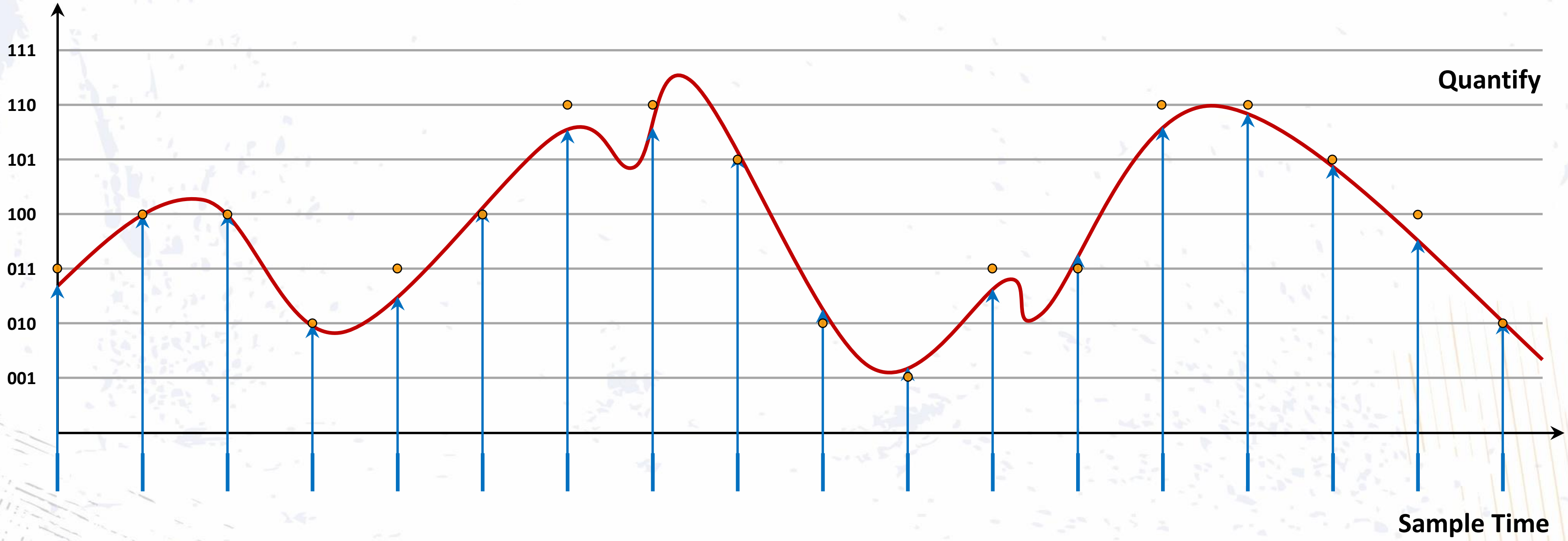
MICROCHIP

Regional Training Centers

Section 9
10 Bits High Speed ADC

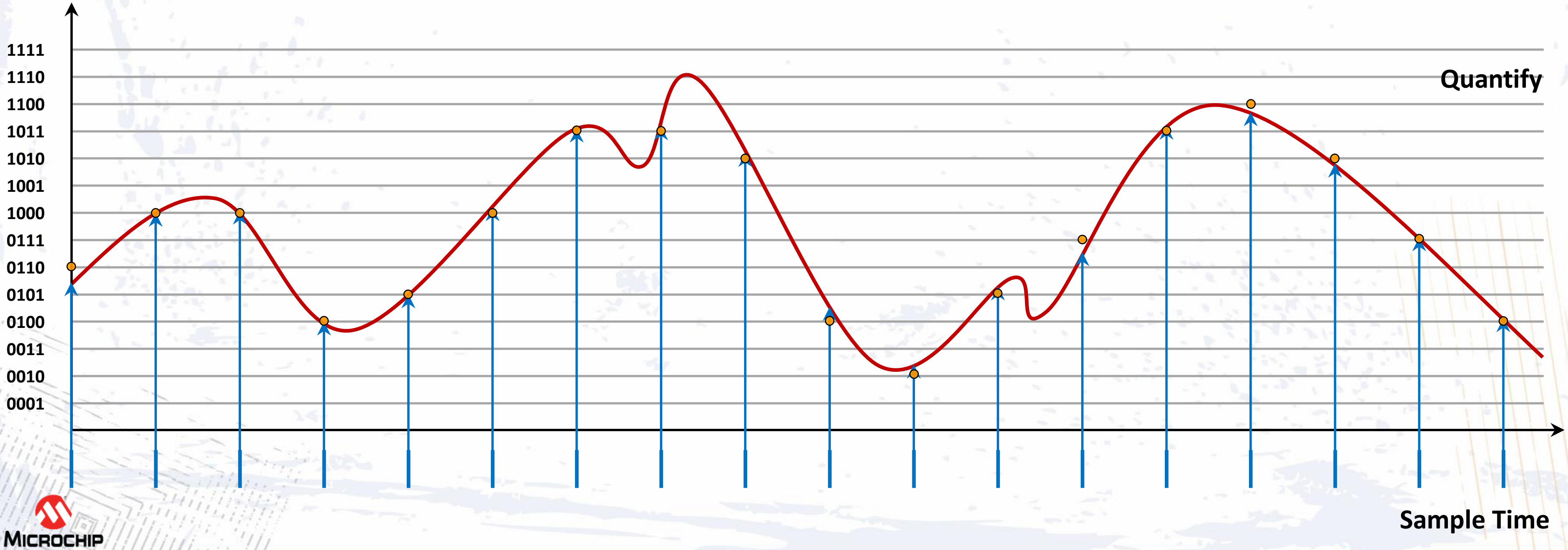
What's ADC ?

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



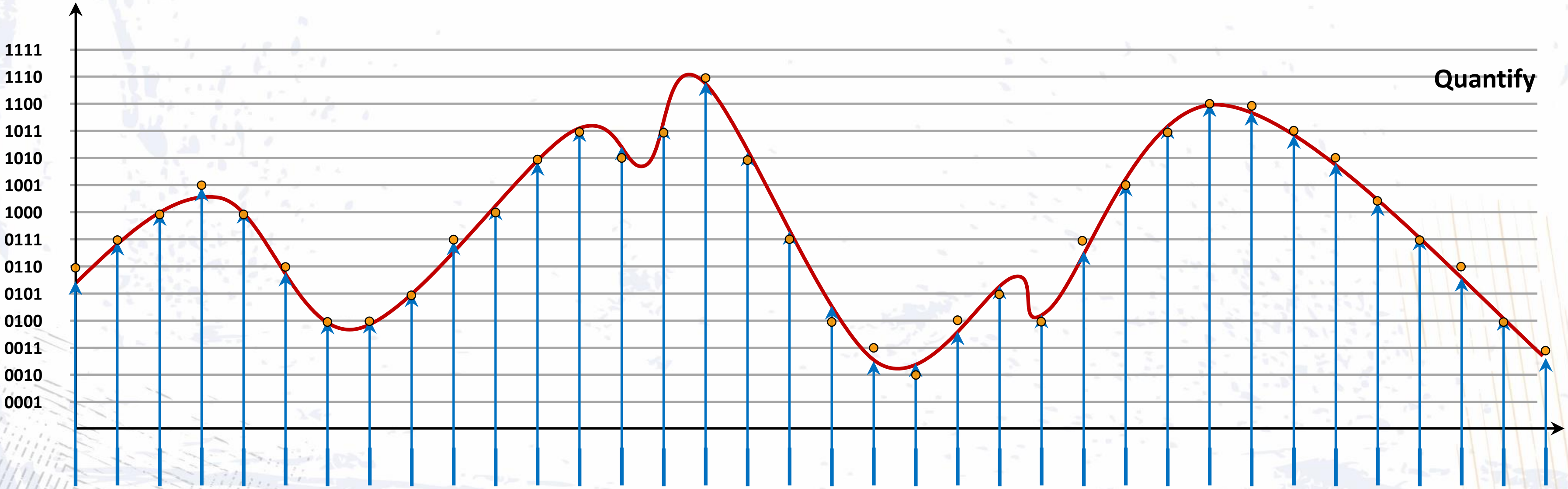
Resolution

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



Sample Rate

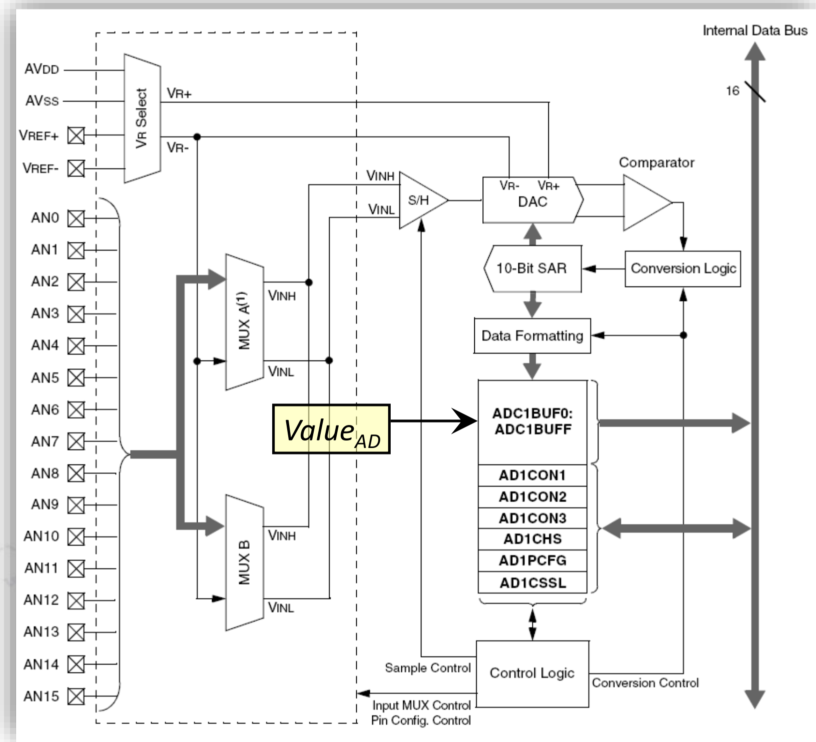
- The analog signal is required to define the rate at which new digital values are sampled from the analog signal.
- This faithful reproduction is only possible if the sampling rate is higher than twice the highest frequency of the signal. (wiki)



PIC24F ADC Architecture

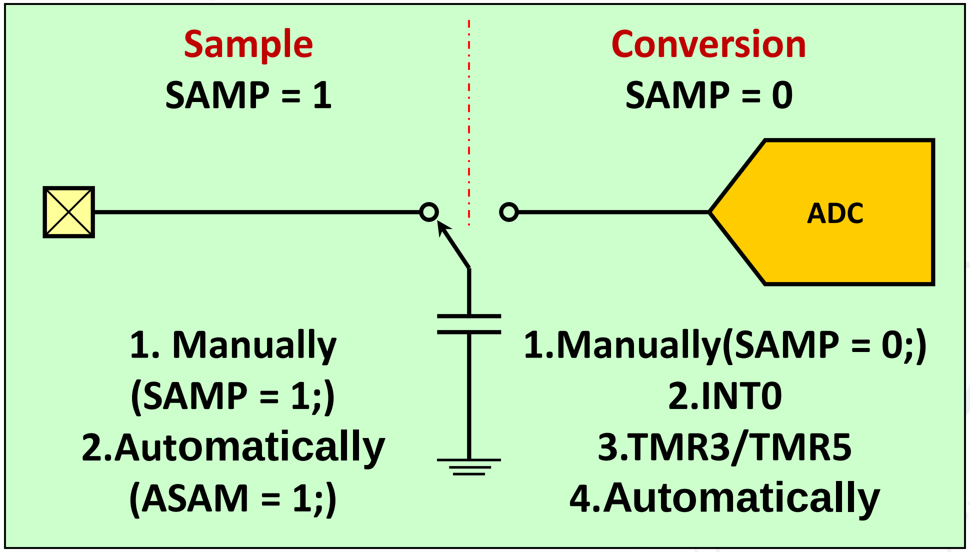
- Successive Approximation (SAR) conversion, Conversion speeds of up to 500 ksps. Up to 16 analog input, 10 Bits resolution.
- Automatic Channel Scan, Selectable trigger source
- 16 result buffer, Selectable Buffer Fill modes, multiplexer Alternate.
- External voltage reference input pins.

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



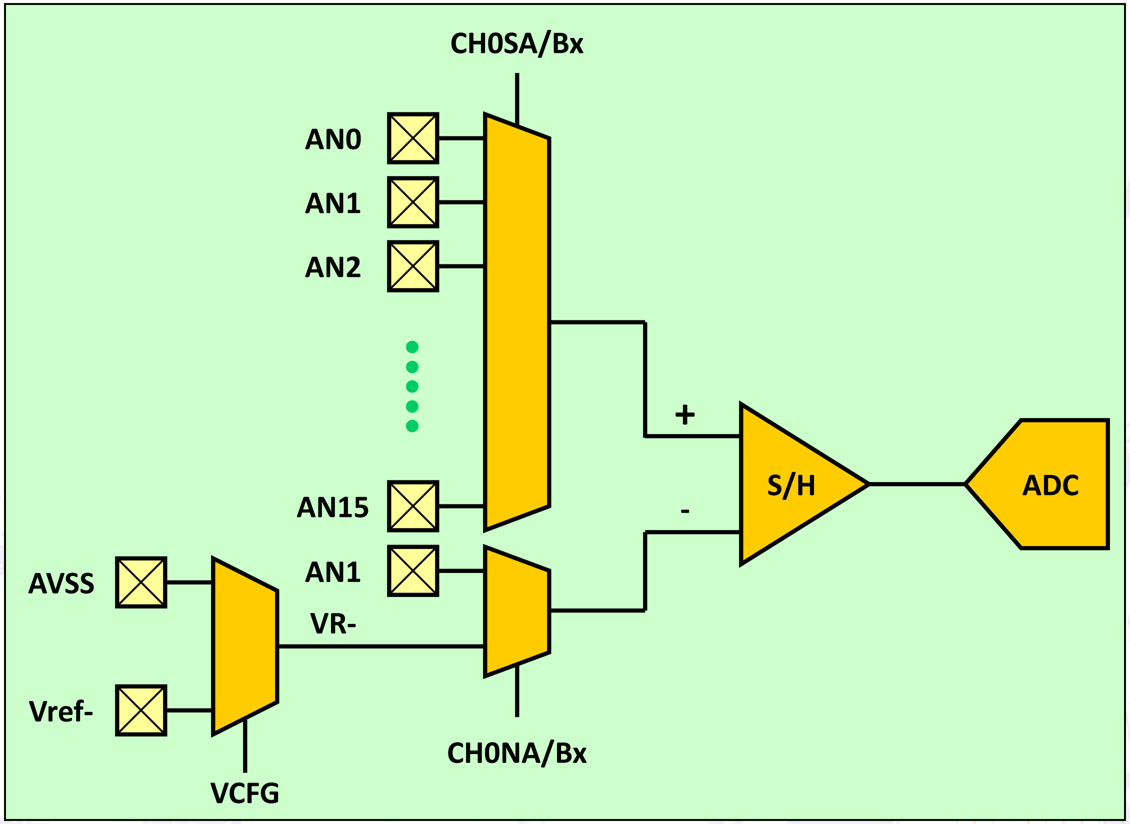
ADC Sample, Conv. Trigger Source

- A complete ADC conversion include sample and conversion state.
- There are two sample trigger source: Manually & Automatically.
- There are four conversion trigger source: Manually, External interrupt, Timer(TMR3/5) & Automatically.
- Sample state must go first before conversion state.



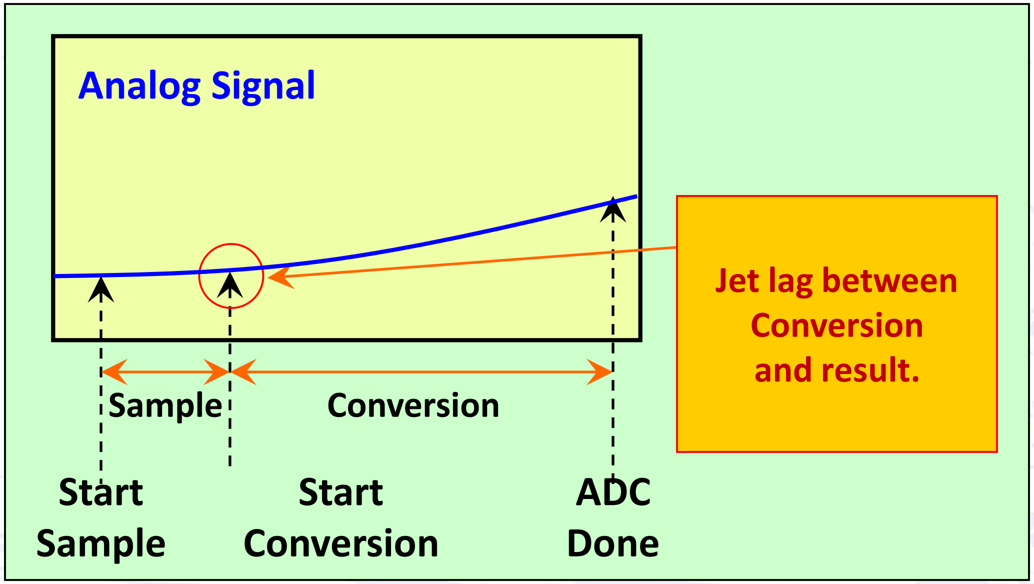
ADC Channel Selection & Gain

- Any one of up to 16 analog inputs to connect to the positive input of the S/H.
- Two options V_{R-} or AN1 to connect to the negative input of the S/H.
- The real input signal is positive – negative.



Sample and Conversion Sequence

- The ADC sample and conversion time are requirements from the specification.
- The sample time must greater than 1 T_{AD} and conversion time need 12 T_{AD}. The maximize ADC sample rate is 500Ksps. (T_{AD} must greater than 75nS)



Jet lag between Conversion and result.

MCC’s ADC Function

- MCC Provide below common functions:

  void ADC1_Initialize(void)

  // Initial ADC.

  void ADC1_ChannelSelect(ADC1_CHANNEL channel)

  // Channel Selection.

  void ADC1_CallBack(void)

  // Callback function.

  void ADC1_Tasks(void)

  // Status maintain function, must execute at main loop or ISR.

MCC's ADC Function

- MCC Provide below common functions:

```
void ADC1_Start(void);
        // Manually trigger sample state ( SAMP <- 1 ).
void ADC1_Stop(void);
        // Manually trigger conversion state ( SAMP <- 0 ).
uint16_t ADC1_ConversionResultGet(void);
        // Read ADC conversion result (get ADC1BUF0)
uint16_t ADC1_ConversionResultBufferGet(uint16_t *buffer);
        // Read ADC conversion results (get ADC1BUF0 ~ ADC1BUFn -> buffer[ ])
```

Lab9 ADC Single CH Manually

Lab9 ADC Single CH Automatically

- Try to initial ADC1 module, than use ADC1 to conversion Potentiometer(VR) voltage.
- Show ADC1 result to Character LCD Module.
- ADC1 module set to 10 Bits, Auto trigger sample & conversion mode.
- ADC1 result format is 16 Bits, unsigned, Decimal.
- Please connect RB4(AN4) to Potentiometer(VR1) to get analog voltage of potentiometer.

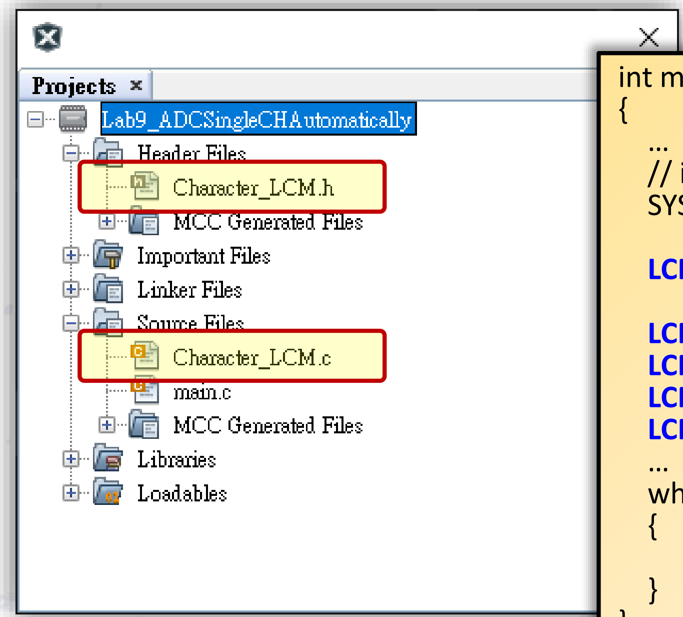
- **Let’s go!**

Lab9 Hints

- You can use character LCD module libraries to driver your LCD. The Libraries is ready. (Character_LCM.c, Character_LCM.h)
- LCD libraries provide follow common function:
  void LCM_Init( ) // Initial LCD Module.
  int LCM_IsBusy( ) //Read LCD Module BUSY status. (1:Busy,0:Not Busy)
  void LCM_SetCursor(char X , char Y) //Setting cursor position.
  void LCM_PutASCII(unsigned char) //Put ASCII to cursor.
  void LCM_PutROMString(const unsigned char *String)
  //Put const string to cursor.
  void LCM_PutRAMString(unsigned char *String) //Put string to cursor.
  void LCM_PutHex(unsigned char Hex) //Put HEX to cursor.
  void LCM_PutNumber(unsigned int Number , unsigned char Digit);
  //Put number to cursor.

Lab9 Hints

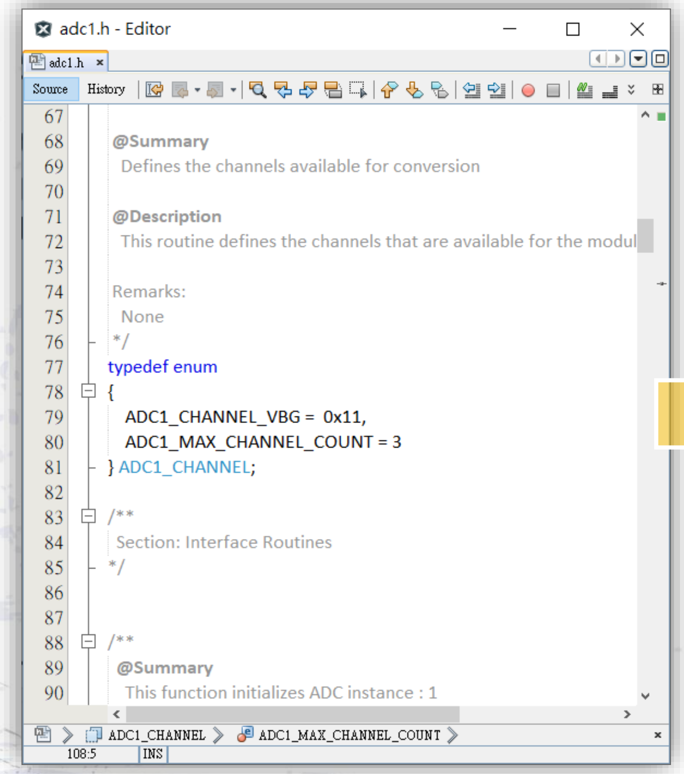
- **You need add LCD control libraries to your project before invoke it.**
- **Notice ! Invoke void LCM_Init( ) first, before other LCD's function.**



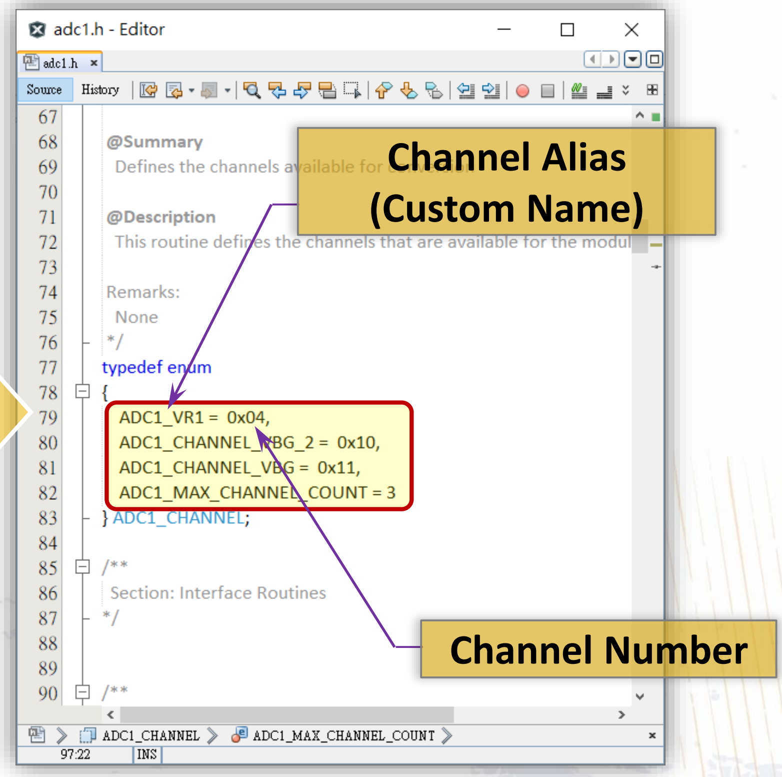
```
int main(void)
{
    …
    // initialize the device
    SYSTEM_Initialize();

    LCM_Init();

    LCM_SetCursor(0, 0);
    LCM_PutROMString(“Hello World !! ");
    LCM_SetCursor(0, 1);
    LCM_PutNumber(1234,4);
    …
    while (1)
    {
        …
    }
}
```

MCC’s bug for ADC for v3.66

- **ADC channel definition can’t be generated correctly at v3.66. You must modify adc1.h after generate code.**



```

  @Summary
    Defines the channels available for conversion

  @Description
    This routine defines the channels that are available for the modul

  Remarks:
    None
 */
typedef enum
{
    ADC1_CHANNEL_VBG =  0x11,
    ADC1_MAX_CHANNEL_COUNT = 3
} ADC1_CHANNEL;

/**
 Section: Interface Routines
 */


/**
  @Summary
    This function initializes ADC instance : 1
```



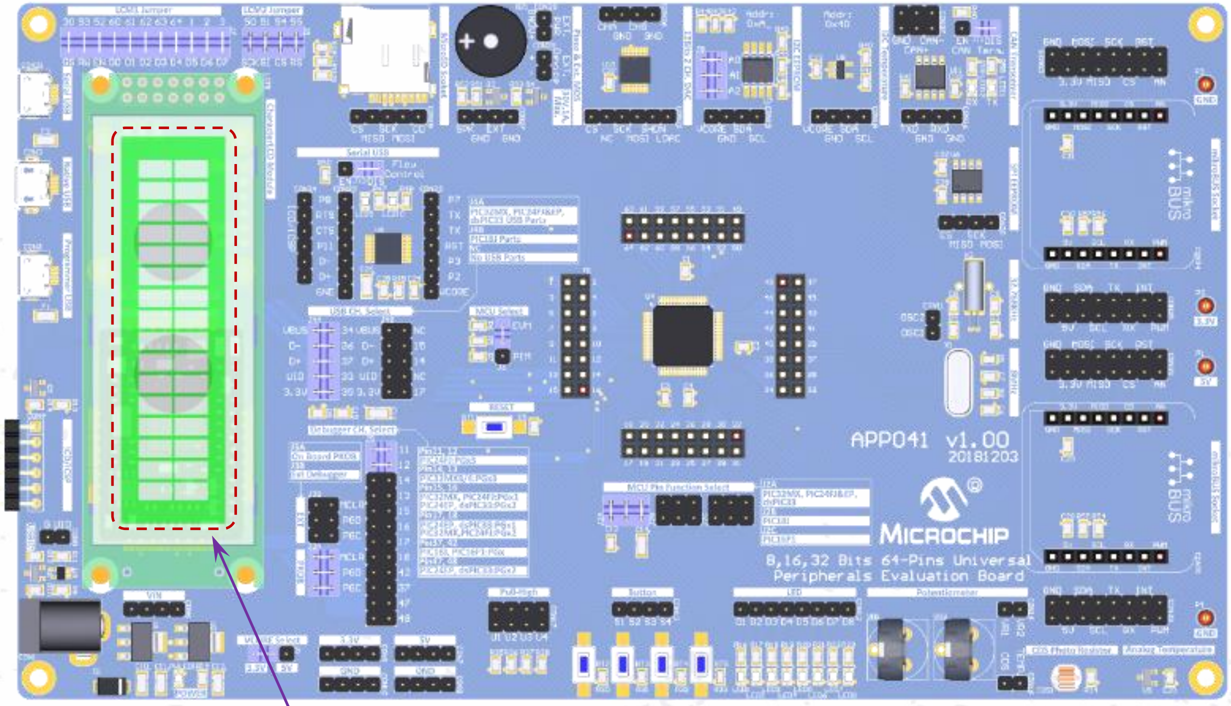
```

  @Summary
    Defines the channels a

  @Description
    This routine defines the channels that are available for the modul

  Remarks:
    None
 */
typedef enum
{
    ADC1_VR1 =  0x04,
    ADC1_CHANNEL_VBG_2 =  0x10,
    ADC1_CHANNEL_VBG =  0x11,
    ADC1_MAX_CHANNEL_COUNT = 3
} ADC1_CHANNEL;

/**
 Section: Interface Routines
 */


/**
```

Channel Alias (Custom Name)

Channel Number

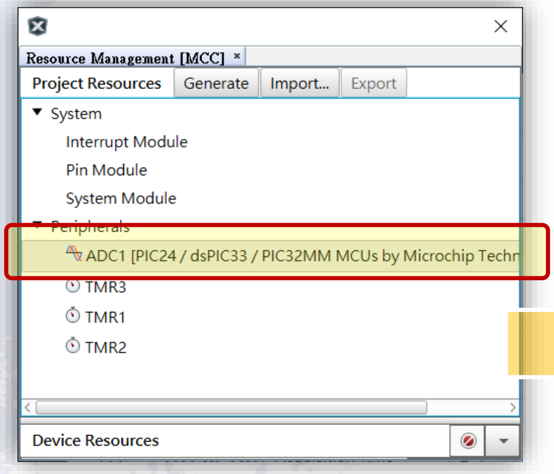
Lab9 ADC Single CH Automatically

Result

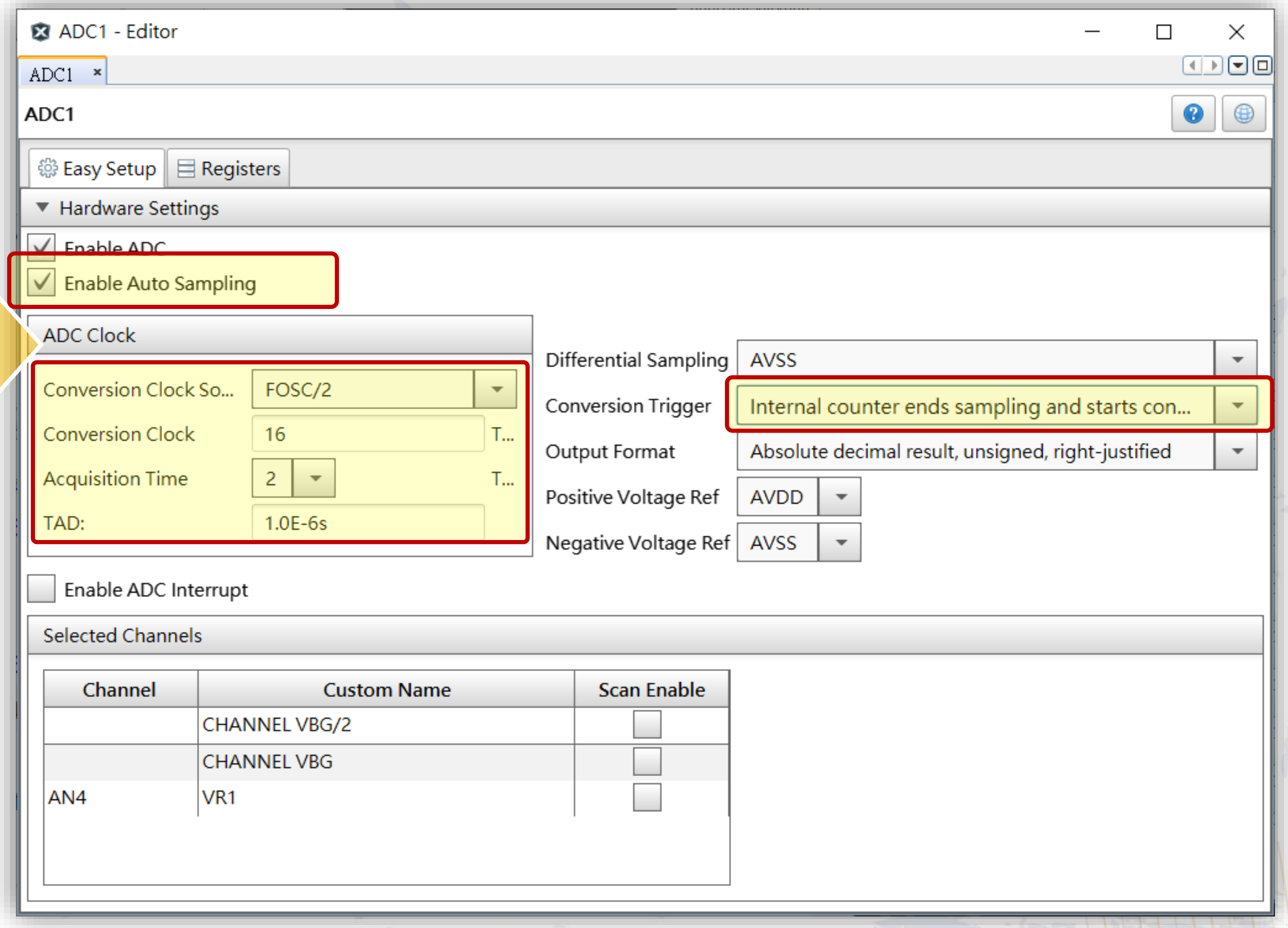


VR1 Show at LCD Module.
Trigger by Automatically

Lab9 ADC Single CH Automatically MCC's Setting & Code Example



- System
  - Interrupt Module
  - Pin Module
  - System Module
- Peripherals
  - ADC1 [PIC24 / dsPIC33 / PIC32MM MCUs by Microchip Techn
  - TMR3
  - TMR1
  - TMR2



☑ Enable ADC

☑ Enable Auto Sampling

ADC Clock

Conversion Clock So... FOSC/2

Conversion Clock 16 T...

Acquisition Time 2 T...

TAD: 1.0E-6s

Differential Sampling AVSS

Conversion Trigger Internal counter ends sampling and starts con...

Output Format Absolute decimal result, unsigned, right-justified

Positive Voltage Ref AVDD

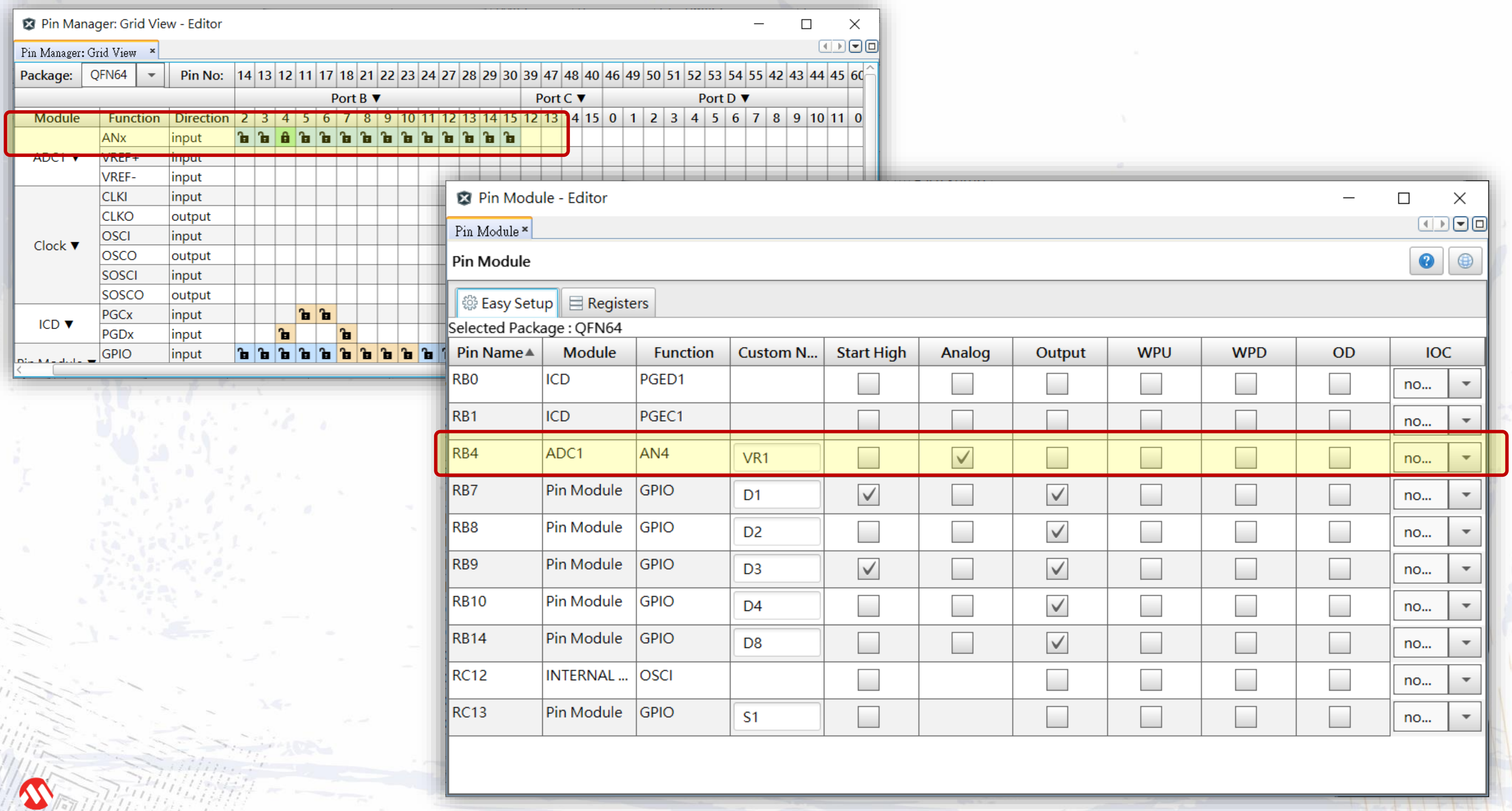
Negative Voltage Ref AVSS

☐ Enable ADC Interrupt

Selected Channels

Channel	Custom Name	Scan Enable
	CHANNEL VBG/2	☐
	CHANNEL VBG	☐
AN4	VR1	☐

Lab9 ADC Single CH Automatically MCC's Setting & Code Example



Module	Function	Direction
ADC1 ▼	ANx	input
	VREF+	input
	VREF-	input
Clock ▼	CLKI	input
	CLKO	output
	OSCI	input
	OSCO	output
	SOSCI	input
	SOSCO	output
ICD ▼	PGCx	input
	PGDx	input
Pin Module ▼	GPIO	input

Selected Package : QFN64

Pin Name▲	Module	Function	Custom N...	Start High	Analog	Output	WPU	WPD	OD	IOC
RB0	ICD	PGED1		☐	☐	☐	☐	☐	☐	no...
RB1	ICD	PGEC1		☐	☐	☐	☐	☐	☐	no...
RB4	ADC1	AN4	VR1	☐	☑	☐	☐	☐	☐	no...
RB7	Pin Module	GPIO	D1	☑	☐	☑	☐	☐	☐	no...
RB8	Pin Module	GPIO	D2	☐	☐	☑	☐	☐	☐	no...
RB9	Pin Module	GPIO	D3	☑	☐	☑	☐	☐	☐	no...
RB10	Pin Module	GPIO	D4	☐	☐	☑	☐	☐	☐	no...
RB14	Pin Module	GPIO	D8	☐	☐	☑	☐	☐	☐	no...
RC12	INTERNAL ...	OSCI		☐		☐	☐	☐	☐	no...
RC13	Pin Module	GPIO	S1	☐		☐	☐	☐	☐	no...

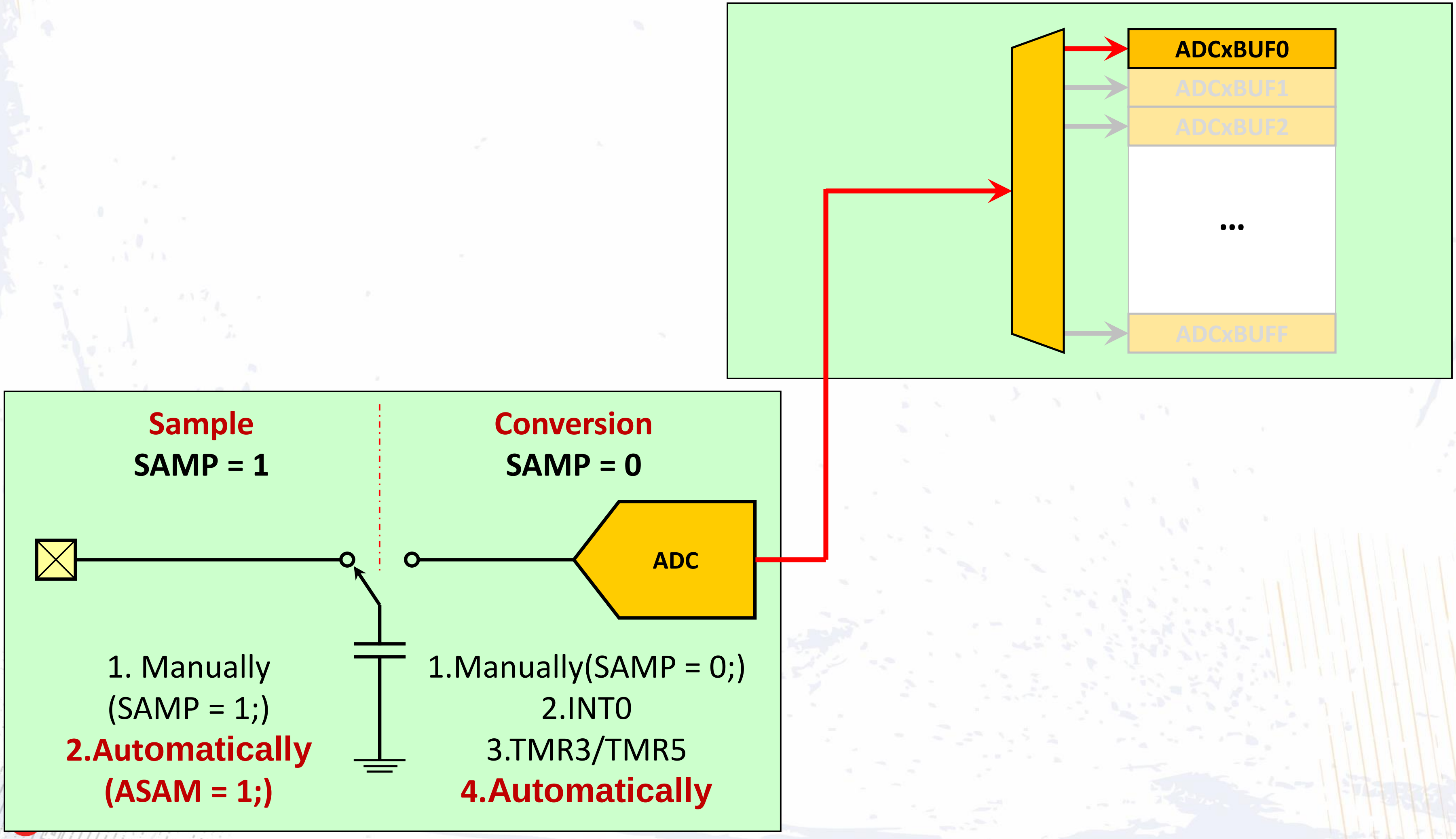
Lab9 ADC Single CH Automatically MCC's Setting & Code Example

```
volatile unsigned int ADCResult;
int main(void)
{
    …
    ADC1_ChannelSelect(ADC1_VR1);
    while (1)
    {
        …
        ADC1_Tasks();
        …
    }
}

void ADC1_CallBack(void)
{
    ADCResult = ADC1_ConversionResultGet();

    LCM_SetCursor(4, 1);
    LCM_PutNumber(ADCResult, 4);
}
```

```
typedef enum
{
    ADC1_VR1 =  0x04,
    ADC1_CHANNEL_VBG_2 =  0x10,
    ADC1_CHANNEL_VBG =  0x11,
    ADC1_MAX_CHANNEL_COUNT = 3
} ADC1_CHANNEL;
```

Lab9 ADC Single CH Automatically

Block Diagram



ADCxBUF0
ADCxBUF1
ADCxBUF2
…
ADCxBUFF

Sample
SAMP = 1

Conversion
SAMP = 0

ADC

1. Manually
(SAMP = 1;)
2.Automatically
(ASAM = 1;)

1.Manually(SAMP = 0;)
2.INT0
3.TMR3/TMR5
4.Automatically

Lab10 ADC Single CH Timer Trigger

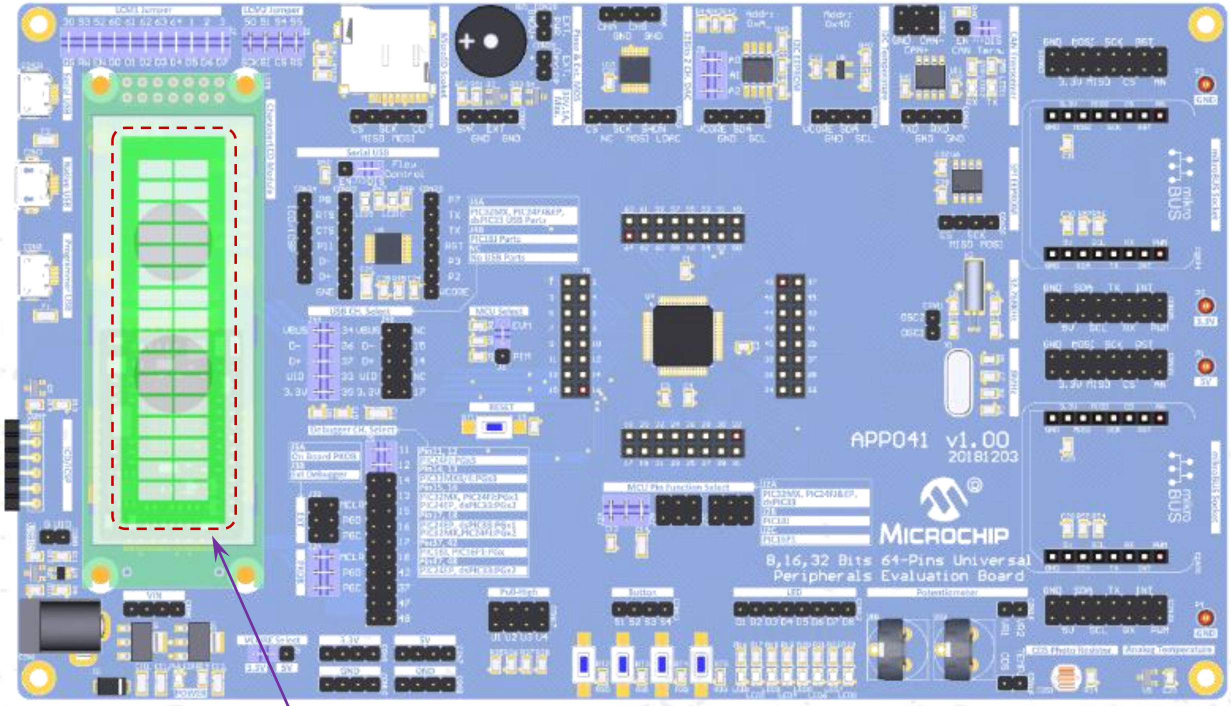
Lab10 ADC Single CH Timer Trigger

- Try to change conversion trigger source to timer 3.
- Timer 3 period set to 50mS.
- ADC1 Settings same as Lab9.

- **Let's go!**

Lab10 ADC Single CH Timer Trigger

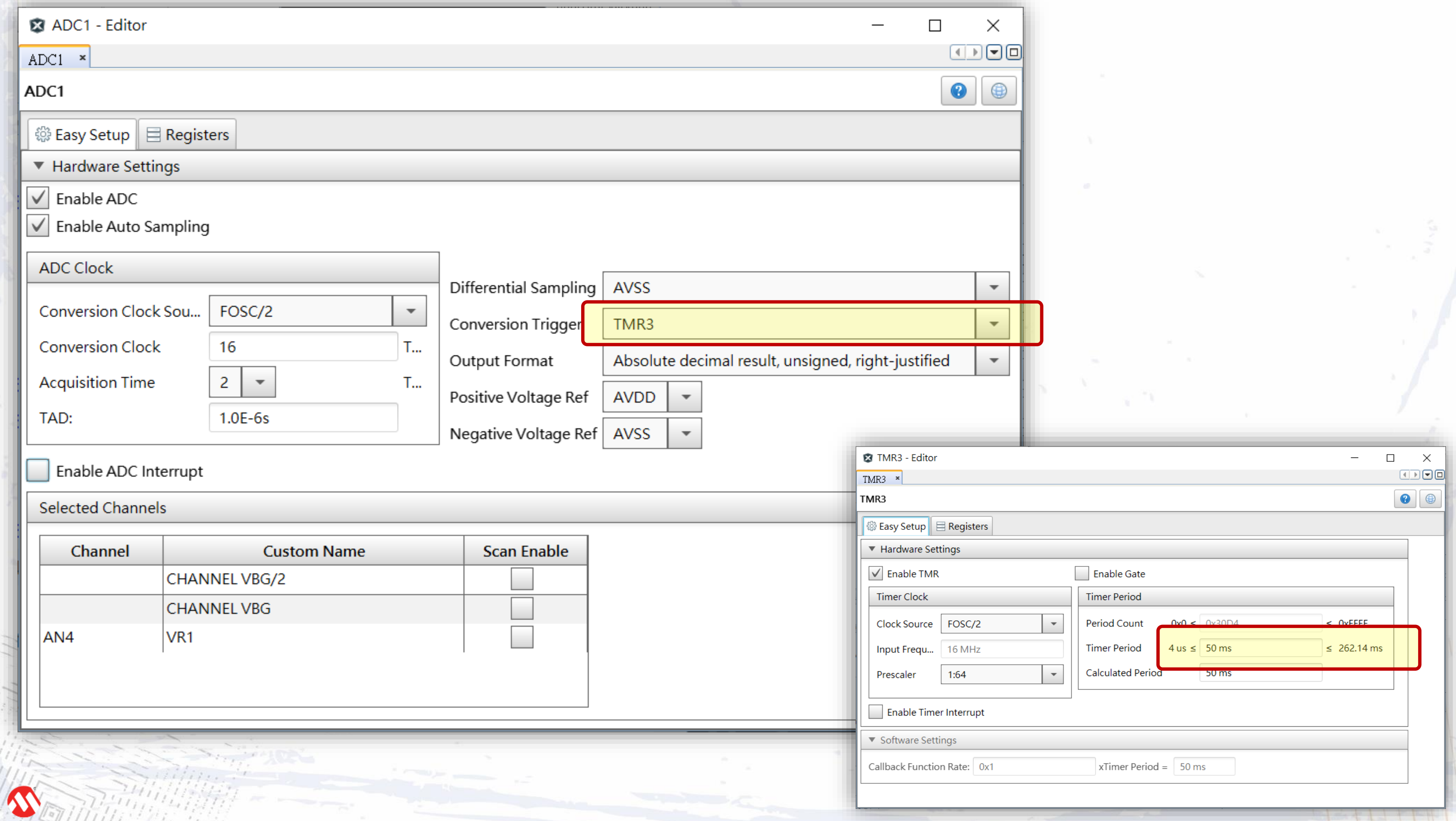
Result



VR1 Show at LCD Module.
Trigger by TMR3

Lab10 ADC Single CH Timer Trigger

MCC’s Setting & Code Example



ADC1 - Editor

ADC1

Easy Setup | Registers

Hardware Settings

☑ Enable ADC

☑ Enable Auto Sampling

ADC Clock

Conversion Clock Sou...: FOSC/2

Conversion Clock: 16 T...

Acquisition Time: 2 T...

TAD: 1.0E-6s

Differential Sampling: AVSS

Conversion Trigger: TMR3

Output Format: Absolute decimal result, unsigned, right-justified

Positive Voltage Ref: AVDD

Negative Voltage Ref: AVSS

☐ Enable ADC Interrupt

Selected Channels

Channel	Custom Name	Scan Enable
	CHANNEL VBG/2	☐
	CHANNEL VBG	☐
AN4	VR1	☐

TMR3 - Editor

TMR3

Easy Setup | Registers

Hardware Settings

☑ Enable TMR

☐ Enable Gate

Timer Clock

Clock Source: FOSC/2

Input Frequ...: 16 MHz

Prescaler: 1:64

Timer Period

Period Count: 0x0 ≤ 0x30D4 ≤ 0xFFFF

Timer Period: 4 us ≤ 50 ms ≤ 262.14 ms

Calculated Period: 50 ms

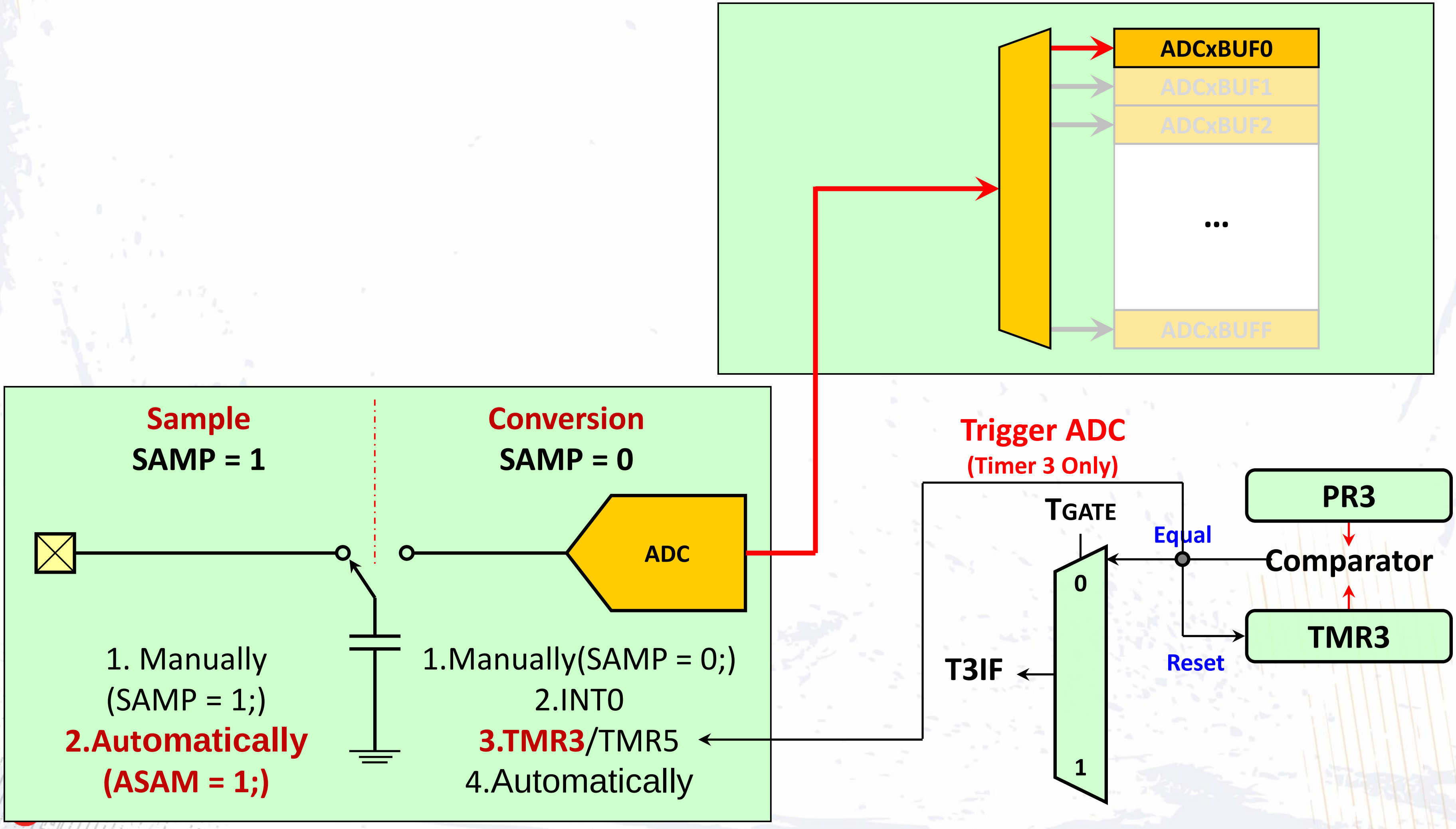
☐ Enable Timer Interrupt

Software Settings

Callback Function Rate: 0x1 xTimer Period = 50 ms

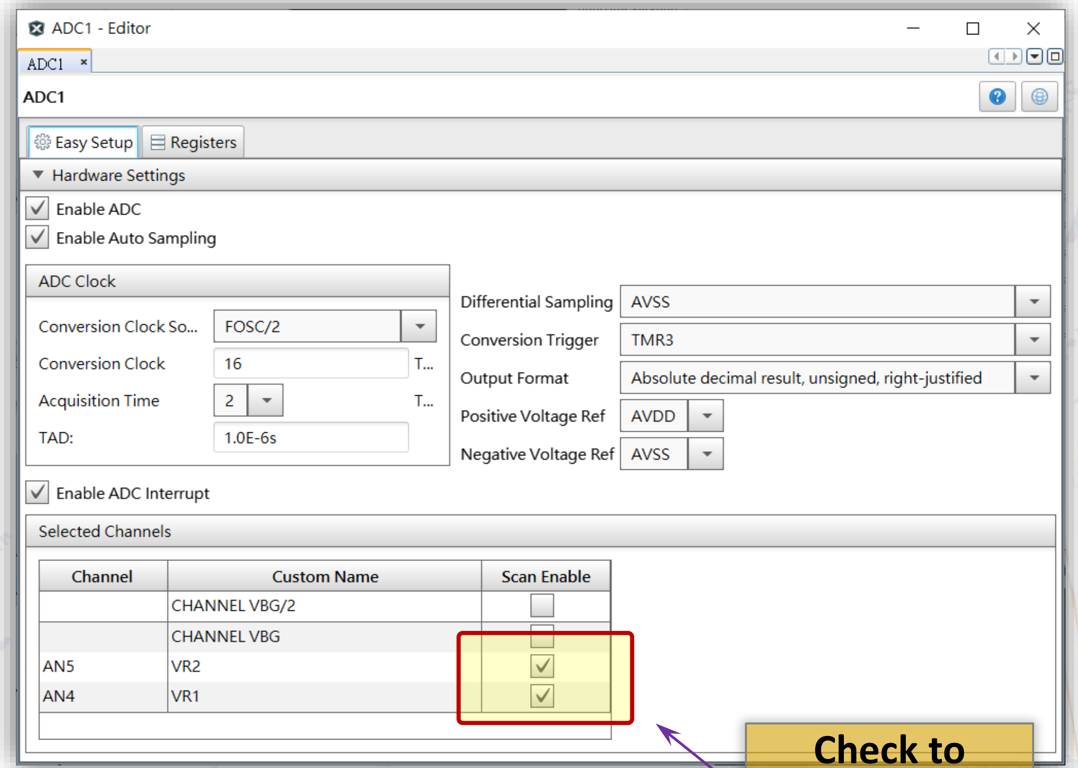
Lab10 ADC Single CH Timer Trigger

Block Diagram



Automatic Channel Scan

- Multiplexer A support channel scan function.
- Channel switch at each conversion done.
- Inputs are always scanned from lower to higher numbered inputs, starting at the first selected channel after each interrupt occurs.



☑ Enable ADC

☑ Enable Auto Sampling

ADC Clock

Conversion Clock So...: FOSC/2

Conversion Clock: 16 T...

Acquisition Time: 2 T...

TAD: 1.0E-6s

Differential Sampling: AVSS

Conversion Trigger: TMR3

Output Format: Absolute decimal result, unsigned, right-justified

Positive Voltage Ref: AVDD

Negative Voltage Ref: AVSS

☑ Enable ADC Interrupt

Selected Channels

Channel	Custom Name	Scan Enable
	CHANNEL VBG/2	☐
	CHANNEL VBG	☐
AN5	VR2	☑
AN4	VR1	☑

Check to Enable Scan

Lab11 ADC Scan with Interrupt

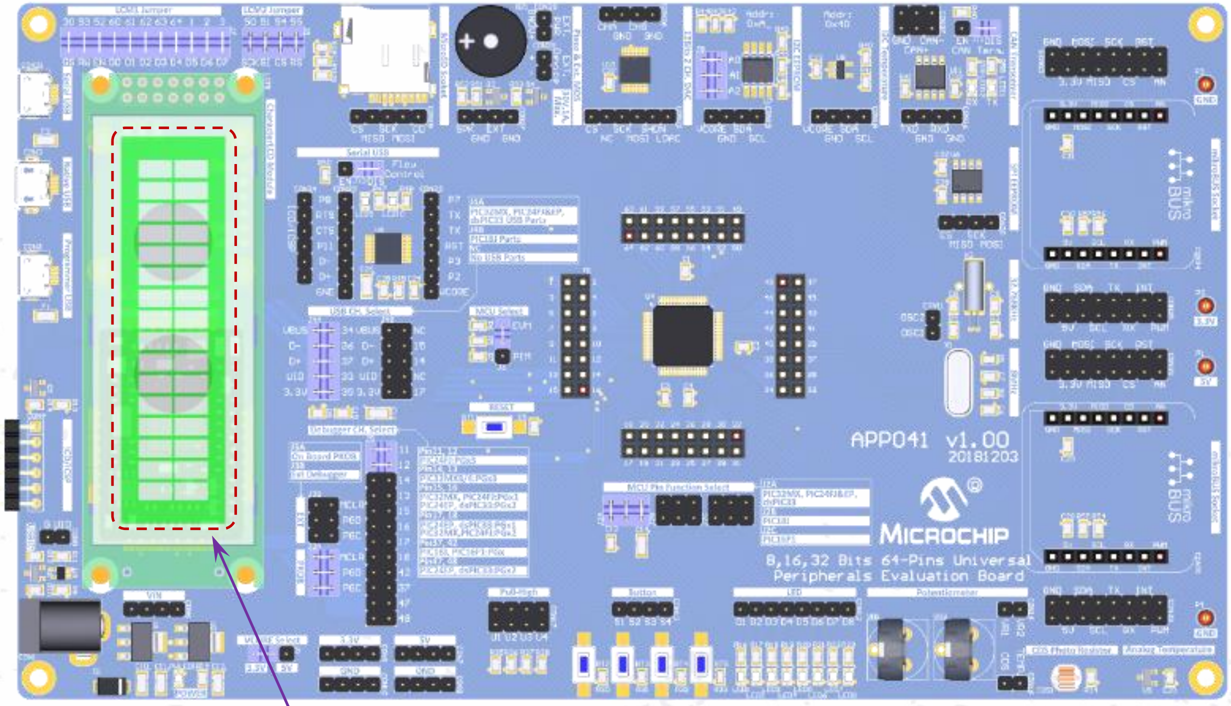
Lab11 ADC Scan with Interrupt

- **Try to enable channel scan function to conversion more analog signal by ADC1.**
- **Change ADC1 code style from polling to interrupt.**
- **ADC1 interrupt priority set to 5.**
- **Please connect RB5(AN5) to Potentiometer(VR2) to get analog voltage of potentiometer, also.**

- **Let's go!**

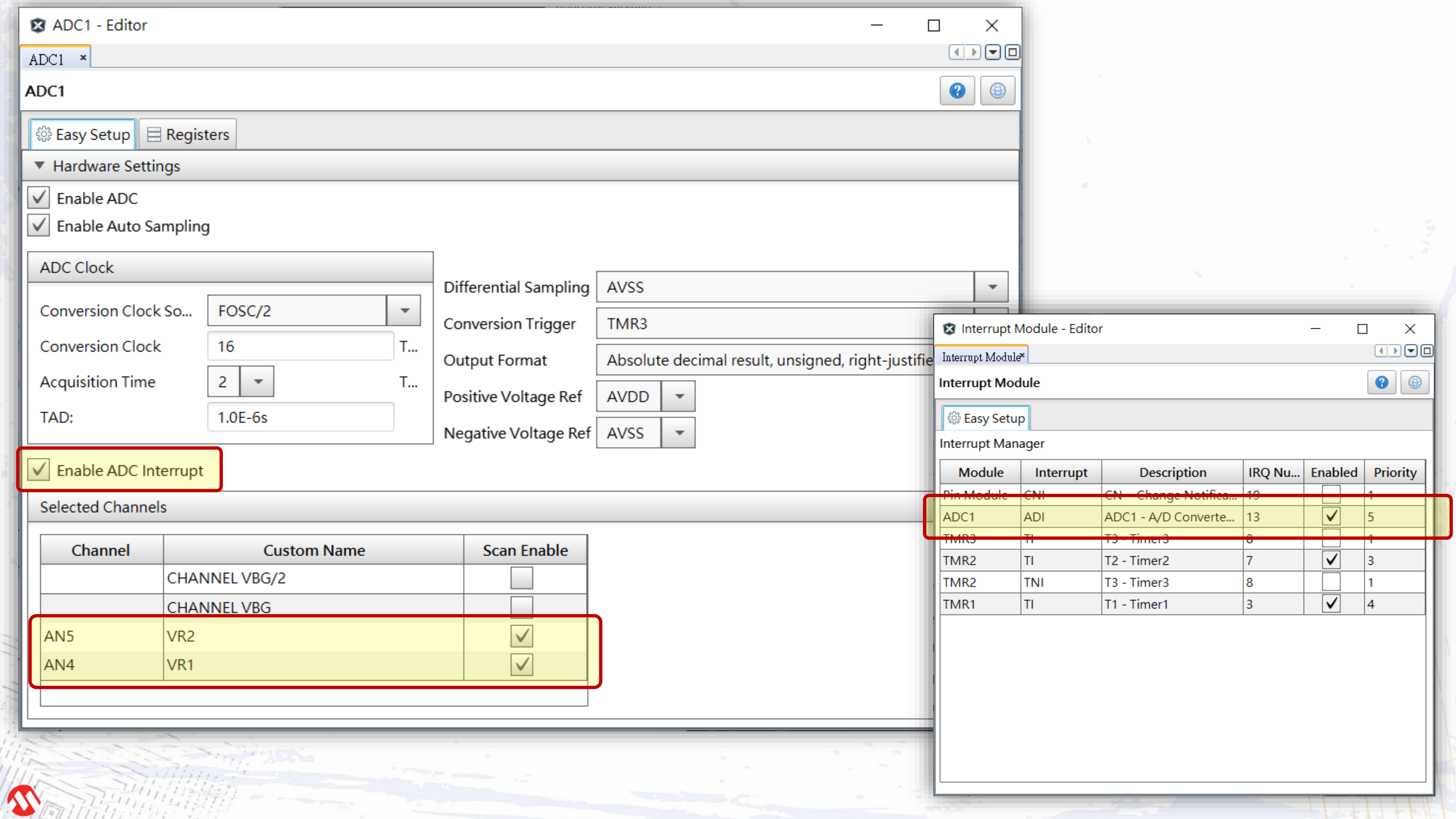
Lab11 ADC Scan with Interrupt

Result



**VR1 & VR2 Show at
LCD Module, respectively**

Lab11 ADC Scan with Interrupt MCC's Setting & Code Example



Hardware Settings

☑ Enable ADC

☑ Enable Auto Sampling

ADC Clock

Conversion Clock So...	FOSC/2	
Conversion Clock	16	T...
Acquisition Time	2	T...
TAD:	1.0E-6s	

Differential Sampling: AVSS

Conversion Trigger: TMR3

Output Format: Absolute decimal result, unsigned, right-justifie

Positive Voltage Ref: AVDD

Negative Voltage Ref: AVSS

☑ Enable ADC Interrupt

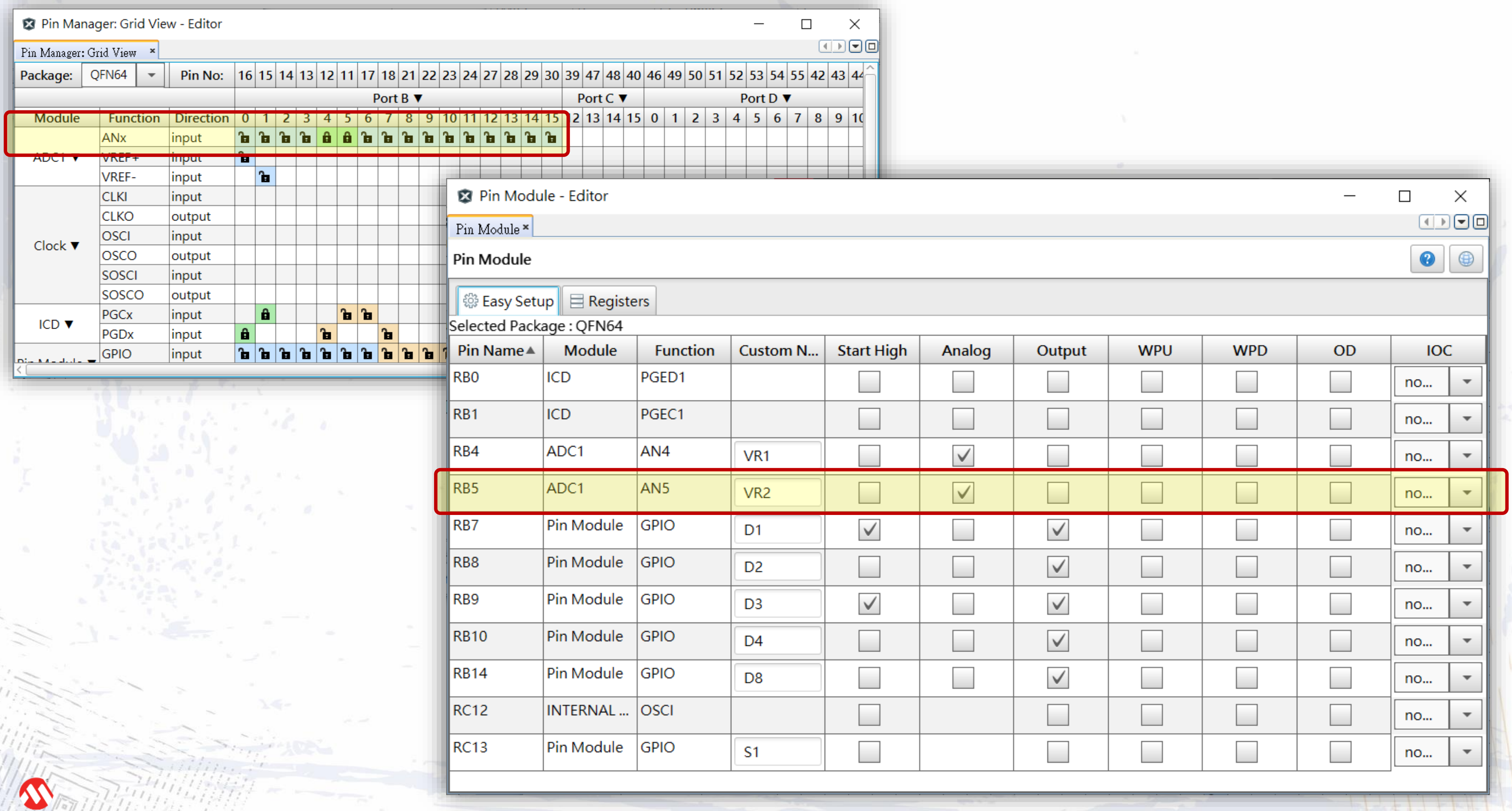
Selected Channels

Channel	Custom Name	Scan Enable
	CHANNEL VBG/2	☐
	CHANNEL VBG	☐
AN5	VR2	☑
AN4	VR1	☑

Interrupt Manager

Module	Interrupt	Description	IRQ Nu...	Enabled	Priority
Pin Module	CNI	CN - Change Notifica...	19	☐	1
ADC1	ADI	ADC1 - A/D Converte...	13	☑	5
TMR3	TI	T3 - Timer3	8	☐	1
TMR2	TI	T2 - Timer2	7	☑	3
TMR2	TNI	T3 - Timer3	8	☐	1
TMR1	TI	T1 - Timer1	3	☑	4

Lab11 ADC Scan with Interrupt
MCC’s Setting & Code Example



Module	Function	Direction
ADC1 ▼	ANx	input
	VREF+	input
	VREF-	input
Clock ▼	CLKI	input
	CLKO	output
	OSCI	input
	OSCO	output
	SOSCI	input
	SOSCO	output
ICD ▼	PGCx	input
	PGDx	input
Pin Module ▼	GPIO	input

Selected Package : QFN64

Pin Name▲	Module	Function	Custom N...	Start High	Analog	Output	WPU	WPD	OD	IOC
RB0	ICD	PGED1		☐	☐	☐	☐	☐	☐	no...
RB1	ICD	PGEC1		☐	☐	☐	☐	☐	☐	no...
RB4	ADC1	AN4	VR1	☐	☑	☐	☐	☐	☐	no...
RB5	ADC1	AN5	VR2	☐	☑	☐	☐	☐	☐	no...
RB7	Pin Module	GPIO	D1	☑	☐	☑	☐	☐	☐	no...
RB8	Pin Module	GPIO	D2	☐	☐	☑	☐	☐	☐	no...
RB9	Pin Module	GPIO	D3	☑	☐	☑	☐	☐	☐	no...
RB10	Pin Module	GPIO	D4	☐	☐	☑	☐	☐	☐	no...
RB14	Pin Module	GPIO	D8	☐	☐	☑	☐	☐	☐	no...
RC12	INTERNAL ...	OSCI		☐		☐	☐	☐	☐	no...
RC13	Pin Module	GPIO	S1	☐		☐	☐	☐	☐	no...

Lab11 ADC Scan with Interrupt MCC's Setting & Code Example

```
volatile unsigned char AD1Flag;
volatile unsigned int ADCResult[2];
int main(void)
{
    …
    while (1)
    {
        …
        if (AD1Flag)
        {
            AD1Flag = 0;
            LCM_SetCursor(4, 1);
            LCM_PutNumber(ADCResult[0], 4);
            LCM_SetCursor(13, 1);
            LCM_PutNumber(ADCResult[1] >> 1, 3);
        }
        …
    }
}

void ADC1_CallBack(void)
{
    ADC1_ConversionResultBufferGet((uint16_t *) ADCResult);
    AD1Flag = 1;
}
```