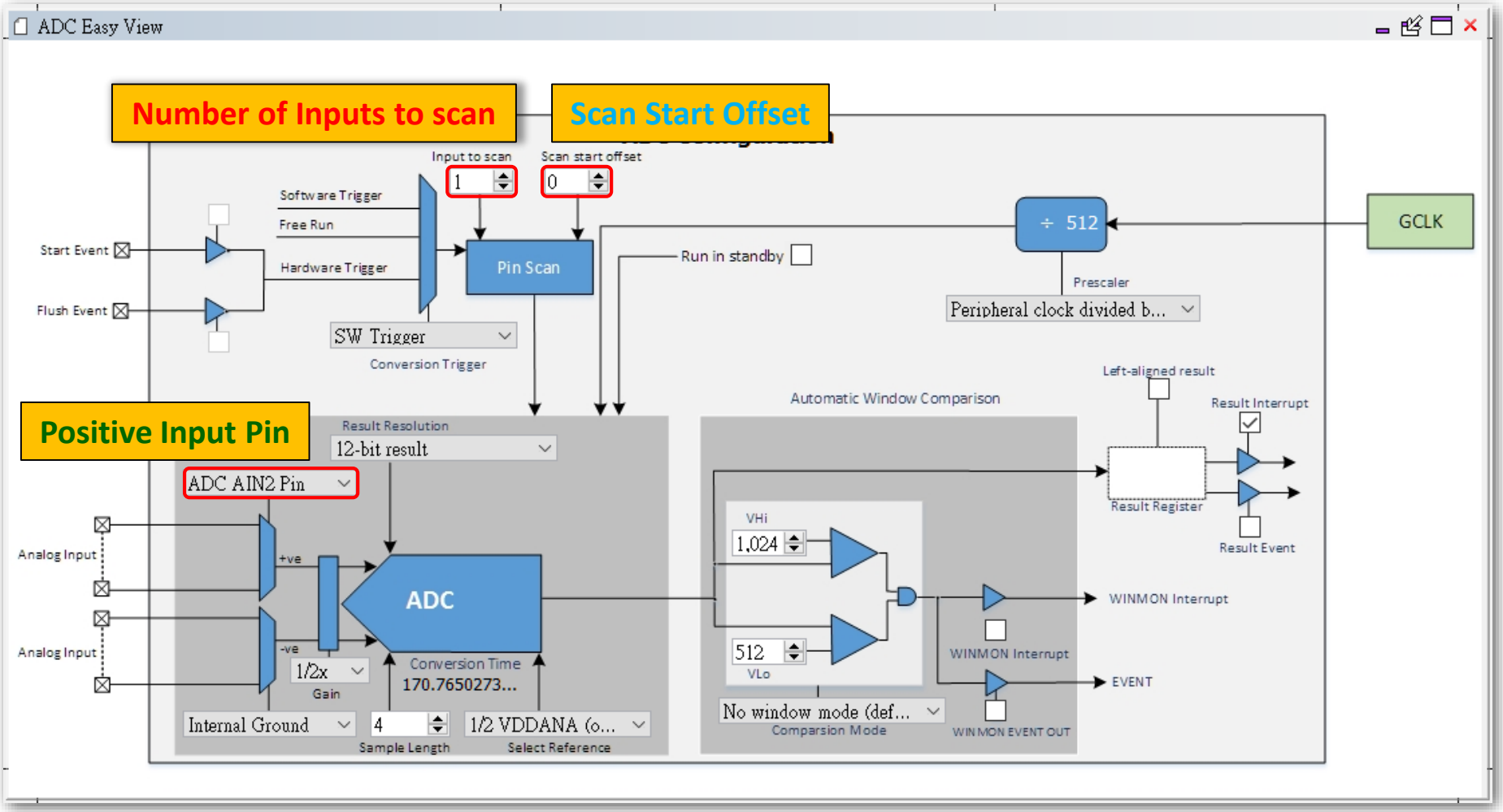
© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Interrupt Configuration Options Example



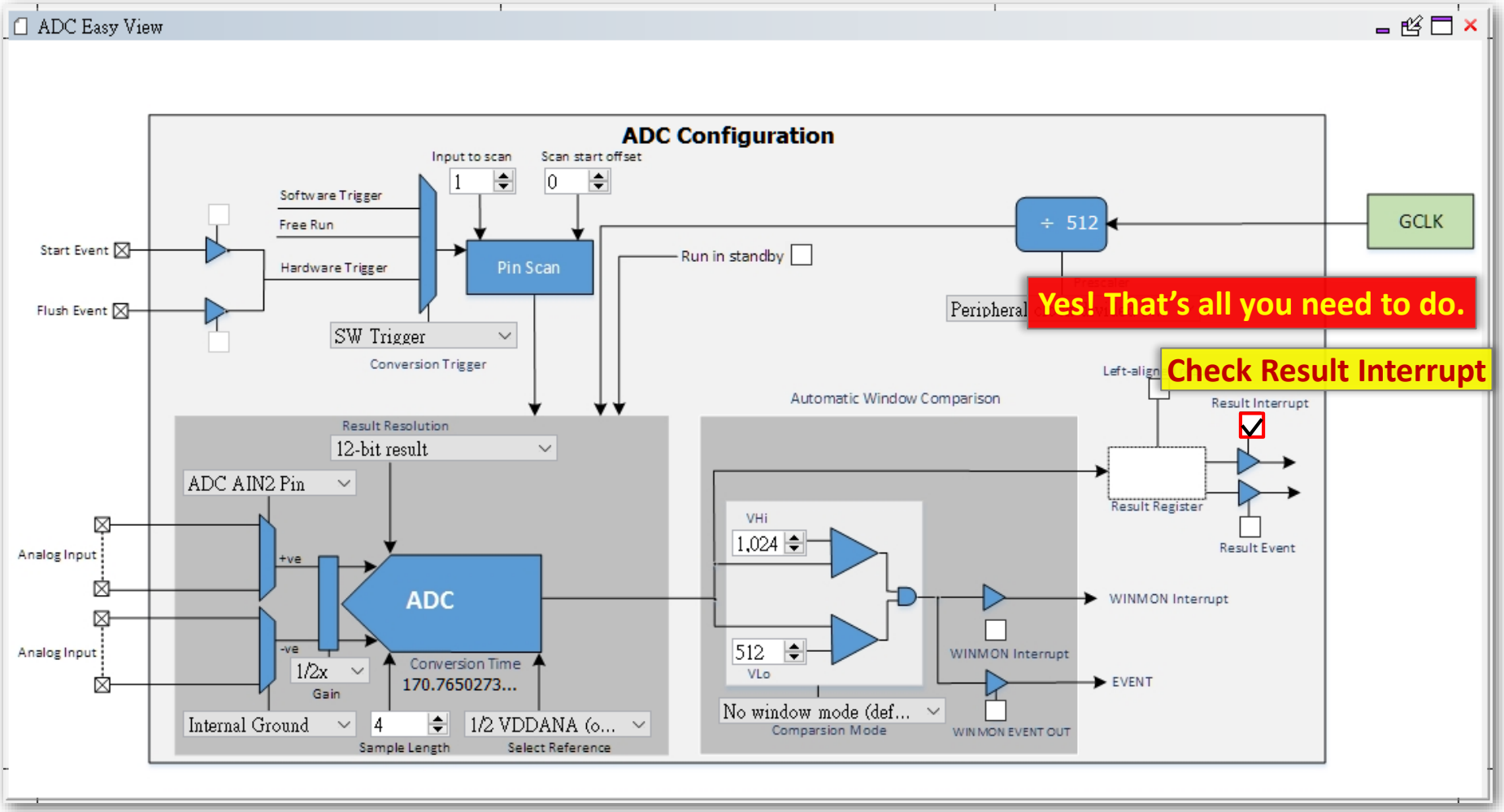
© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Pin Scan
Easy View Configuration Example



© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Interrupt
Easy View Configuration Example



© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Interrupt Callback Interface Routines

```
/*** ADC Interface Routines***/
void ADC_Initialize( void );
void ADC_Enable( void );
void ADC_Disable( void );
void ADC_ChannelSelect( ADC_POSINPUT positiveInput, ADC_NEGINPUT negativeInput );
void ADC_ConversionStart( void );
uint16_t ADC_ConversionResultGet( void );
void ADC_ComparisonWindowSet(uint16_t low_threshold, uint16_t high_threshold);
void ADC_WindowModeSet(ADC_WINMODE mode);
void ADC_CallbackRegister( ADC_CALLBACK callback, uintptr_t context );
```

© 2019 Microchip Technology Incorporated. All Rights Reserved.

ADC Pin Scan, Interrupt Callback Software Trigger Code Example

```
 …
void ADC_Complete( ADC_STATUS status, uintptr_t context )
{
   if( status & ADC_INTFLAG_RESRDY_Msk )
   {
      printf( "%d\r\n", ADC_ConversionResultGet() );
   }
   ADC_ConversionStart();
}

int main (void)
{  …
   ADC_CallbackRegister( ADC_Complete, (uintptr_t )NULL );
   ADC_Enable();
   ADC_ConversionStart();
   while(1)
   {  …
   }
}
```

© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab13 ADC Pin Scan

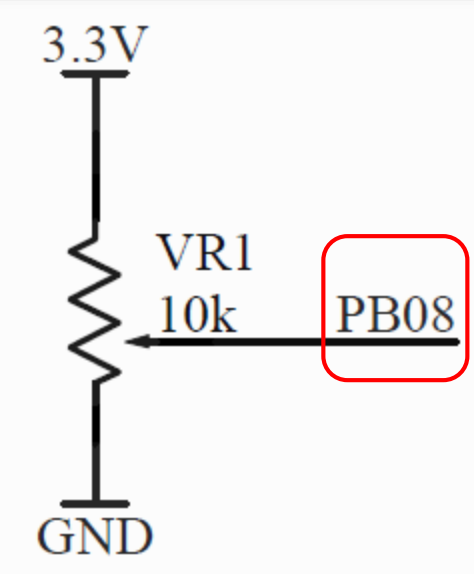
- Change ADC polling to Interrupt callback mode and enable pin scan function.
- Get the ADC conversion value from VR1(AIN[2]) and Analog Temperature Sensor(AIN[3]) with one second period and then use UART to output ADC values to PC.
- ADC settings same as Lab11.

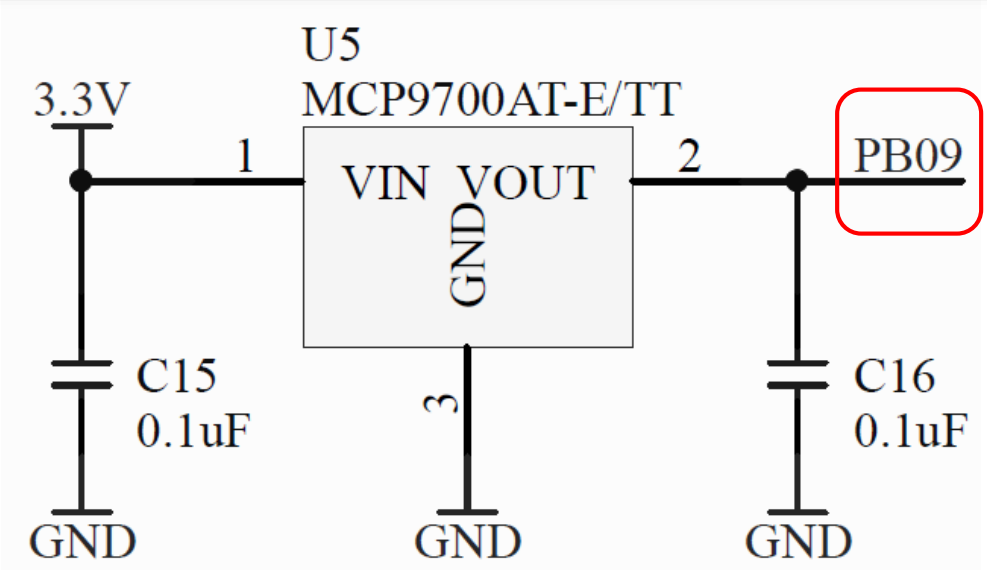
- **Let's go!**

© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab13 ADC Pin Scan

Pin(1)			I/O Pin	Supply	A	B(2)(3)				
SAMD21E	SAMD21G	SAMD21J			EIC	REF	ADC	AC	PTC	DAC
	7	11	PB08	VDDANA	EXTINT[8]		AIN[2]		Y[14]	
	8	12	PB09	VDDANA	EXTINT[9]		AIN[3]		Y[15]	





© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab13 ADC Pin Scan

Step 1 (Use Configuration Options)

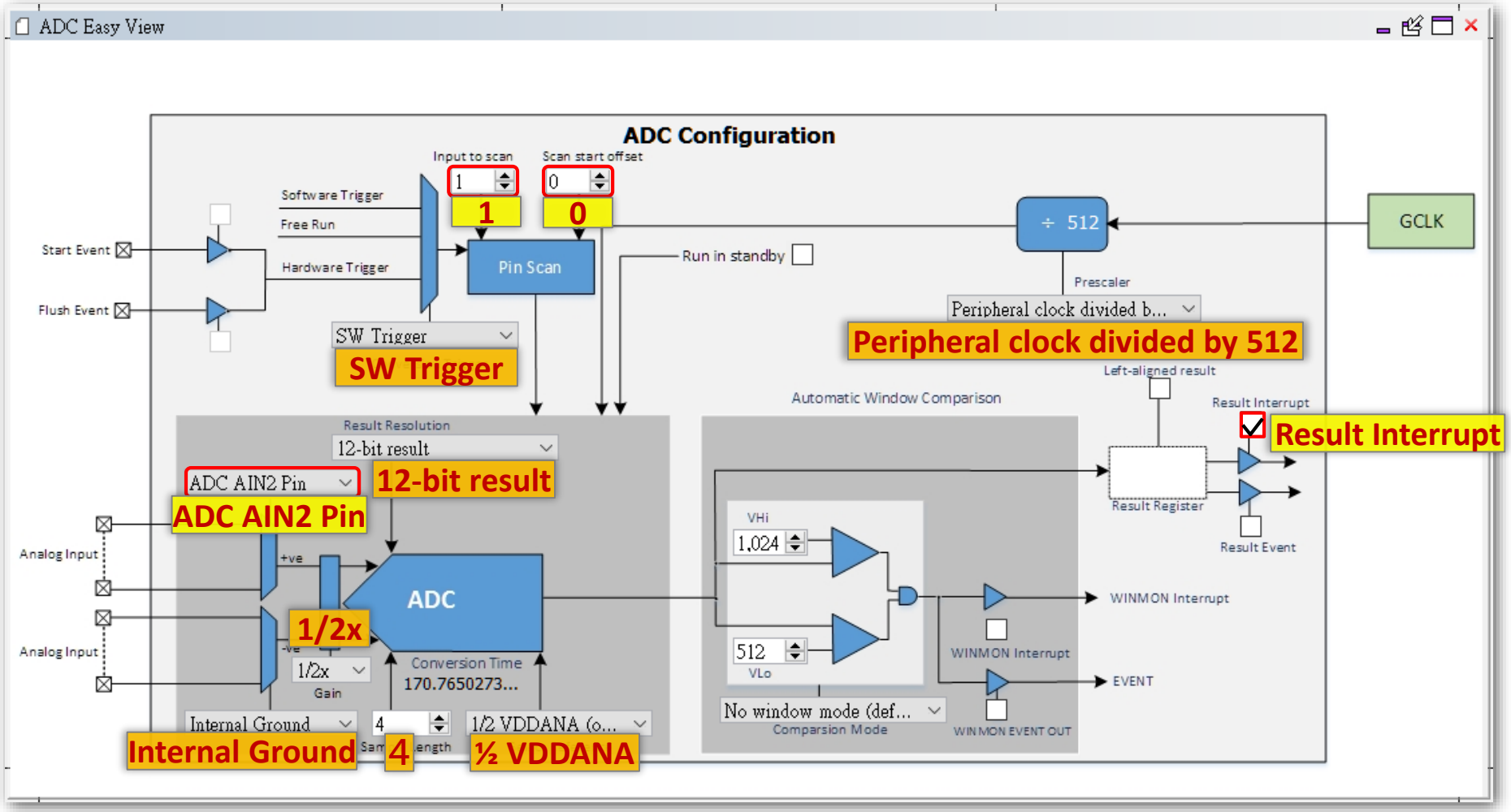
Configuration Options

- ADC
  - Select Prescaler: Peripheral clock divided by 512 — **Prescaler : divided by 512**
  - Select Sample Length (cycles): 4 — **Sample Length : 4**
  - **** Conversion Time is 170.765027322 uS ****
  - Select Gain: 1/2x — **Gain : 1/2x**
  - Select Reference: 1/2 VDDANA (only for VDDANA > 2.0V) — **V_{REF} : ½ VDDANA**
  - Select Conversion Trigger: SW Trigger — **SW Trigger**
  - Channel Configuration
    - Select Positive Input: ADC AIN2 Pin — **ADC AIN2 Pin**
    - Select Negative Input: Internal Ground — **Internal Ground**
    - Number of inputs to scan: 1 — **1**
    - Scan start offset: 0 — **0**
  - Result Configuration
    - Select Result Resolution: 12-bit result — **12-bit result**
    - Left Aligned Result: ☐
    - Enable Result Ready Interrupt: ☑ — **Check Enable Result Ready Interrupt**
    - Enable Result Ready Event Out: ☐

© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab13 ADC Pin Scan

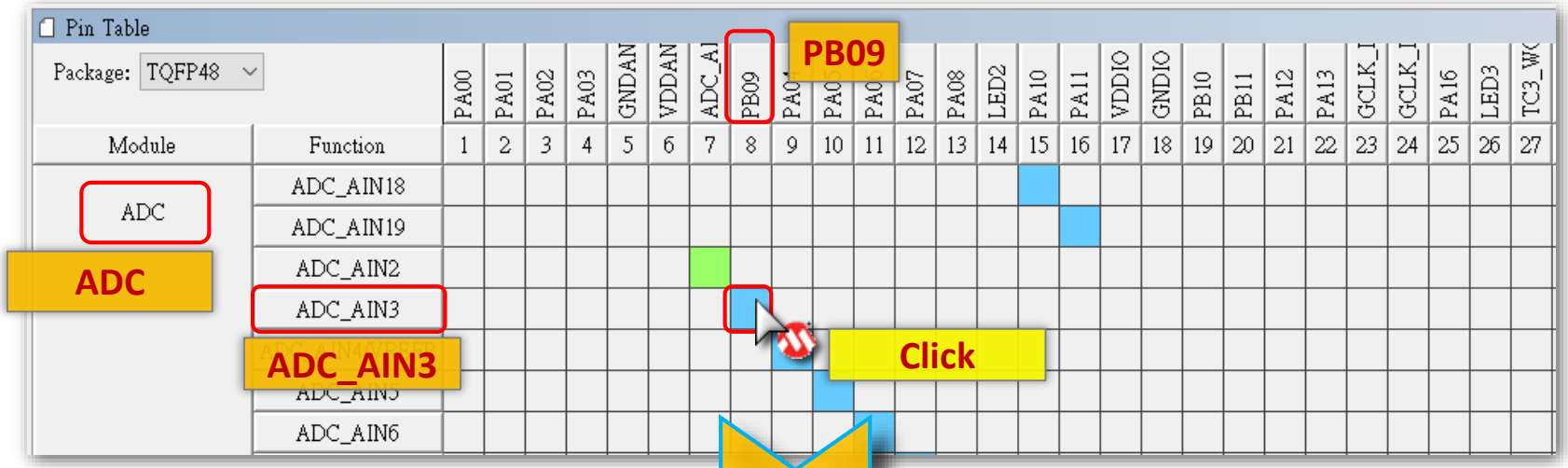
Step 1 (Use Easy View)

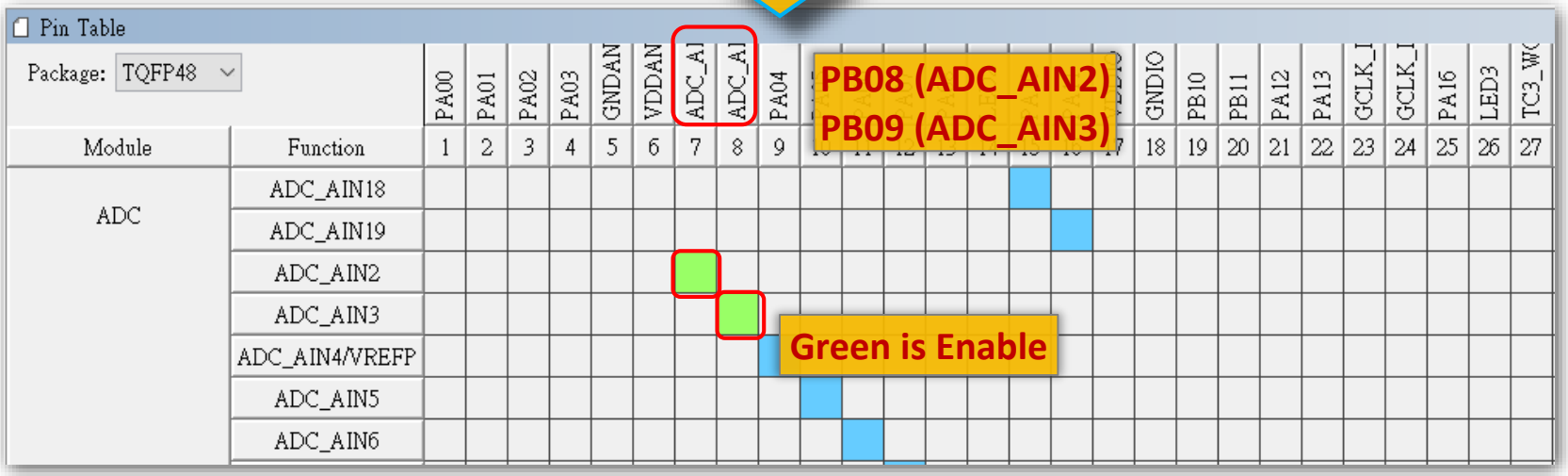


© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab13 ADC Pin Scan

Step 2



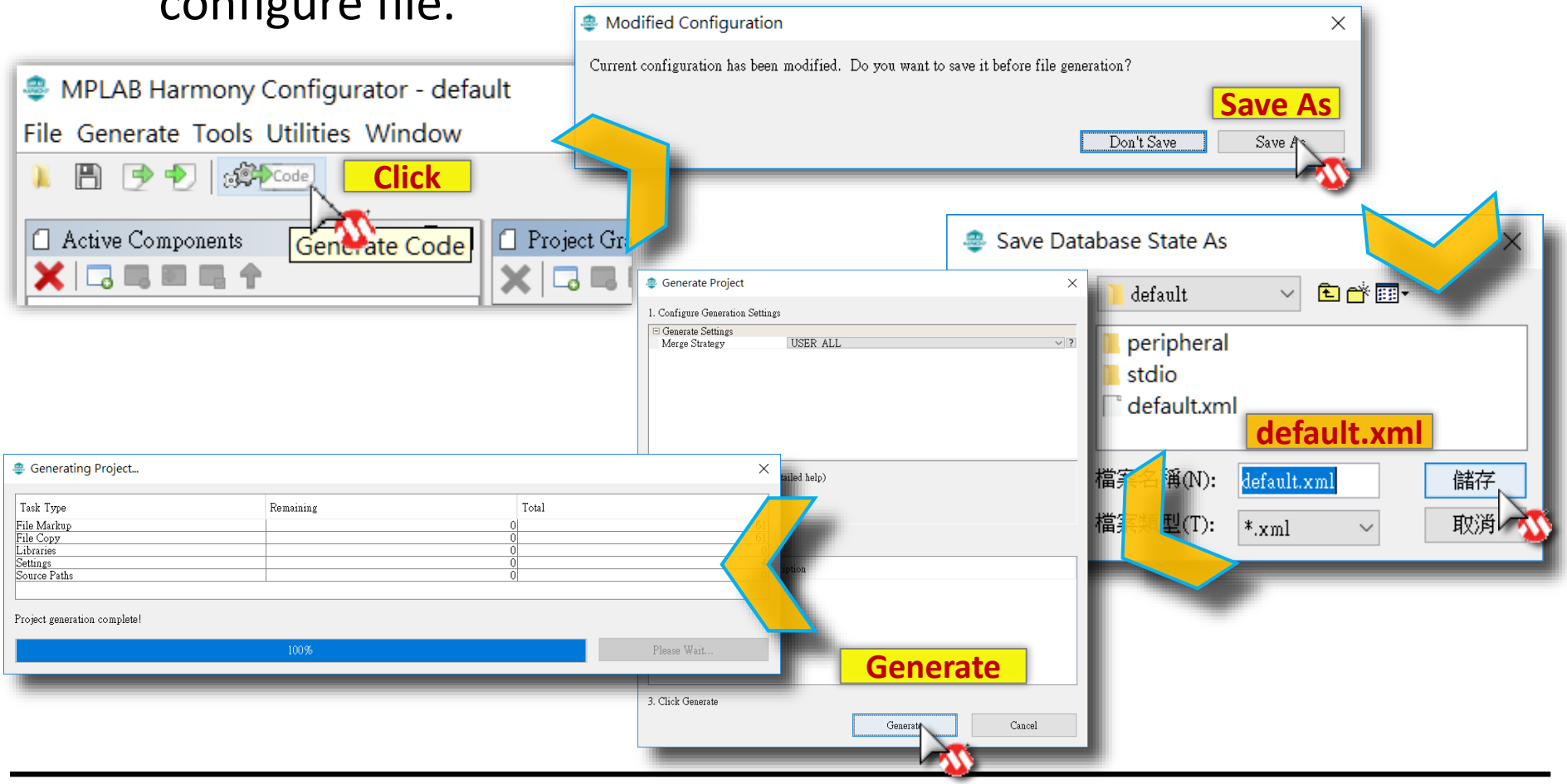


© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab13 ADC Pin Scan

Step 3

- Click to Generate Code and save changes to MHC configure file.



© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab13 ADC Pin Scan

Step 4

a Add code segment to your main.c

```
// TODO 13.01
uint16_t ADC_Result[2];
volatile uint8_t ADC_IsCompleted = 0;
volatile int8_t ADC_ChannelIdx = 0;

// TODO 13.02
void ADC_Complete( ADC_STATUS status, uintptr_t context )
{
  if( status & ADC_INTFLAG_RESRDY_Msk )
  {
    ADC_Result[ADC_ChannelIdx] = ADC_ConversionResultGet();
    ADC_ChannelIdx = ( ADC_ChannelIdx + 1 ) % 2;
    if( ADC_ChannelIdx == 0 )
    {
        ADC_IsCompleted = 1;
    }
  }
}
…

int main (void)
{  …
```

© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab13 ADC Pin Scan

Step 5

a **Add code segment to your main.c**

```
...

int main (void)
{
   ...

   // TODO 13.03
   ADC_CallbackRegister( ADC_Complete, (uintptr_t )NULL );
   ADC_Enable();

   while(1)
   {
      ...
      if( TC3_IsOverflow )
      {
         // TODO 13.04
         ADC_ConversionStart();
      }
      ...
   }
}
```

© 2019 Microchip Technology Incorporated. All Rights Reserved.

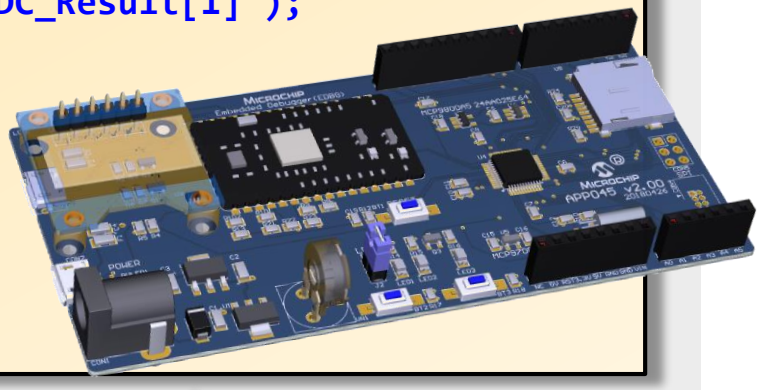
Lab13 ADC Pin Scan

Step 6

a Add code segment to your main.c

```
int main (void)
{
   …
   while(1)
   {
      …
      // TODO 13.05
      if( ADC_IsCompleted )
      {
        ADC_IsCompleted = 0;

        printf( "VR1 Value : %4d\r\n", ADC_Result[0] );
        printf( "Temperature Value : %4d\r\n", ADC_Result[1] );
      }
      …
   }
```



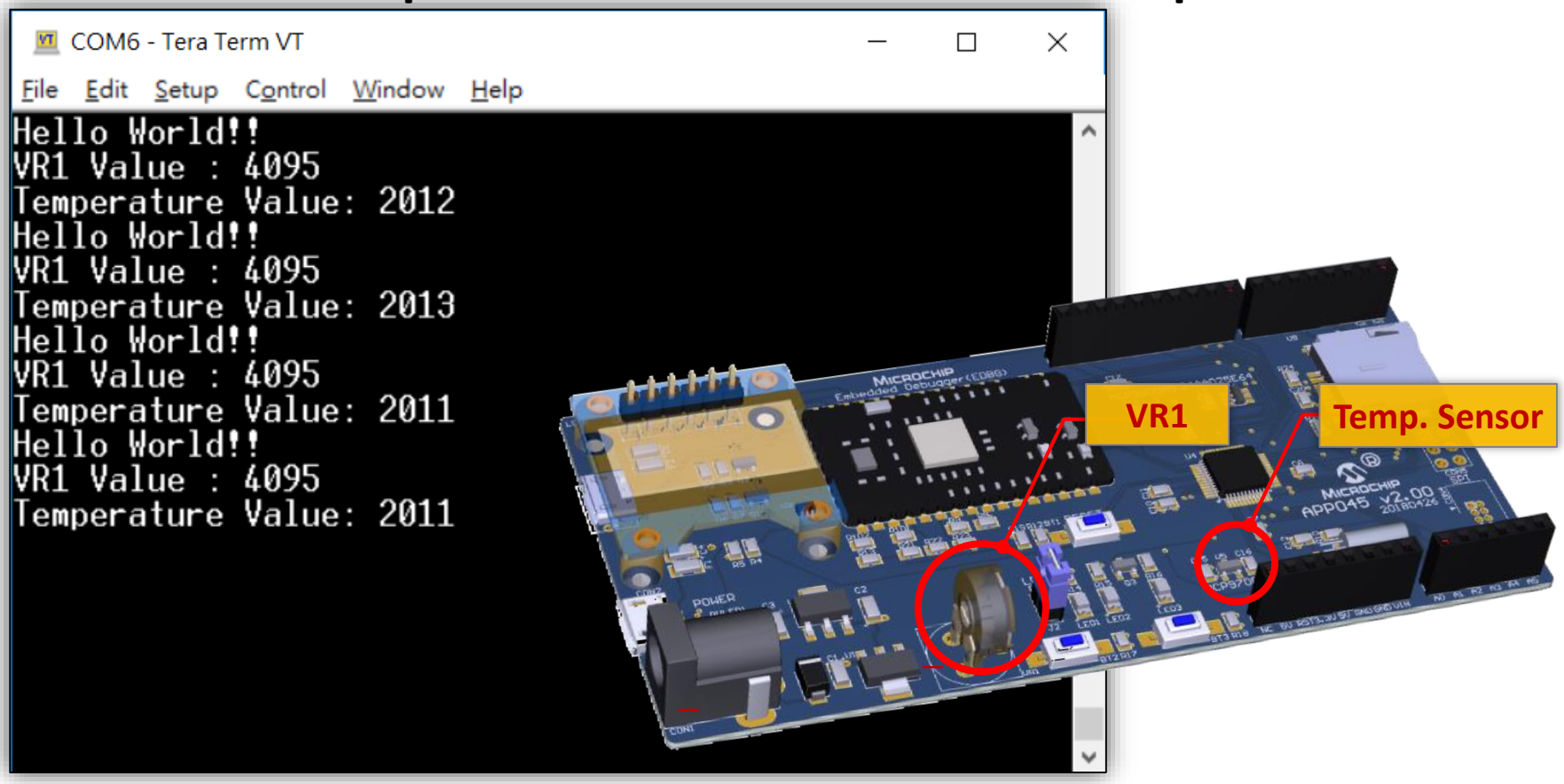
b Program firmware to target board then observe result.

© 2019 Microchip Technology Incorporated. All Rights Reserved.

Lab13 ADC Pin Scan

Result

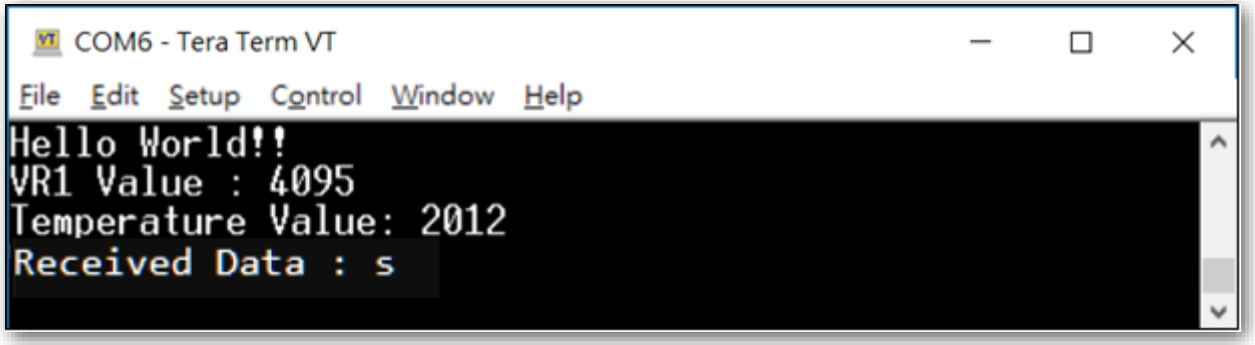
- "Hello World!!" string and VR1 / Temperature Sensor ADC value will output to terminal with one second period.



© 2019 Microchip Technology Incorporated. All Rights Reserved.

Bonus Lab

- Use VT100 command to control TeraTerm UART output as




```
printf("\033[2J");     // Clear screen
printf("\033[?25l");    // Hide cursor
printf("\033[y;xH");  // Set cursor to (x,y)
                      // Left-up corner is from (1,1)
```

- **Let's go!**

© 2019 Microchip Technology Incorporated. All Rights Reserved.