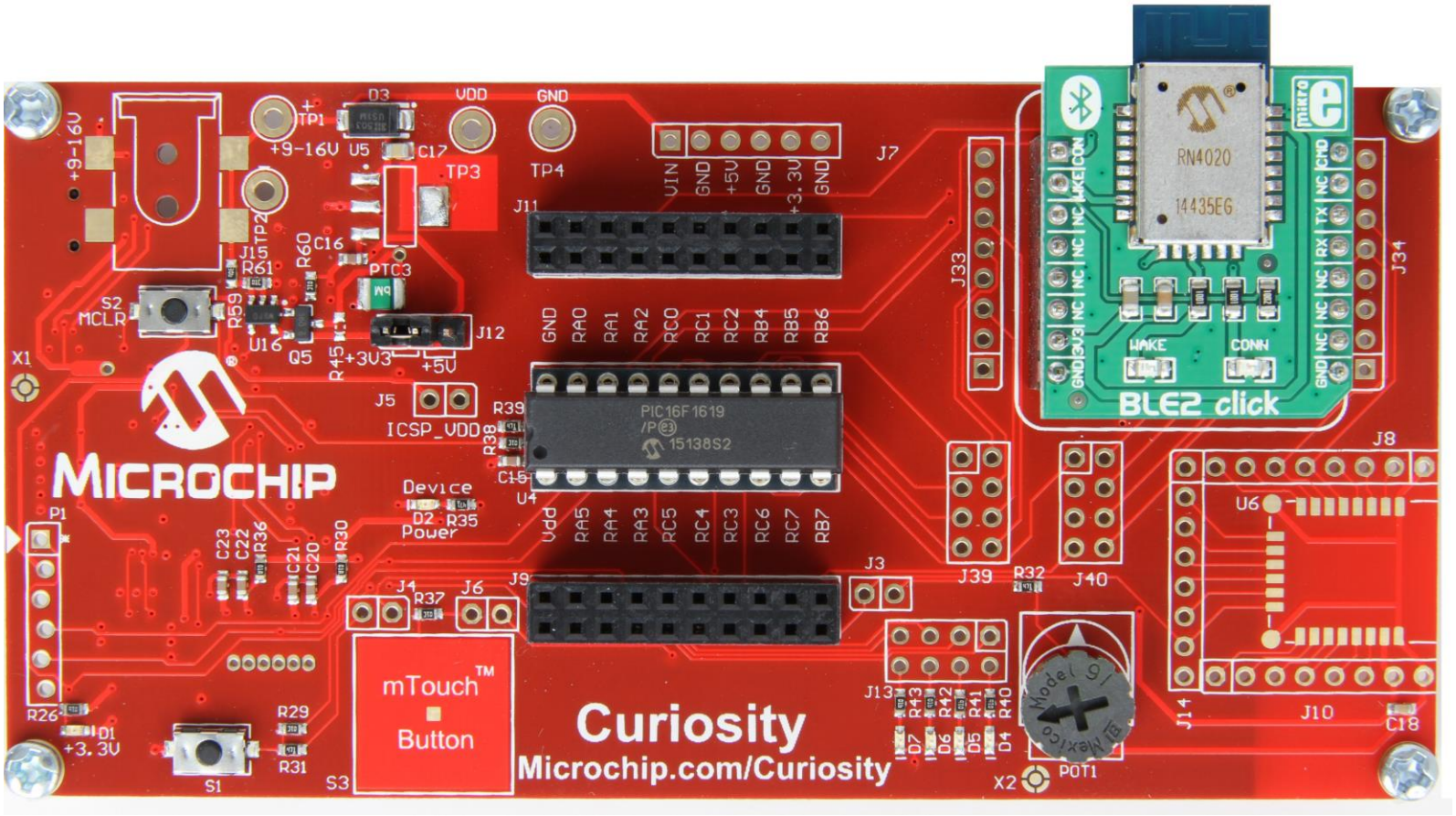
IoT Development Tools for PIC32



PIC32MM Curiosity

January 29, 2018

FRED EADY

Presented by:

DesignNews

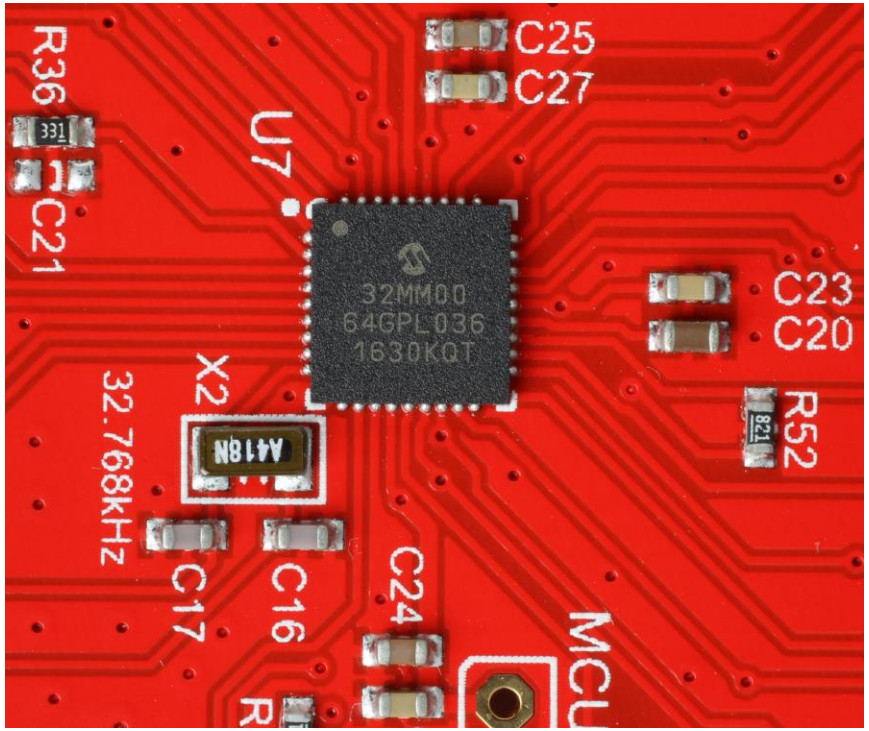
CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

IoT Development Tools for PIC32

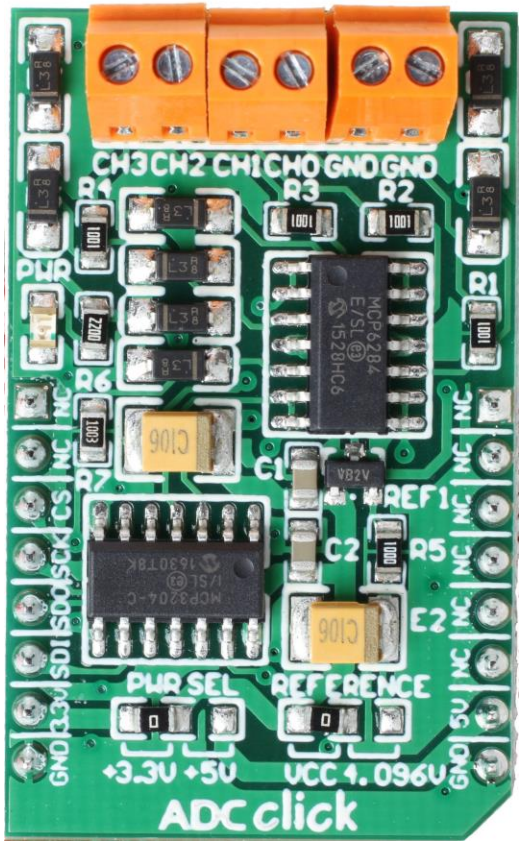
AGENDA

- PIC32MM Curiosity Hardware
- PIC32 Software Development Tools
- ADC click Project
- Double click – DAC click Project
- Hasta Mañana

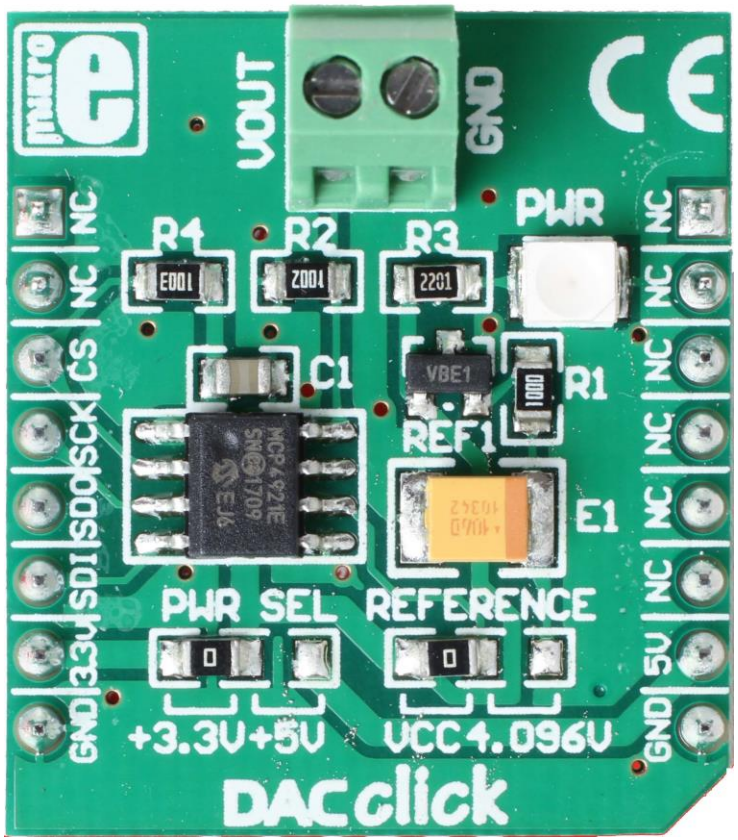


IoT Development Tools for PIC32

PIC32MM Curiosity Hardware – click Part Numbers



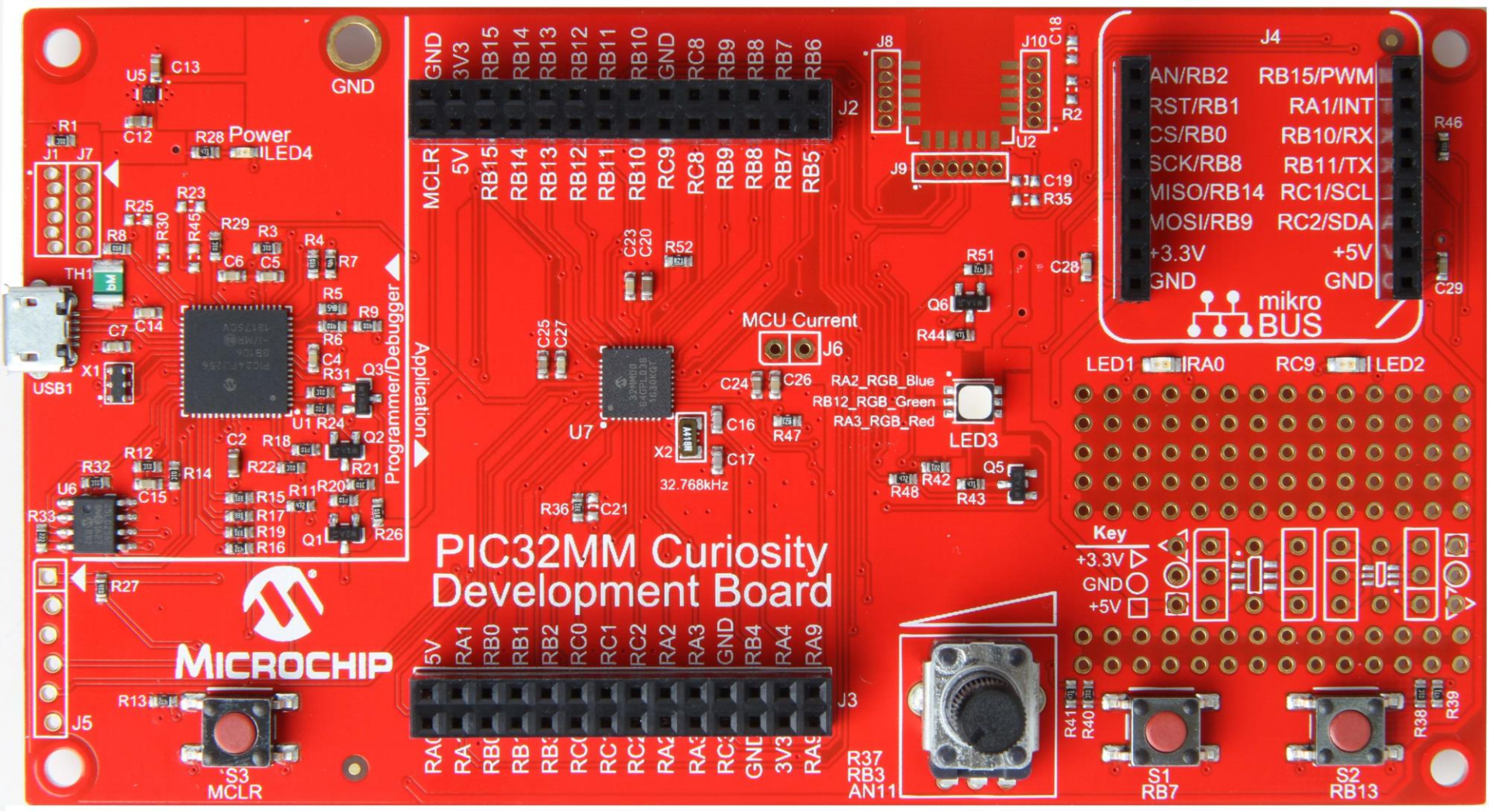
ADC CLICK MCP3204
MIKROE-922
1471-1301-ND



DAC CLICK MIKROBUS
MIKROE-950
1471-1317-ND

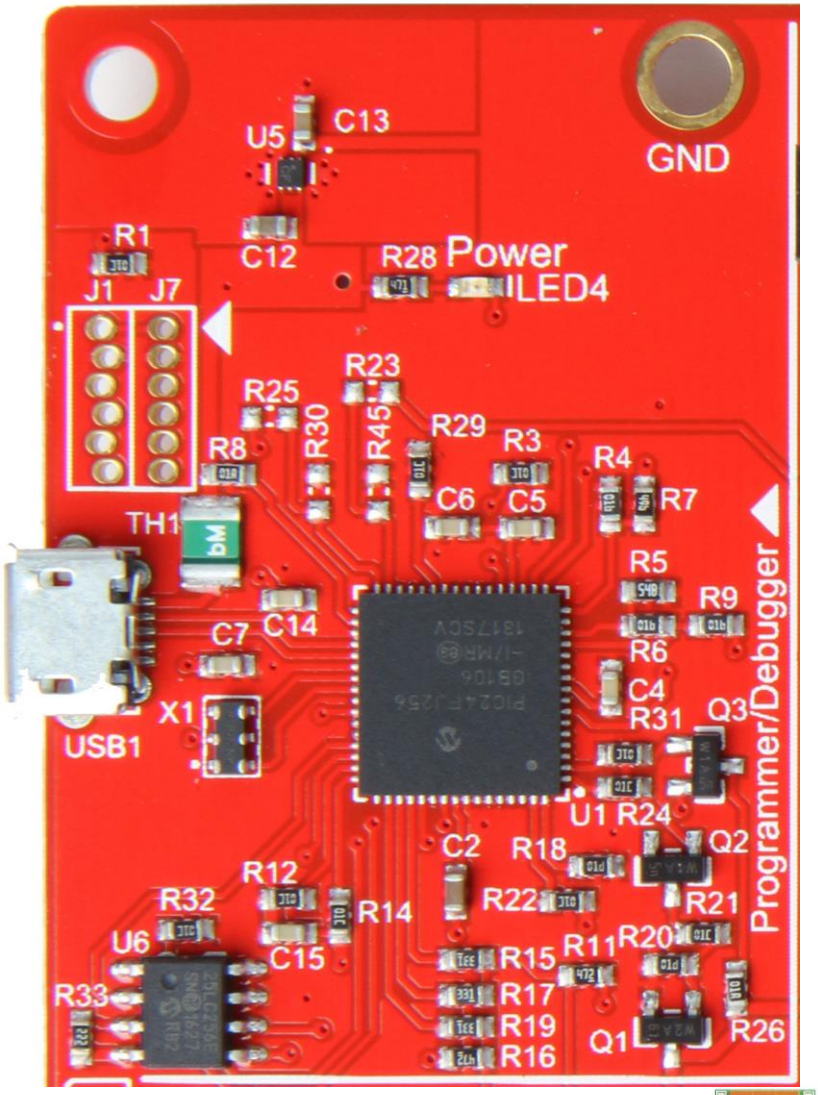
IoT Development Tools for PIC32

PIC32MM Curiosity Hardware



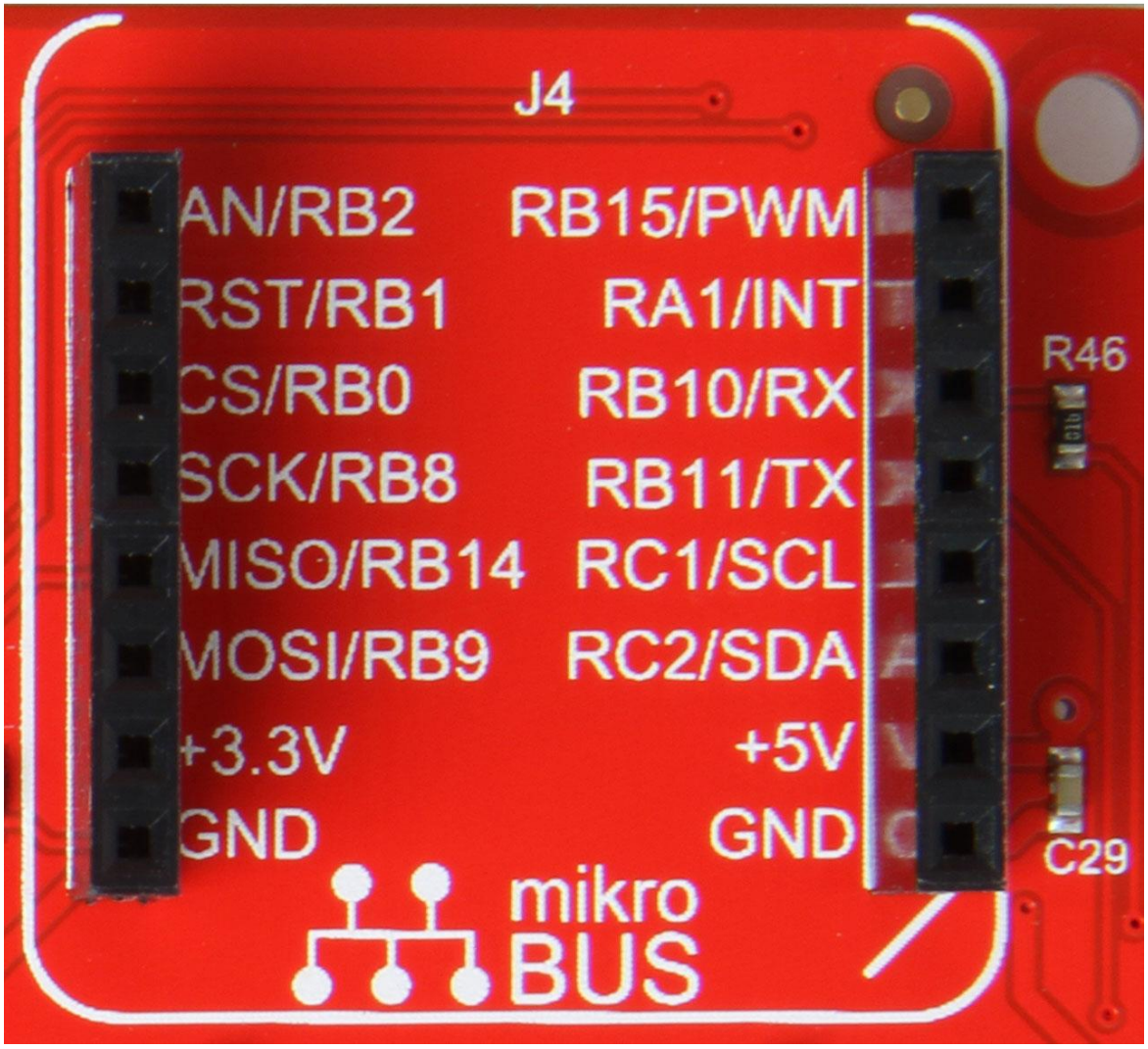
IoT Development Tools for PIC32

PIC32MM Curiosity Hardware - PKOB



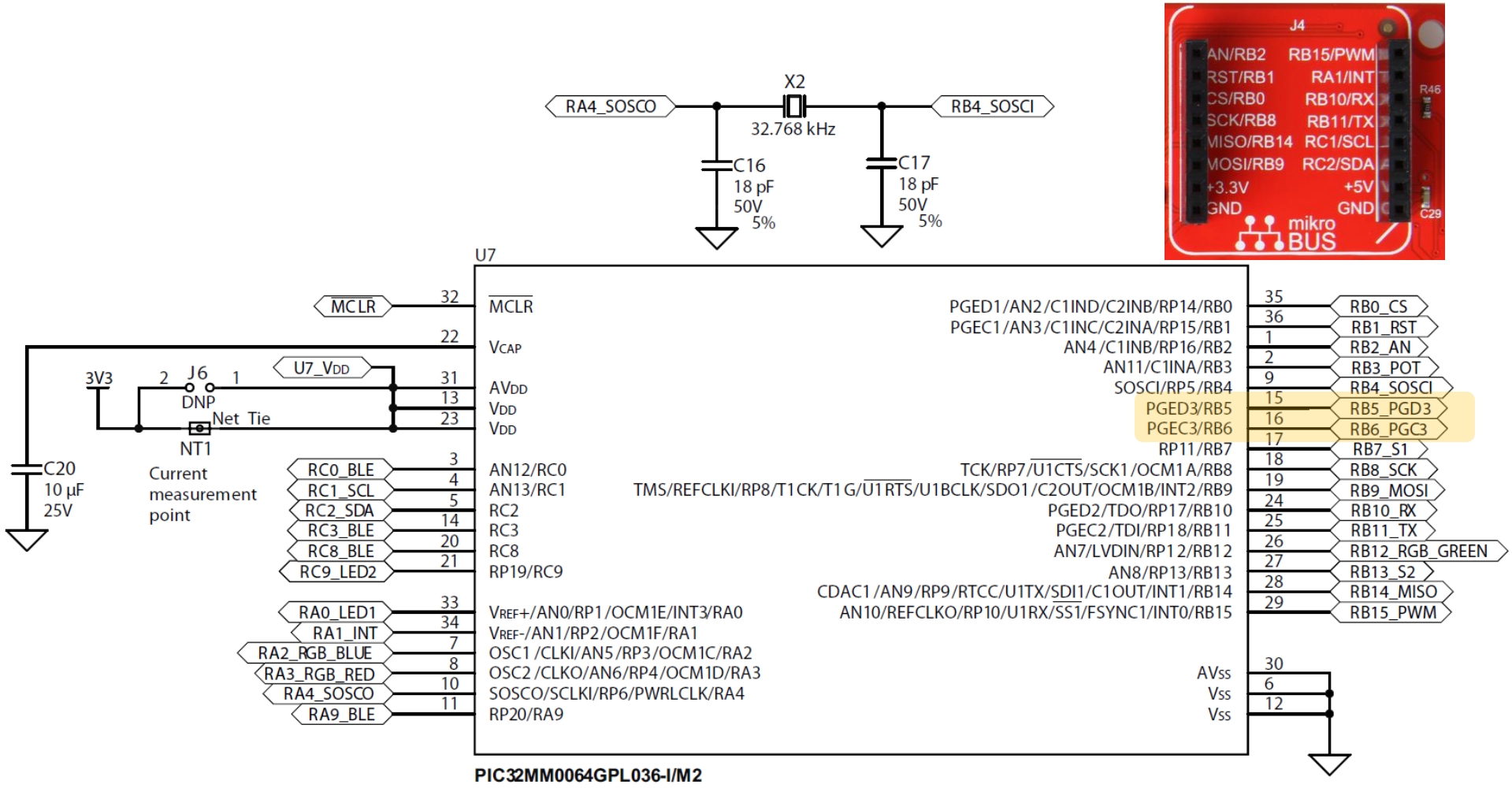
IoT Development Tools for PIC32

PIC32MM Curiosity Hardware - mikroBUS



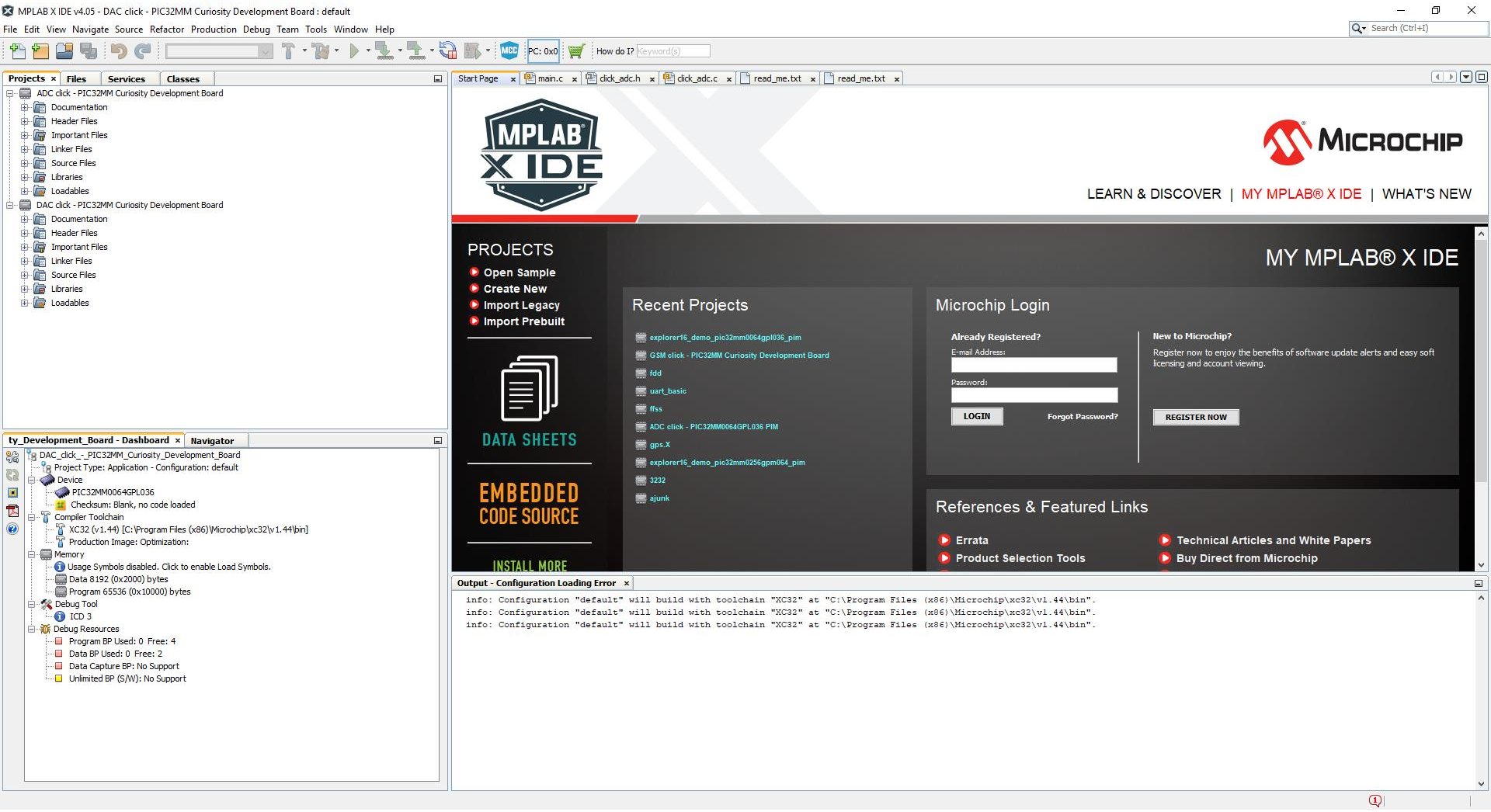
IoT Development Tools for PIC32

PIC32MM Curiosity Hardware – Program/Debug Pins



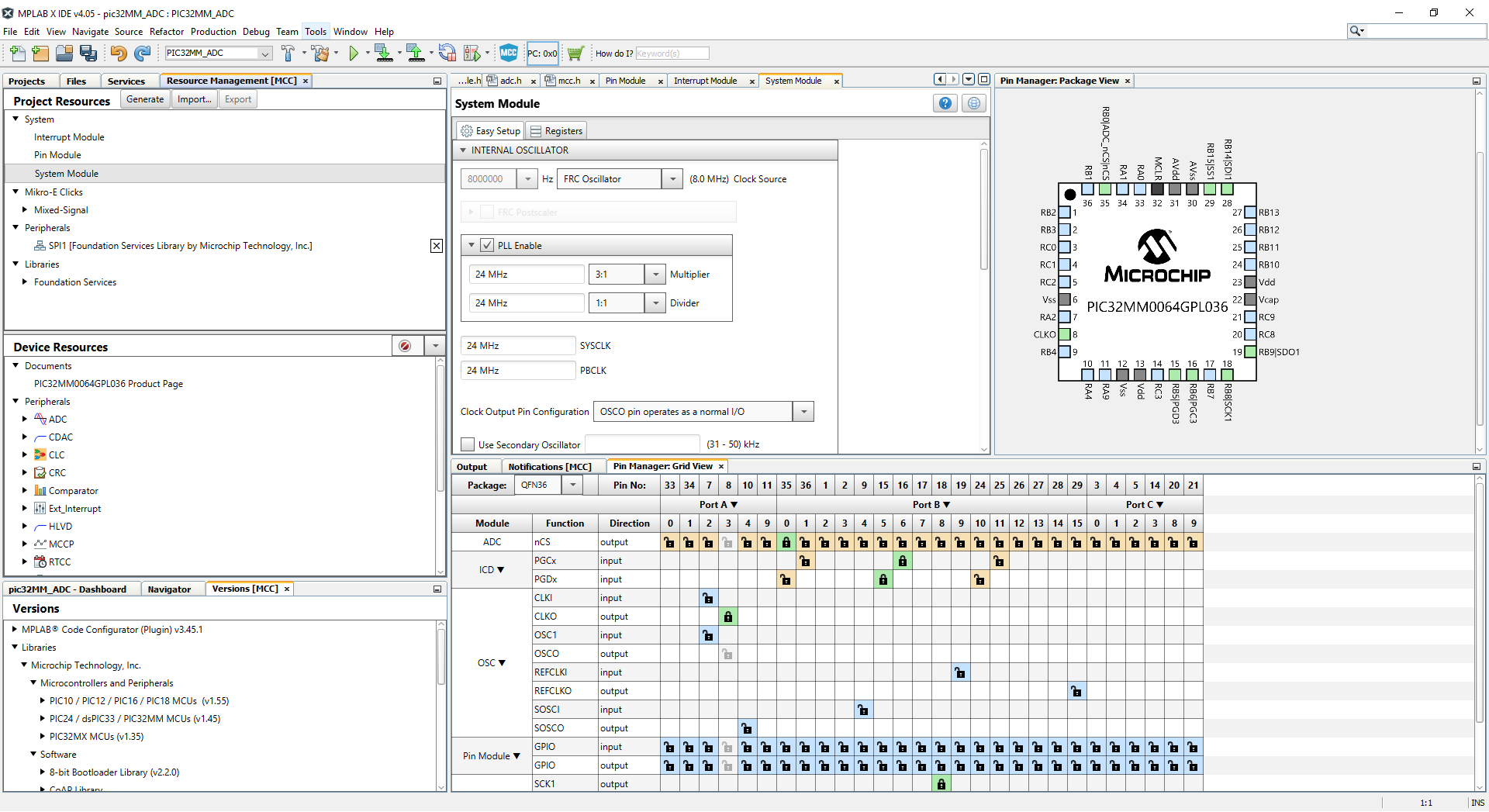
IoT Development Tools for PIC32

PIC32 Software Development Tools – MPLAB X



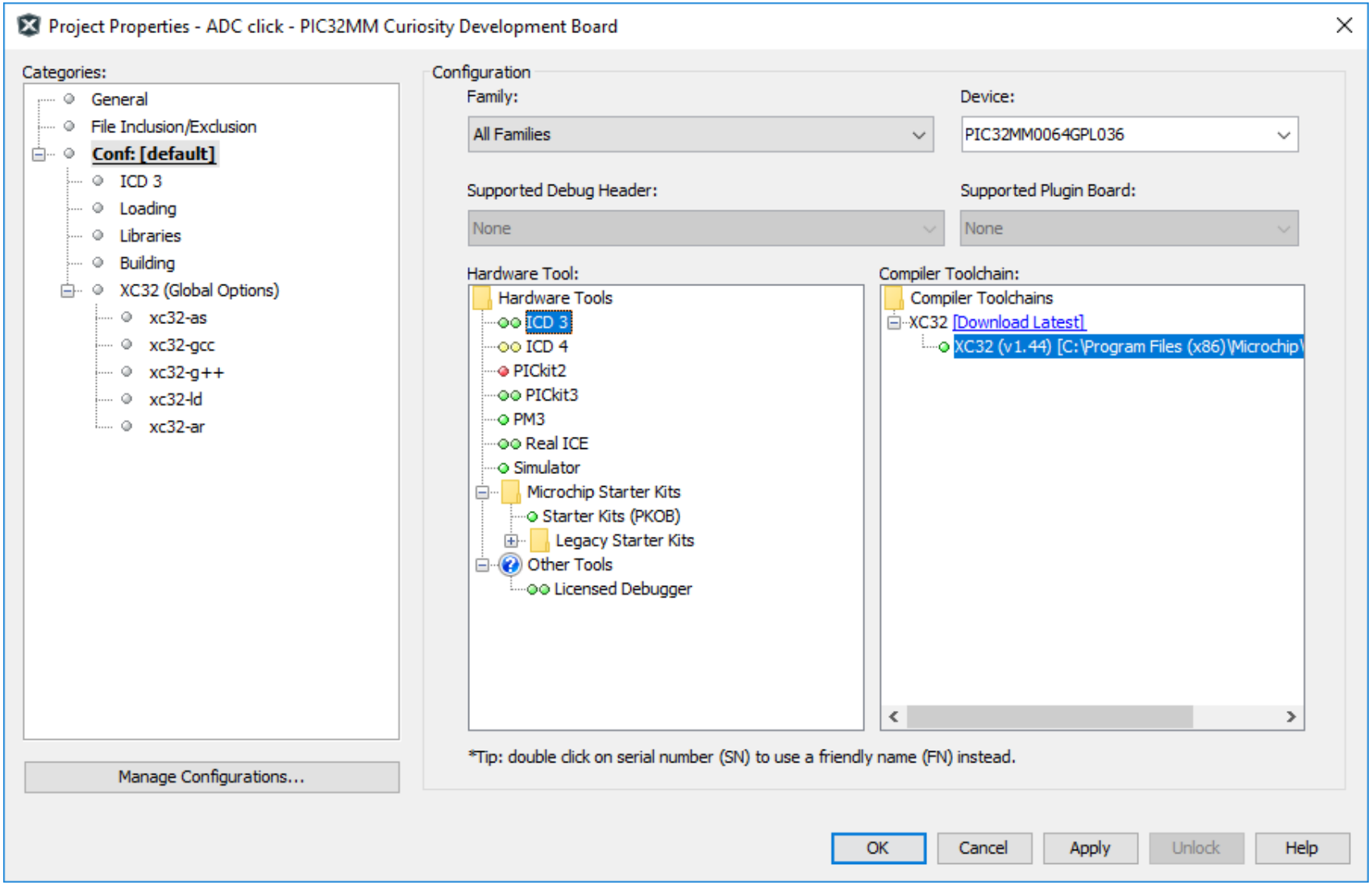
IoT Development Tools for PIC32

PIC32 Software Development Tools – MCC



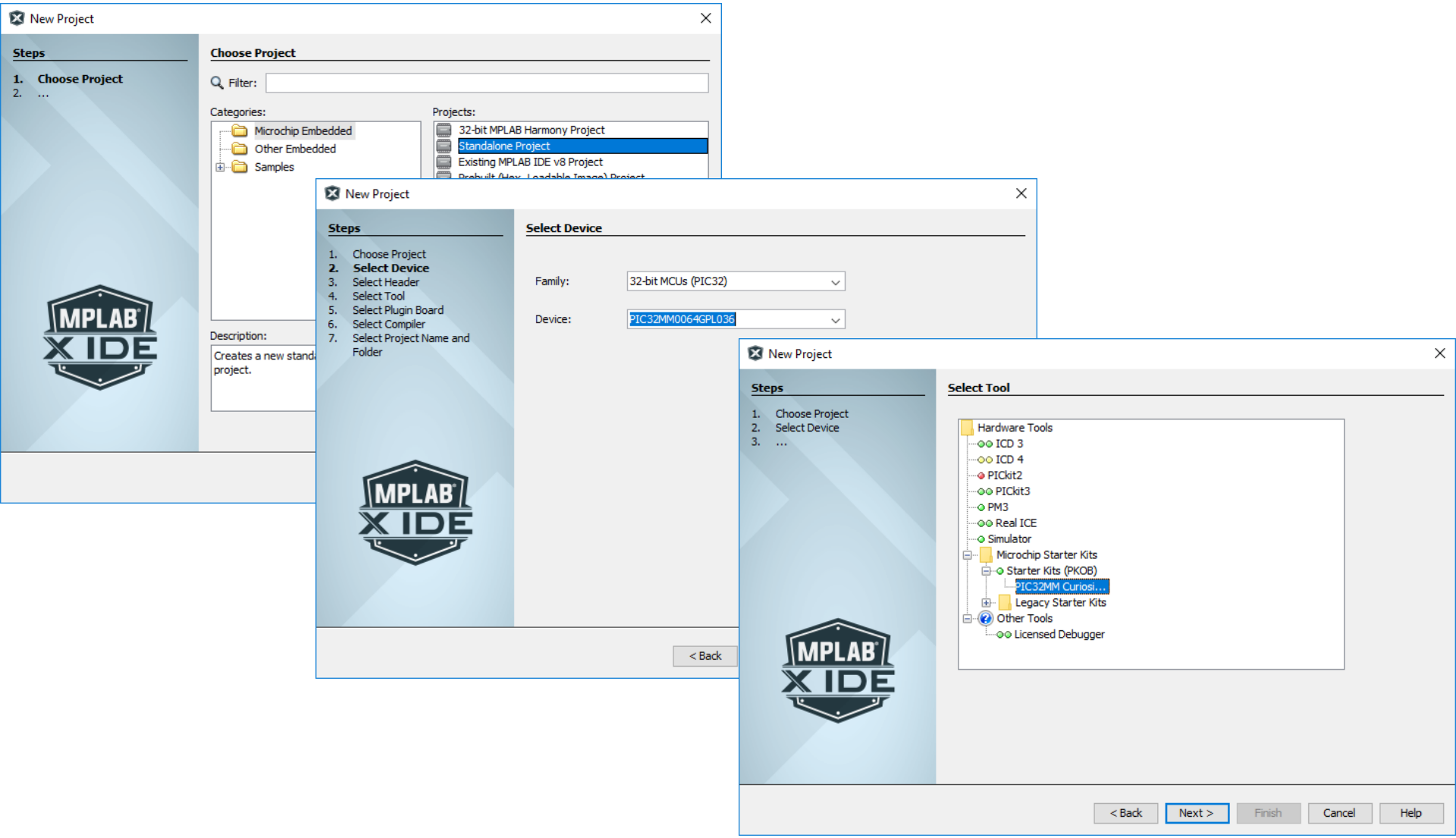
IoT Development Tools for PIC32

PIC32 Software Development Tools – XC32



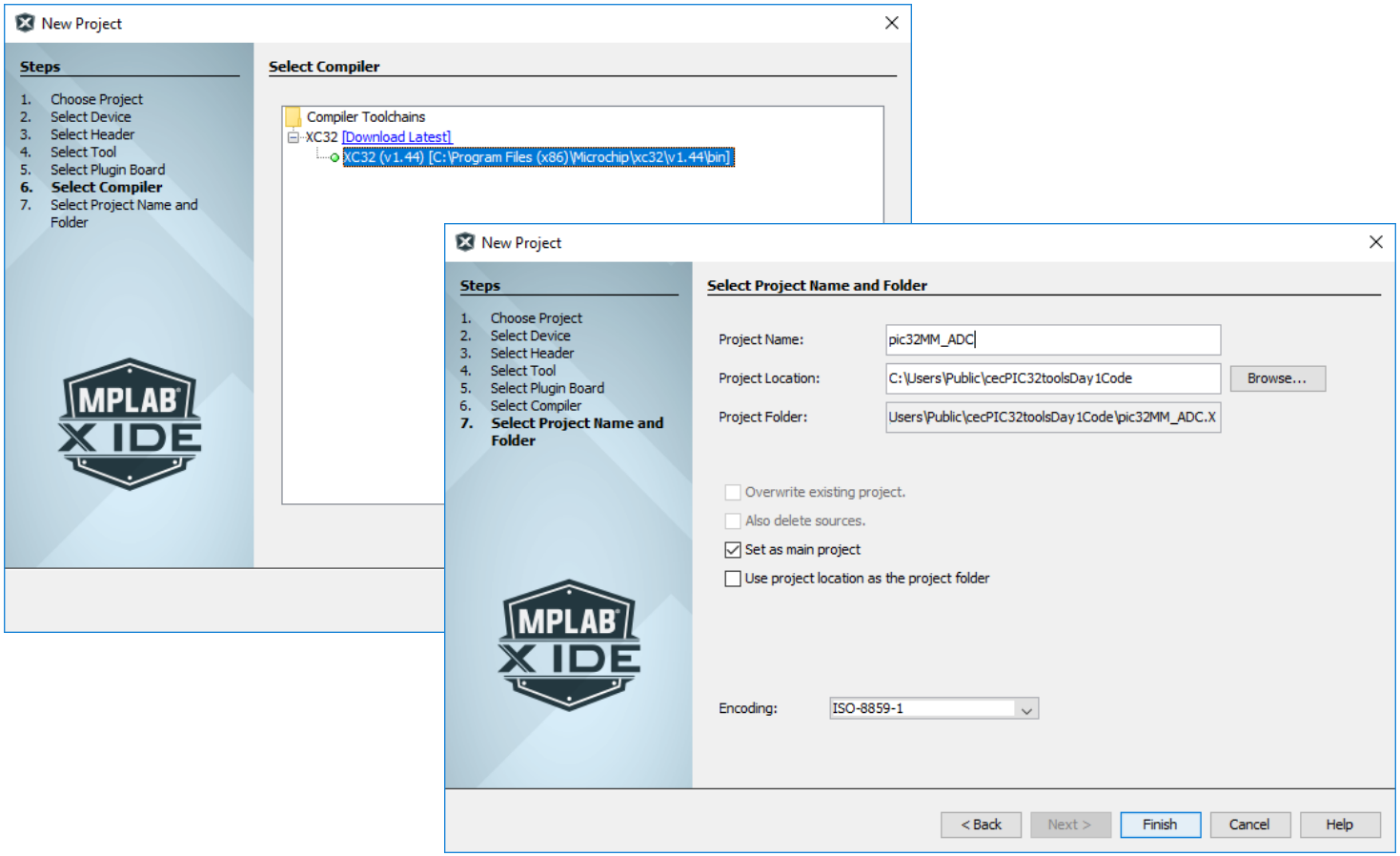
IoT Development Tools for PIC32

ADC click Project - Setup



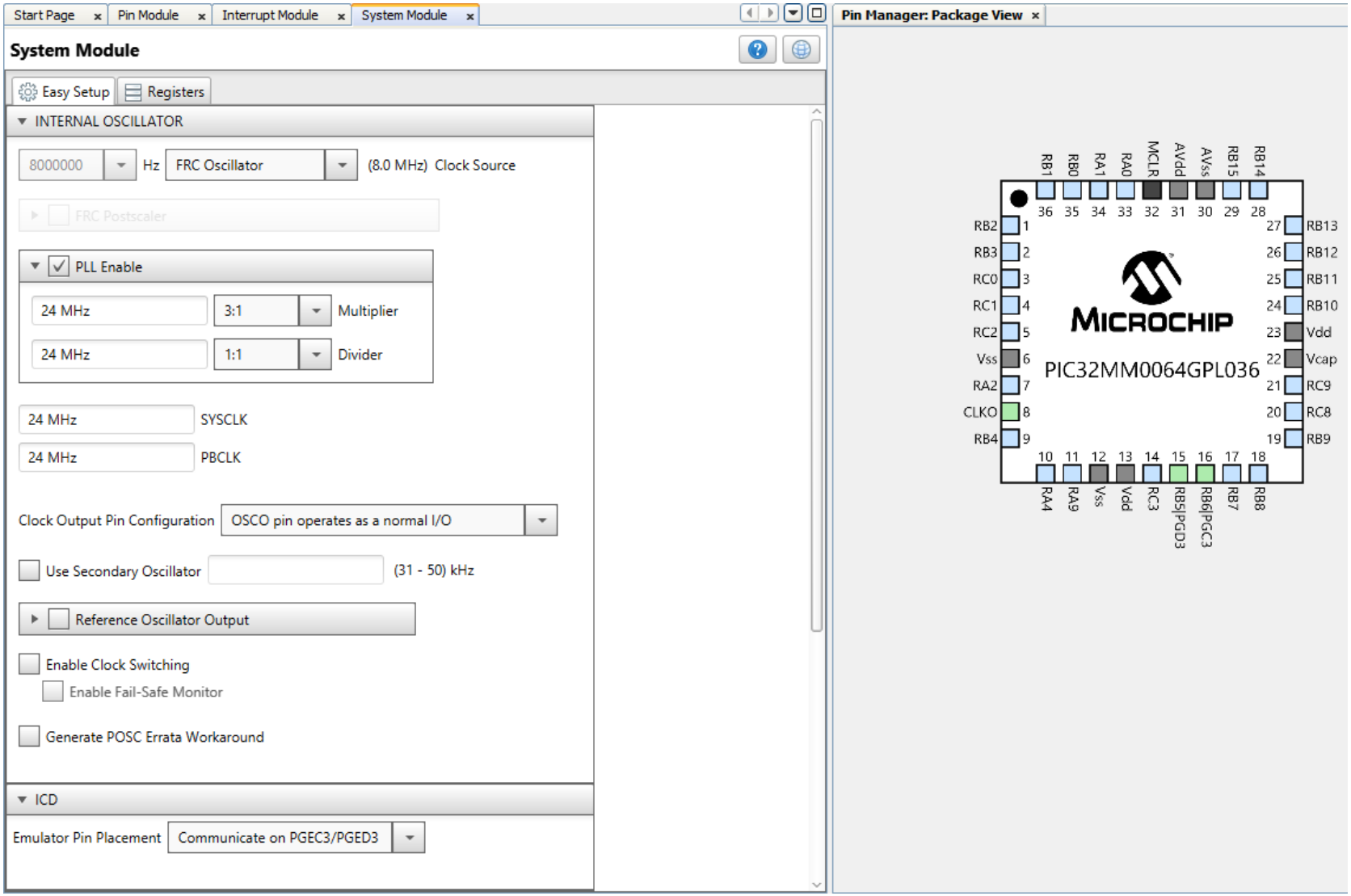
IoT Development Tools for PIC32

ADC click Project - Setup



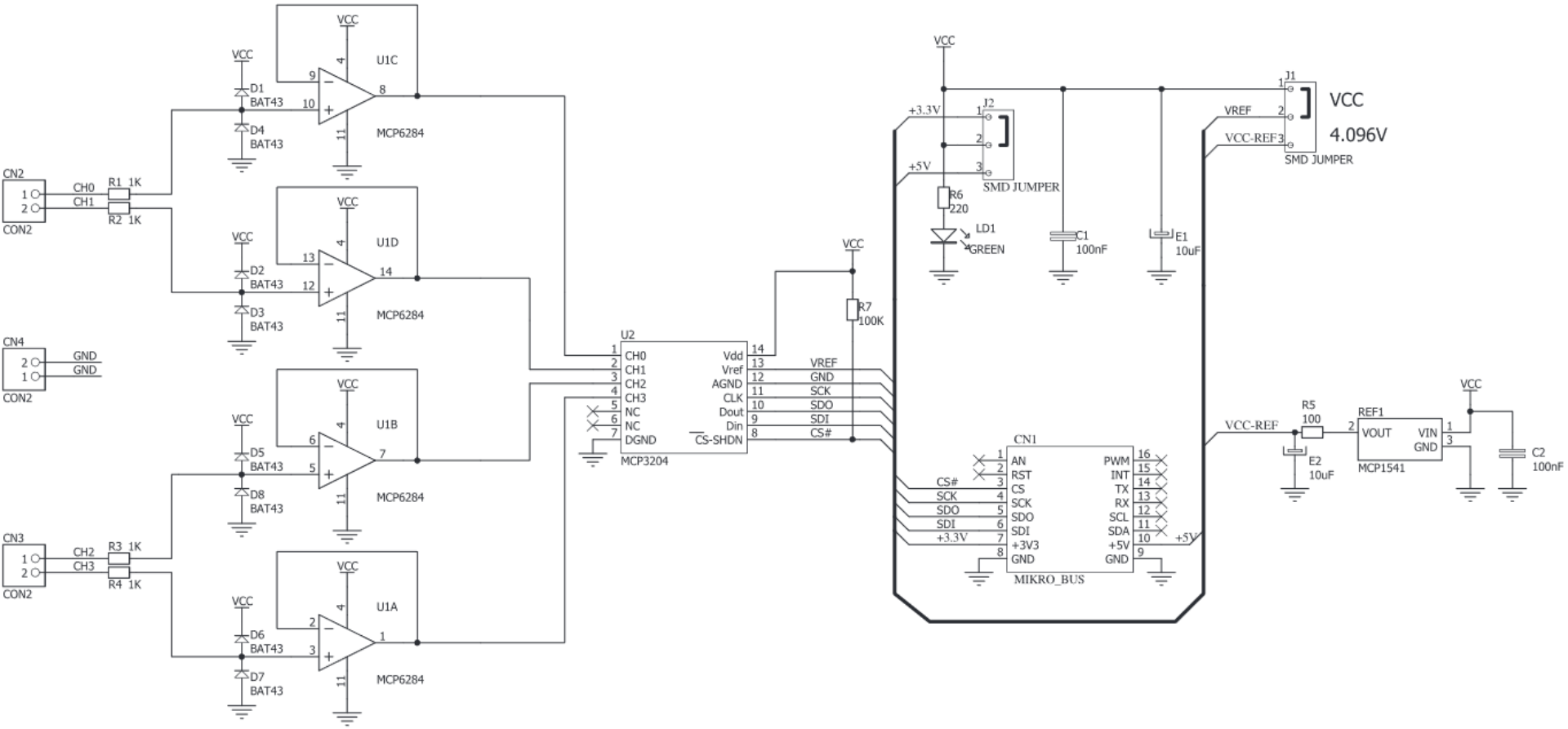
IoT Development Tools for PIC32

ADC click Project – MPLAB Code Configurator



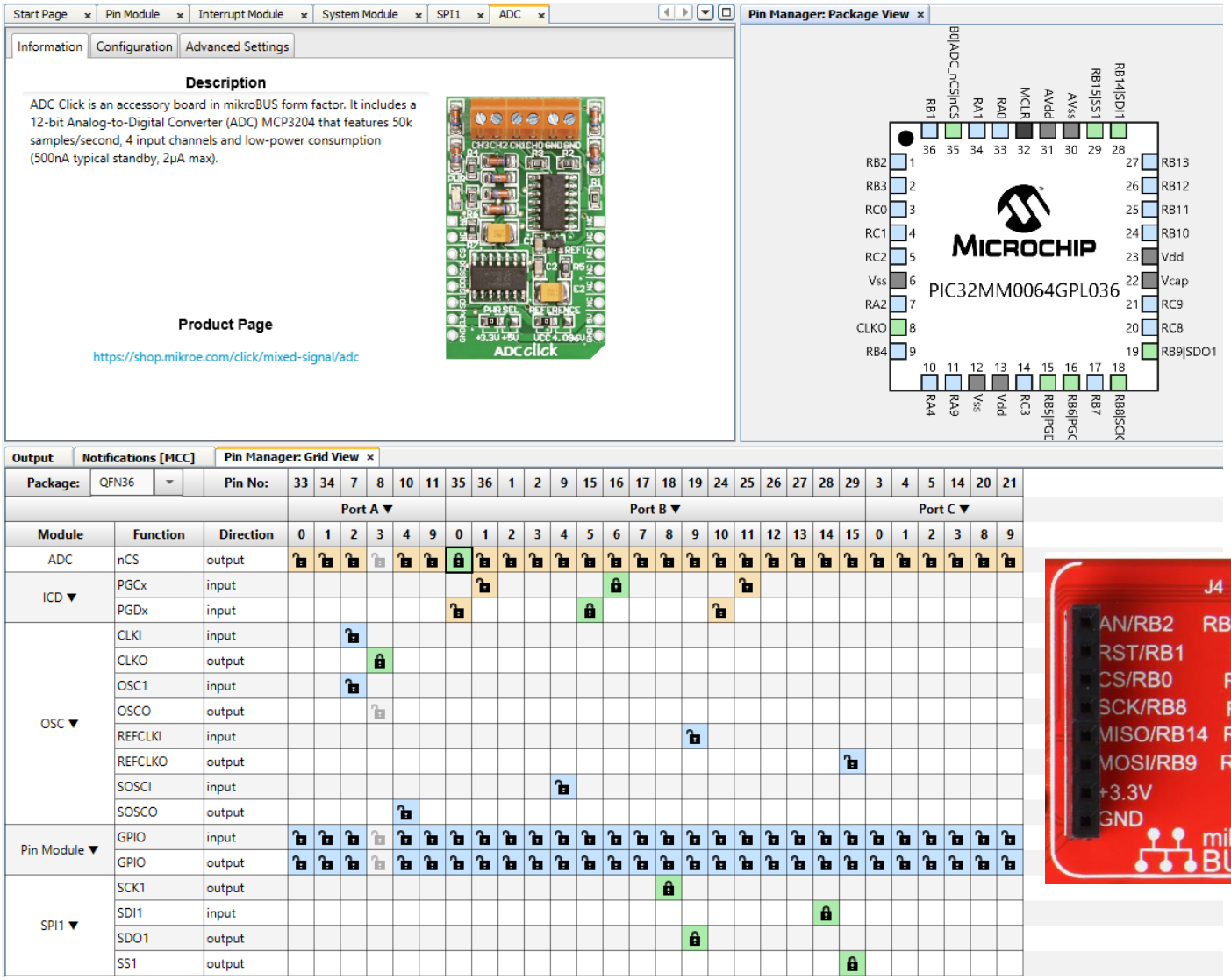
IoT Development Tools for PIC32

ADC click Project – MPLAB Code Configurator



IoT Development Tools for PIC32

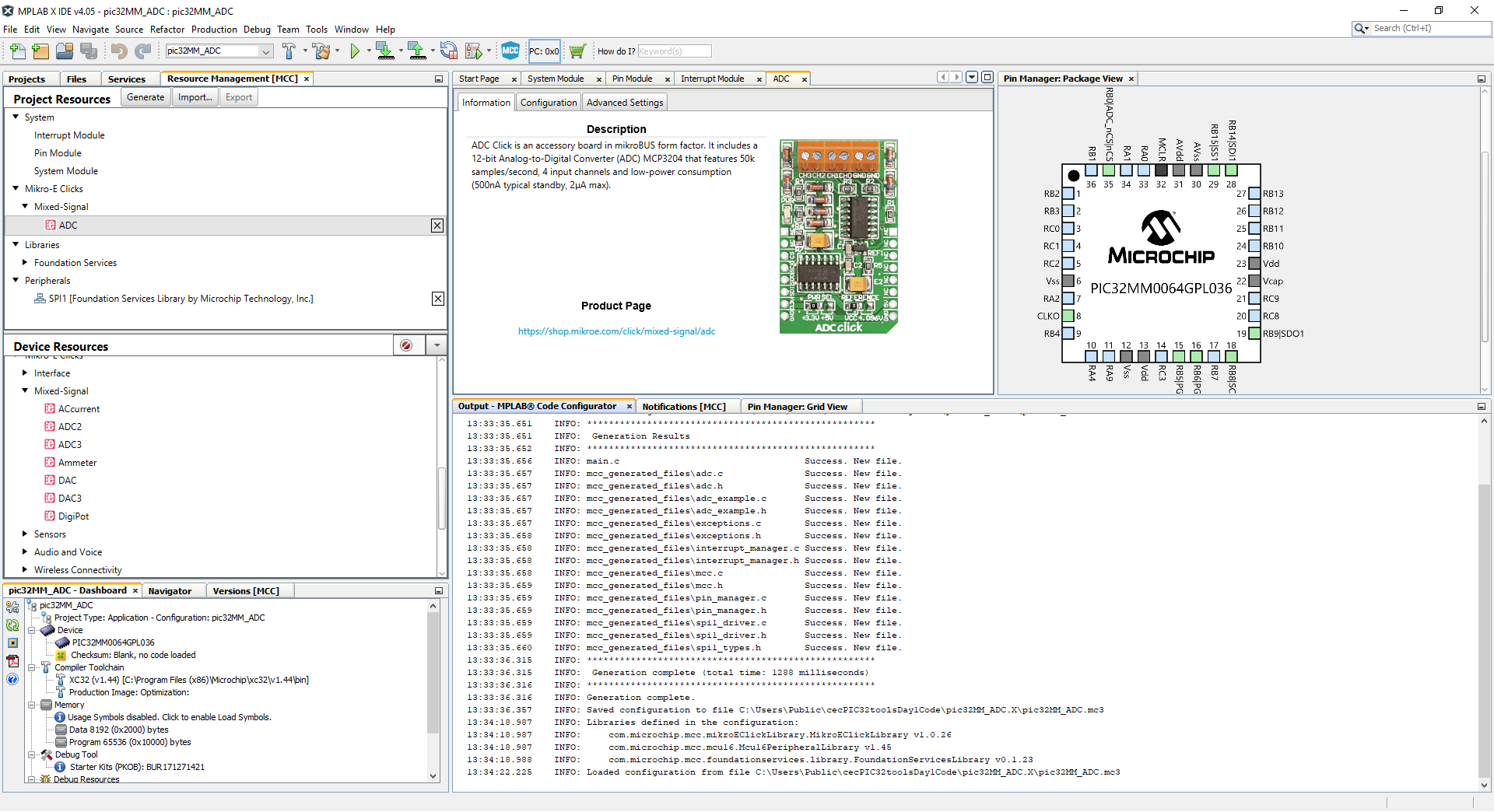
ADC click Project – MPLAB Code Configurator



Presented by:

IoT Development Tools for PIC32

ADC click Project – MPLAB Code Configurator



IoT Development Tools for PIC32

ADC click Project – MPLAB Code Configurator

MPLAB X IDE v4.05 - pic32MM_ADC : pic32MM_ADC

File Edit View Navigate Source Refactor Production Debug Team Tools Window Help

Projects | Files | Services | Resource Management [MCC]

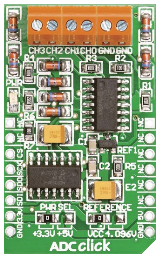
- pic32MM_ADC
  - Header Files
    - MCC Generated Files
      - adc.h
      - adc_example.h
      - exceptions.h
      - interrupt_manager.h
      - mcc.h
      - pin_manager.h
      - spi1_driver.h
      - spi1_types.h
  - Important Files
  - Linker Files
  - Source Files
    - main.c
    - MCC Generated Files
      - adc.c
      - adc_example.c
      - exceptions.c
      - interrupt_manager.c
      - mcc.c
      - pin_manager.c
      - spi1_driver.c
  - Libraries
  - Loadables

Start Page | System Module | Pin Module | Interrupt Module | ADC

Information | Configuration | Advanced Settings

Description

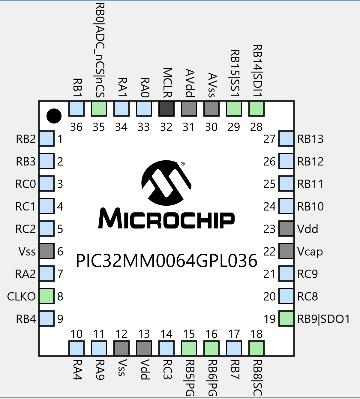
ADC Click is an accessory board in mikroBUS form factor. It includes a 12-bit Analog-to-Digital Converter (ADC) MCP3204 that features 50k samples/second, 4 input channels and low-power consumption (500nA typical standby, 2μA max).



Product Page

https://shop.mikroe.com/click/mixed-signal/adc

Pin Manager: Package View



Output - MPLAB® Code Configurator | Notifications [MCC] | Pin Manager: Grid View

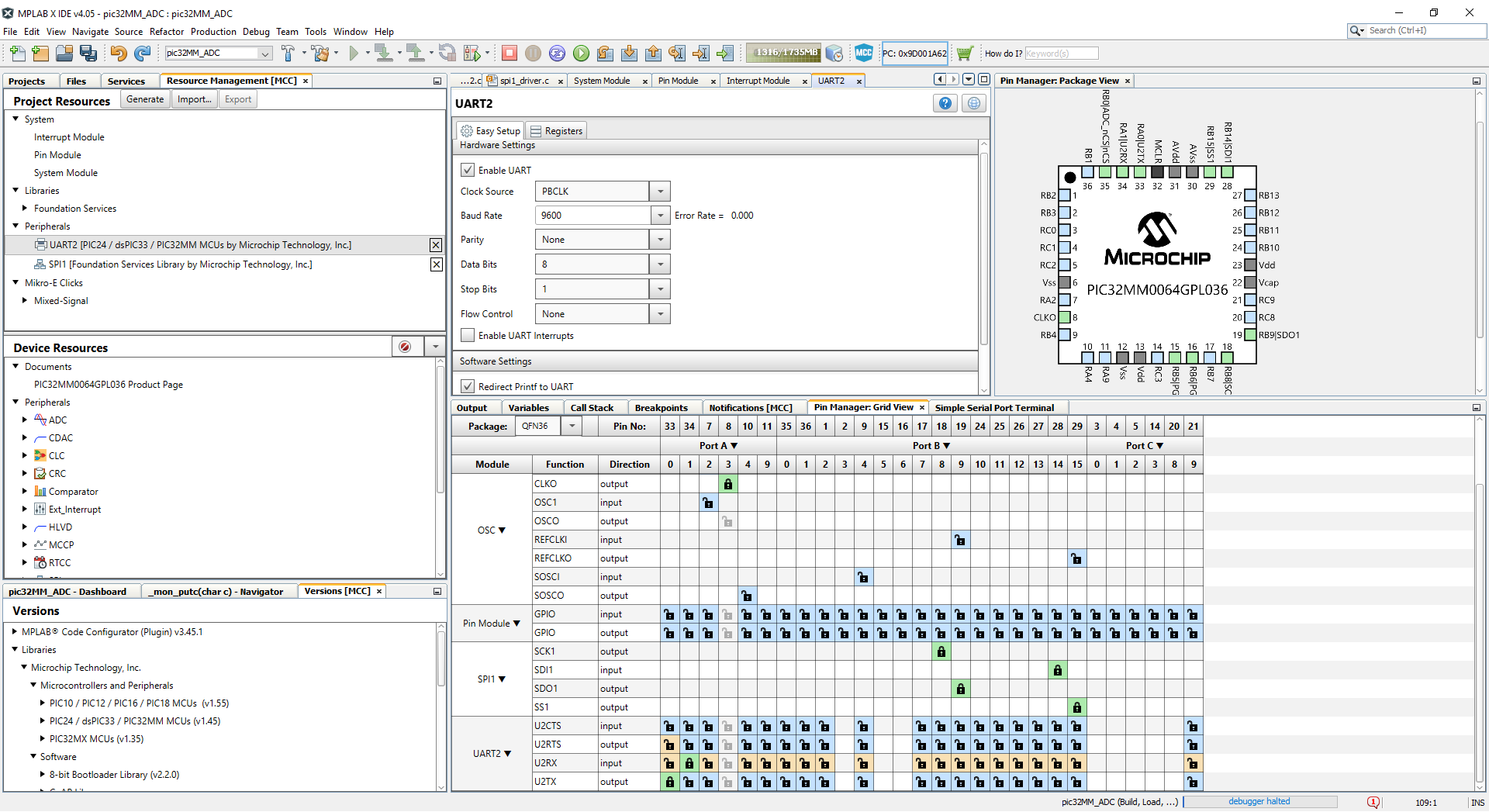
```
13:33:35.651    INFO: ********************************************************
13:33:35.651    INFO:  Generation Results
13:33:35.652    INFO: ********************************************************
13:33:35.656    INFO: main.c                                  Success. New file.
13:33:35.657    INFO: mcc_generated_files\adc.c               Success. New file.
13:33:35.657    INFO: mcc_generated_files\adc.h               Success. New file.
13:33:35.657    INFO: mcc_generated_files\adc_example.c       Success. New file.
13:33:35.657    INFO: mcc_generated_files\adc_example.h       Success. New file.
13:33:35.657    INFO: mcc_generated_files\exceptions.c        Success. New file.
13:33:35.658    INFO: mcc_generated_files\exceptions.h        Success. New file.
13:33:35.658    INFO: mcc_generated_files\interrupt_manager.c Success. New file.
13:33:35.658    INFO: mcc_generated_files\interrupt_manager.h Success. New file.
13:33:35.658    INFO: mcc_generated_files\mcc.c               Success. New file.
13:33:35.659    INFO: mcc_generated_files\mcc.h               Success. New file.
13:33:35.659    INFO: mcc_generated_files\pin_manager.c       Success. New file.
13:33:35.659    INFO: mcc_generated_files\pin_manager.h       Success. New file.
13:33:35.659    INFO: mcc_generated_files\spi1_driver.c       Success. New file.
13:33:35.659    INFO: mcc_generated_files\spi1_driver.h       Success. New file.
13:33:35.660    INFO: mcc_generated_files\spi1_types.h        Success. New file.
13:33:36.315    INFO: ********************************************************
13:33:36.315    INFO:  Generation complete (total time: 1288 milliseconds)
13:33:36.316    INFO: ********************************************************
13:33:36.316    INFO: Generation complete.
13:33:36.357    INFO: Saved configuration to file C:\Users\Public\cecPIC32toolsDay1Code\pic32MM_ADC.X\pic32MM_ADC.mc3
13:34:18.987    INFO: Libraries defined in the configuration:
13:34:18.987    INFO:     com.microchip.mcc.mikroEClickLibrary.MikroEClickLibrary v1.0.26
13:34:18.987    INFO:     com.microchip.mcc.mcu16.Mcu16PeripheralLibrary v1.45
13:34:18.988    INFO:     com.microchip.mcc.foundationservices.library.FoundationServicesLibrary v0.1.23
13:34:22.225    INFO: Loaded configuration from file C:\Users\Public\cecPIC32toolsDay1Code\pic32MM_ADC.X\pic32MM_ADC.mc3
```

pic32MM_ADC - Dashboard | Navigator | Versions [MCC]

- pic32MM_ADC
  - Project Type: Application - Configuration: pic32MM_ADC
  - Device
    - PIC32MM0064GPL036
    - Checksum: Blank, no code loaded
  - Compiler Toolchain
    - XC32 (v1.44) [C:\Program Files (x86)\Microchip\xc32\v1.44\bin]
    - Production Image: Optimization:
  - Memory
    - Usage Symbols disabled. Click to enable Load Symbols.
    - Data 8192 (0x2000) bytes
    - Program 65536 (0x10000) bytes
  - Debug Tool
    - Starter Kits (PKOB): BUR171271421
  - Debug Resources

IoT Development Tools for PIC32

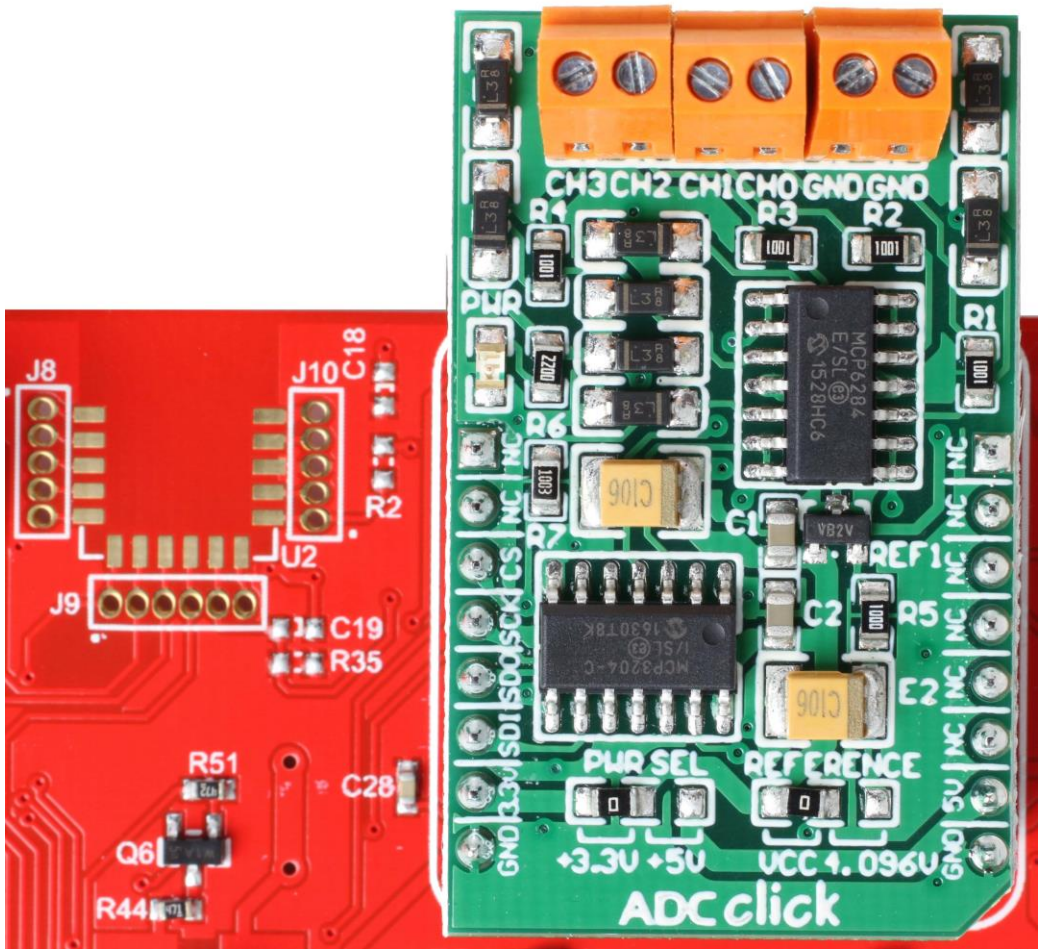
ADC click Project – MPLAB Code Configurator



IoT Development Tools for PIC32

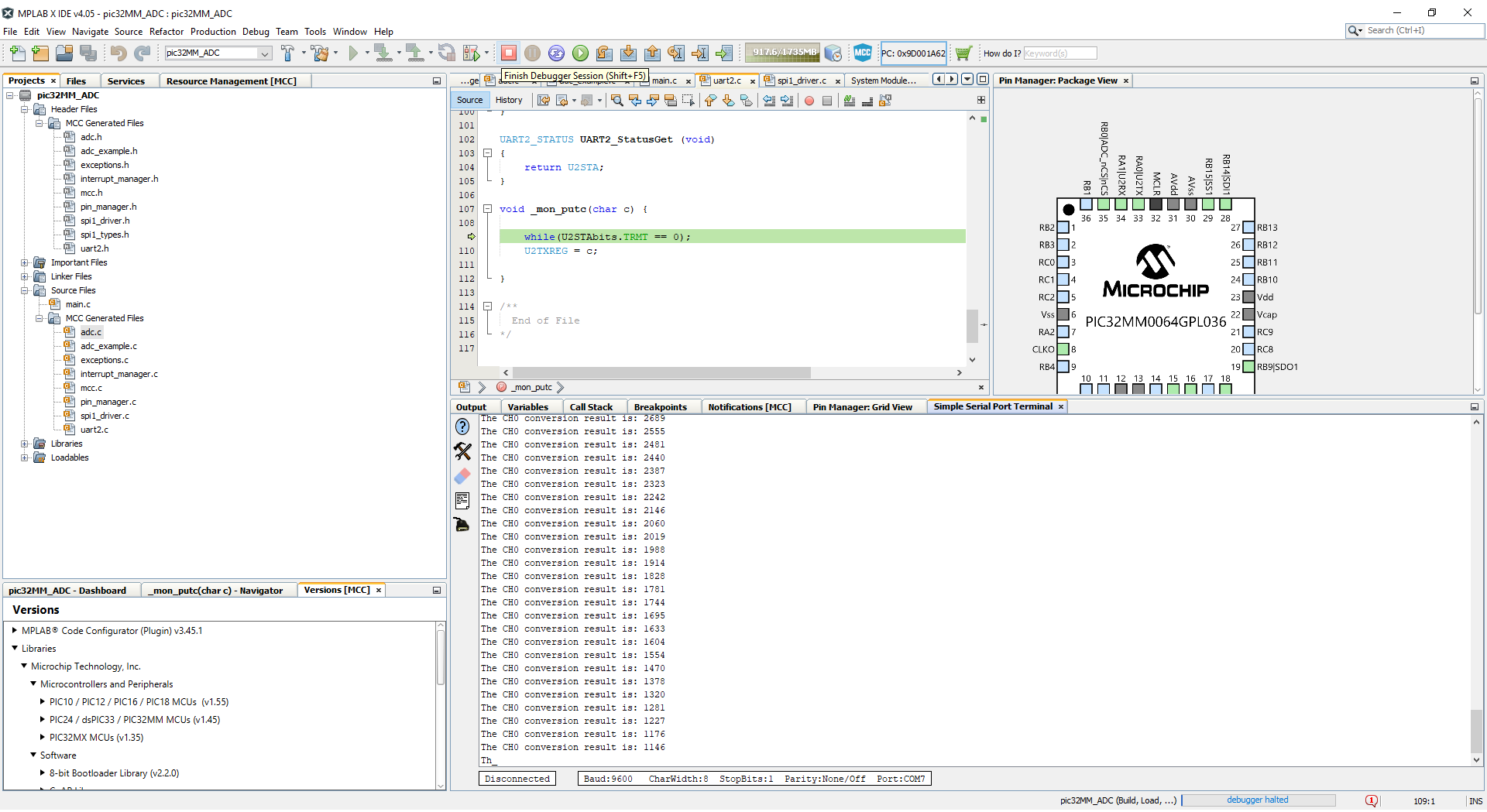
ADC click Project – MPLAB Code Configurator

```
/**
  Section: Driver APIs
 */
uint16_t ADC_getConversionCH0(void) {
    return ADC_getConversionResult(ADC_CH0, SINGLE);
}
uint16_t ADC_getConversionCH1(void) {
    return ADC_getConversionResult(ADC_CH1, SINGLE);
}
uint16_t ADC_getConversionCH2(void) {
    return ADC_getConversionResult(ADC_CH2, SINGLE);
}
uint16_t ADC_getConversionCH3(void) {
    return ADC_getConversionResult(ADC_CH3, SINGLE);
}
uint16_t ADC_getConversionResult(uint8_t channel, uint8_t input_mode) {
    if (!adc_initialized) {
        ADC_initializeClick();
    }
    uint8_t readData[3];
    uint16_t configBits = 0x0400 | (input_mode << 9) | (channel << 6);
    uint16_t conversionResult;
    ADC_startConversion();
    readData[0] = spil_exchangeByte(configBits >> 8);
    readData[1] = spil_exchangeByte(configBits);
    readData[2] = spil_exchangeByte(0x00);
    ADC_stopConversion();
    conversionResult = ((uint16_t)readData[1]) << 8 | readData[2];
    return conversionResult;
}
void ADC_initializeClick(void) {
    spil_master_open(ADC_CLICK);
    LATBbits.LATB0 = 1;
    adc_initialized = 1;
}
void ADC_startConversion(void) {
    LATBbits.LATB0 = 0;
//    __delay_us(CS_DELAY);
}
void ADC_stopConversion(void) {
    LATBbits.LATB0 = 1;
}
```



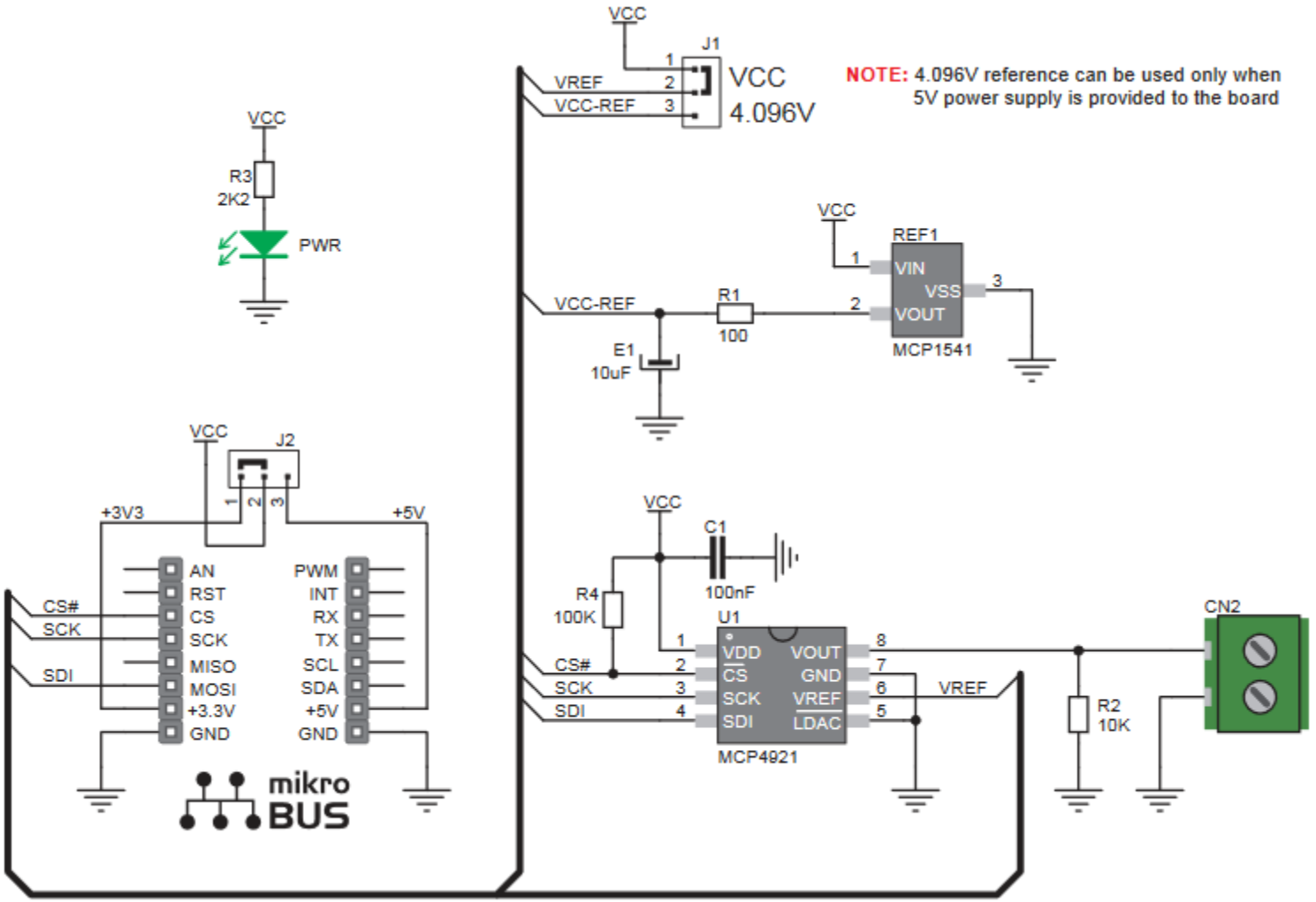
IoT Development Tools for PIC32

ADC click Project – MPLAB Code Configurator



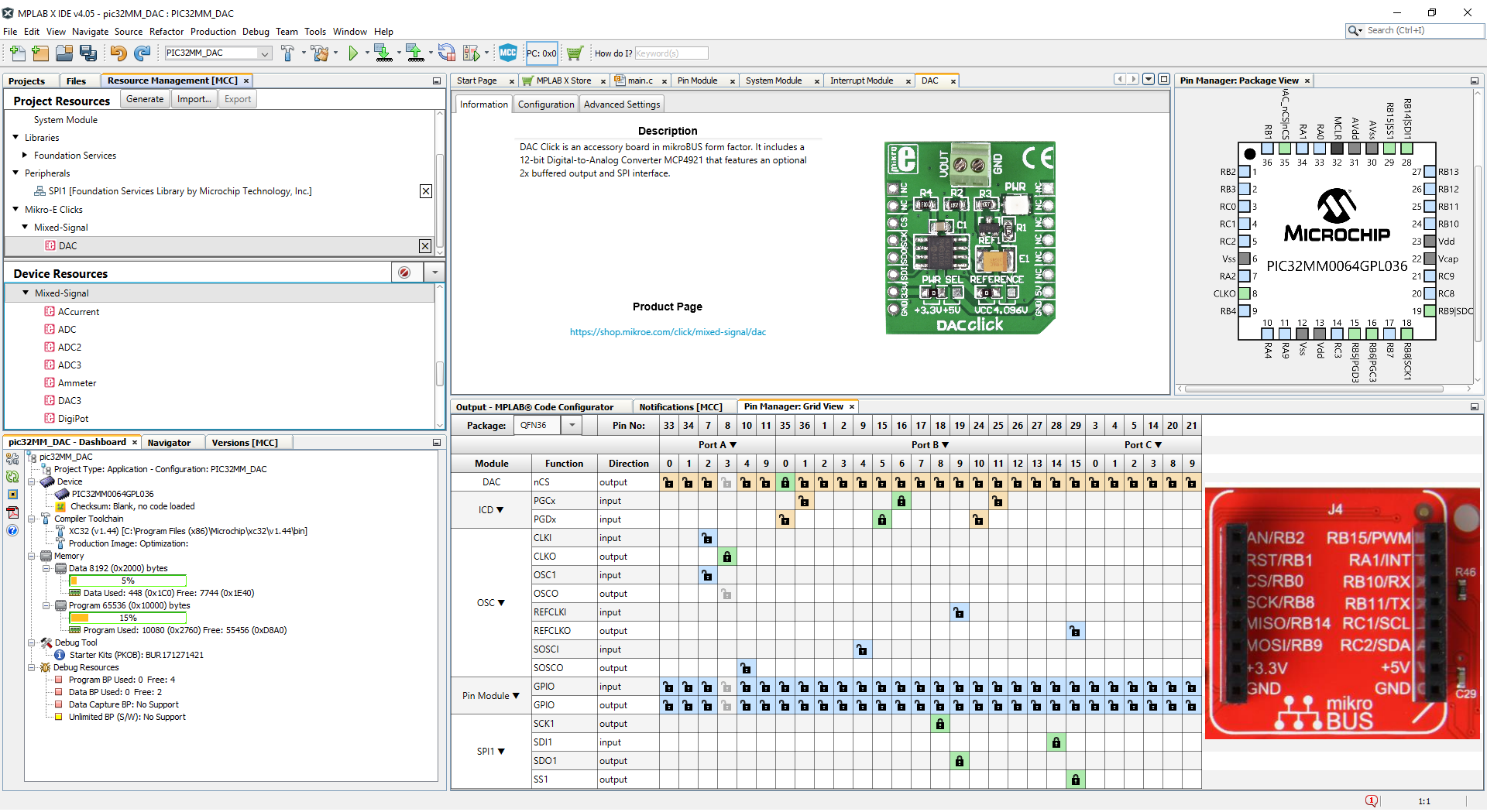
IoT Development Tools for PIC32

Double click – DAC click Project



IoT Development Tools for PIC32

Double click – DAC click Project

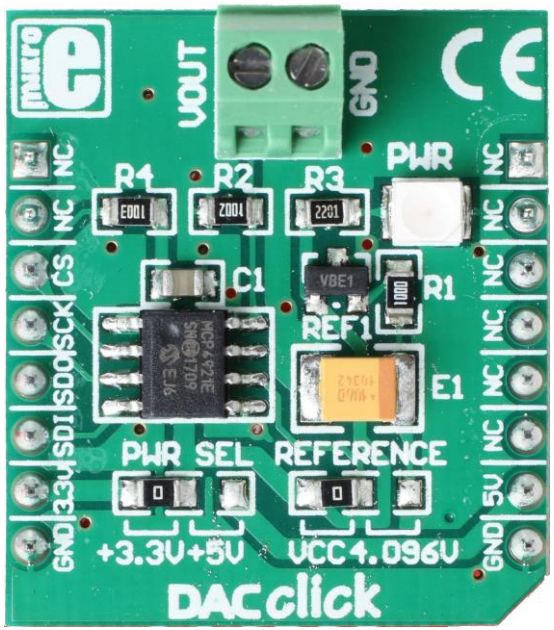


IoT Development Tools for PIC32

Double click – DAC click Project – Delay Function

```
//******************************************************************
//* CORE TIMER DELAY FUNCTION
//* WRITTEN BY FRED EADY
//* NO USB SUPPORT IN THIS VERSION
//* COMPILED USING XC32 v1.44 - MPLABX v4.05
//* LAST UPDATE 01/10/2018
//* NOTES:
//******************************************************************

#ifndef CORETIMERDELAY_H
#define CORETIMERDELAY_H

#ifdef __cplusplus
extern "C" {
#endif

void ctDelayms(uint16_t ms);

#define GetSystemClock()            24000000UL
#define GetPeripheralClock()        24000000UL
#define CoreTicksPerMs              (GetSystemClock() / 2000)

#ifdef __cplusplus
}
#endif
#endif  /* CORETIMERDELAY_H */
```

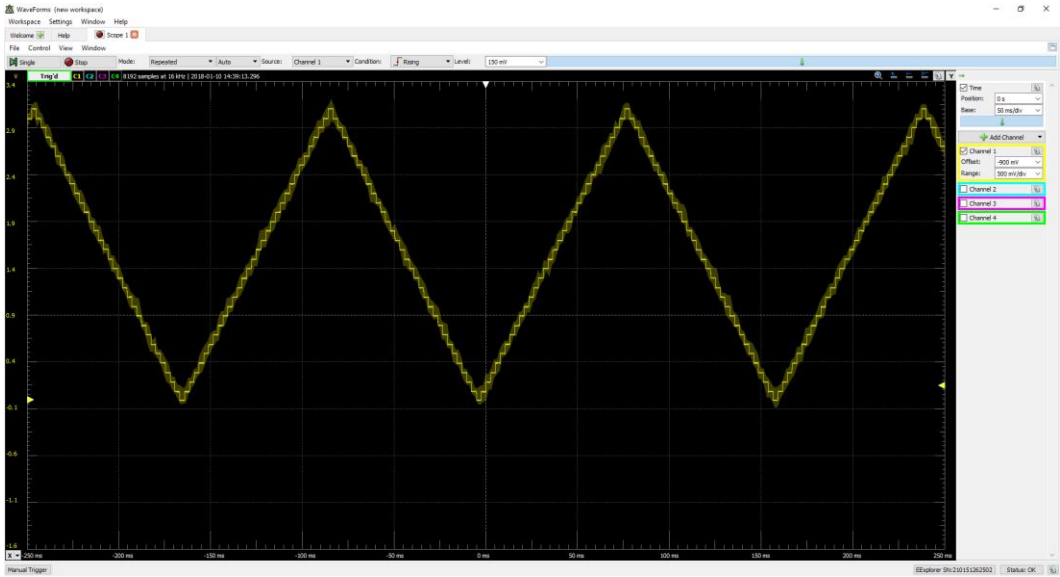
```
#include "mcc_generated_files/mcc.h"
#include <cp0defs.h>
#include <sys/attribs.h>
#include "coreTimerDelay.h"
//*************************************************************
//* MS DELAY
//*************************************************************
void ctDelayms(uint16_t ms)
{
    uint32_t msDelayTime, currentTickCnt;
    currentTickCnt = _CP0_GET_COUNT();
    msDelayTime = (CoreTicksPerMs * ms) + currentTickCnt;
    while((_CP0_GET_COUNT()) < msDelayTime);
}
```



IoT Development Tools for PIC32

Double click – DAC click Project – Generated Example

```
#include "mcc.h"
#include "DAC.h"
#include "DAC_example.h"
#include <stdio.h>

void DAC_example(void)
{
    float voltage = 0;

    uint16_t dacVoltage;

    printf("Ramping our voltage UP");
    while(voltage < 3){
           dacVoltage = (uint16_t) ((4096*voltage)/3.3);
        DAC_Set(dacVoltage);
        ctDelayms(1);
        voltage += 0.1;
    }
    printf("Ramping our voltage DOWN\r\n");
    while(voltage > 0){
        dacVoltage = (uint16_t) ((4096*voltage)/3.3);
        DAC_Set(dacVoltage);
        ctDelayms(1);
        voltage -= 0.1;
    }
}
```



IoT Development Tools for PIC32

Hasta Mañana