

# Subsystem Design

## ADC to UART

### Description

The ADC to UART subsystem example demonstrates how to use the internal ADC to convert an analog signal into a digital representation and transfer the result through UART. The example configures the MCU to act as an external ADC and send raw ADC data through UART. Optionally the MCU can also preprocess the data then send it through I2C. [Download the code for the ADC to UART example.](#)

The following figure shows a block diagram of the system.

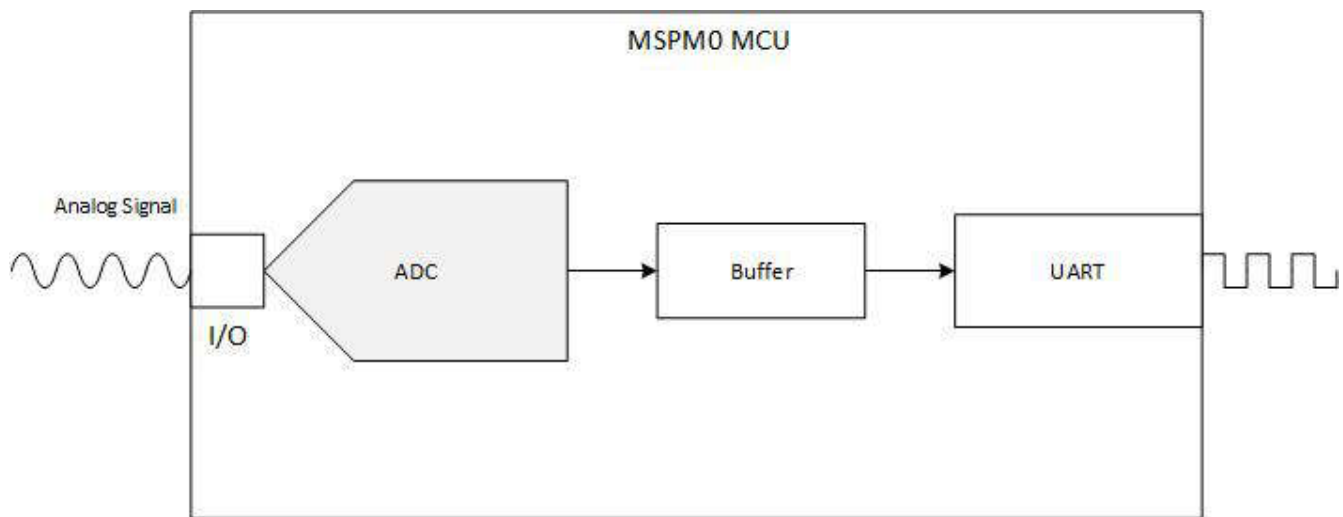


Figure 1-1. Subsystem Functional Block Diagram

### Required Peripherals

The application requires the internal ADC and 1 instance of UART

Sub-block Functionality	Peripheral Used	Notes
Analog Signal Capture	ADC	Called ADC12_0_INST in code
Sending ADC data	UART	2 UART transactions are done to send the full ADC data.

### Compatible Devices

Based on the requirements in the table above, the compatible devices are listed below. The corresponding EVM may be used for quick evaluation.

Compatible Devices	EVM
MSPM0Lxxx	<a href="#">LP-MSPM0L1306</a>
MSPM0Gxxx	<a href="#">LP-MSPM0G3507</a>

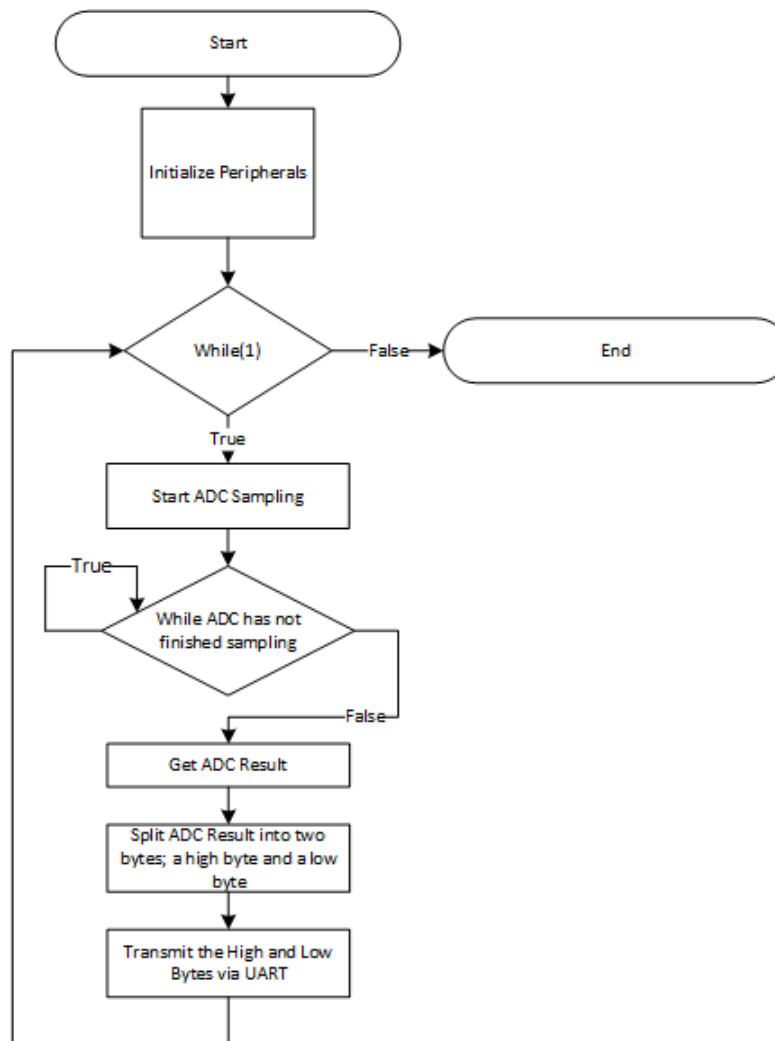
## Design Steps

1. Determine the configuration for the ADC including reference source, reference value, and sampling rate based on the expected analog input and design requirements.
2. Configure the ADC in SysConfig based on requirements in the previous step.
3. Configure the UART peripheral in SysConfig, setting the UART to the intended baud rate and other UART options for the intended communication.
4. Write Application Code to transfer the ADC data from the memory registers to the UART. See the [Software Flowchart](#) for an overview or view the code directly.

## Design Considerations

1. Max Sampling Speed: The sampling speed of the ADC is based on input signal frequency, analog front end, filters or any other design parameters that affect sampling.
2. ADC Reference: Choose the reference to align with the expected max input to utilize the full scale range of the ADC.
3. Clock Settings: The clock source determines the total time for sample and conversion time. The clock divider in tandem with the SCOMP setting determine the total sampling time. SysConfig sets the appropriate SCOMP depending on the sampling time setting.
4. UART configurations can be adjusted depending on the UART system, such as parity, baud rate, and more.

## Software Flowchart



## Application Code

The UART peripheral sends data in packets of 8 bits at a time. The ADC module stores the data into a 16-bit register. To transmit the data through the UART peripheral, the ADC data must be split into high and low bytes. The high byte contains the upper 8 bits while the low byte contains the lower 8 bits. Below is the code to split the ADC result and transmit the data through UART.

```
gADCResult = DL_ADC12_getMemResult(ADC12_0_INST, DL_ADC12_MEM_IDX_0);
uint8_t lowbyte = (uint8_t)(gADCResult & 0xFF);
uint8_t highbyte = (uint8_t)((gADCResult >> 8) & 0xFF);
DL_UART_Main_transmitData(UART_0_INST, highbyte);
DL_UART_Main_transmitData(UART_0_INST, lowbyte);
```

## Additional Resources

- [Download the MSPM0 SDK](#)
- [Learn more about SysConfig](#)
- [MSPM0L1306](#)
- [MSPM0G3507](#)
- [MSPM0 ADC Academy](#)
- [MSPM0 UART Academy](#)

## Revision History

DATE	REVISION	NOTES
December 2023	*	Initial Release

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2023, Texas Instruments Incorporated