

Purpose:

This module provides a set of interface functions to perform a simplified initialization of the A/D converter system on the Tiva TM4C123GH6PM microcontroller. The initialization function must be called before any use is made of the A/D converter. After the initialization is complete, reads of the analog input pins should be made using the `void ADC_MultiRead(uint32_t results[HowMany]))` function.

How to obtain:

The library is obtained by forking from the `ME218_ADMultiLib` repository. This yields a runnable project that acts as a test harness that you can go back to at any time to check your A/D converter hardware.

Usage:

To use the A/D library in your project, copy the `ADMulti.c` file (found in the **Source** folder in the repository project) into the in Source folder in your project and the `ADMulti.h` file (found in the **Headers** folder in the repository project) into your Headers folder, then add these two files into the Project window of your μ Vision project.

The file `ADMulti.c` must be included in any project wishing to use the functions provided by this module.

The header file `ADMulti.h` should be included in any module wishing to use the functions provided by this module.

Revision History:

August 24, 2017 Updated documentation to accommodate distribution through Git. Based on code version 1.0.1
08/22/17

January 29, 2017 First release of formal documentation. Based on c version 1.0 xx/xx/xxx

Initialization**Function:**

```
bool ADC_MultiInit(uint8_t HowMany)
```

Parameters:

```
uint8_t HowMany
```

A number between 1 and 4 indicating the number of Analog inputs you would like to configure.

Returns:

```
bool
```

True = Success; False = HowMany outside the legal range.

Description:

Initializes The A/D converter and the associated Port E pins for use in measuring analog inputs in the range of 0-3.3V. Once this function is called, the Port E pins configured for analog input can no longer be used for digital functions.

Notes:

If you ask for 1 channel, PE0 will be configured as an the analog input.

If you ask for 2 channels, PE0 & PE1 will be configured as the analog inputs.

If you ask for 3 channels, PE0, PE1 & PE2 will be configured as the analog inputs.

If you ask for 4 channels, PE0, PE1,PE2 & PE3 will be configured as the analog inputs.

Usage:

```
ADC_MultiInit(4); // please excuse the magic number
```

This call would initialize the A/D converter and port E pins 0-3 (4 inputs) as analog inputs.

Analog Pin Read

Function:

```
void ADC_MultiRead(uint32_t results[HowMany]))
```

Parameters:

a pointer to the first element of an array of type uint32_t with a number elements matching the number of channels initialized in the call to ADC_MultiInit().

When writing your code, simply use the name of the results array as the parameter to this function. For this syntax to work, the results array needs to be declared as an array, even if it has only one element.

Returns:

Nothing

Description:

Reads the A/D conversion result (12 bits, 0-4095 corresponding to 0-3.3V) from the appropriate A/D converter register(s) and copies the values into the elements of the results array that was passed as the parameter.

Notes:

After this call, results[0] will contain the conversion result from PE0. If more than one channel was configured, then the results will be available as follows: PE1 in results[1], PE2 in results[2], PE3 in results[3].

Usage:

```
// magic number used solely for clarity in the documentation. Do not do this in your code
uint32_t ConversionResults[3];
ADC_MultiRead(ConversionResults )
```

Assuming that the A/D converter had been initialized with ADC_MultiInit(3); this call would read the analog values on PE0, PE1 & PE2 and copy the A/D conversion results into the first 3 elements of the ConversionResults array.