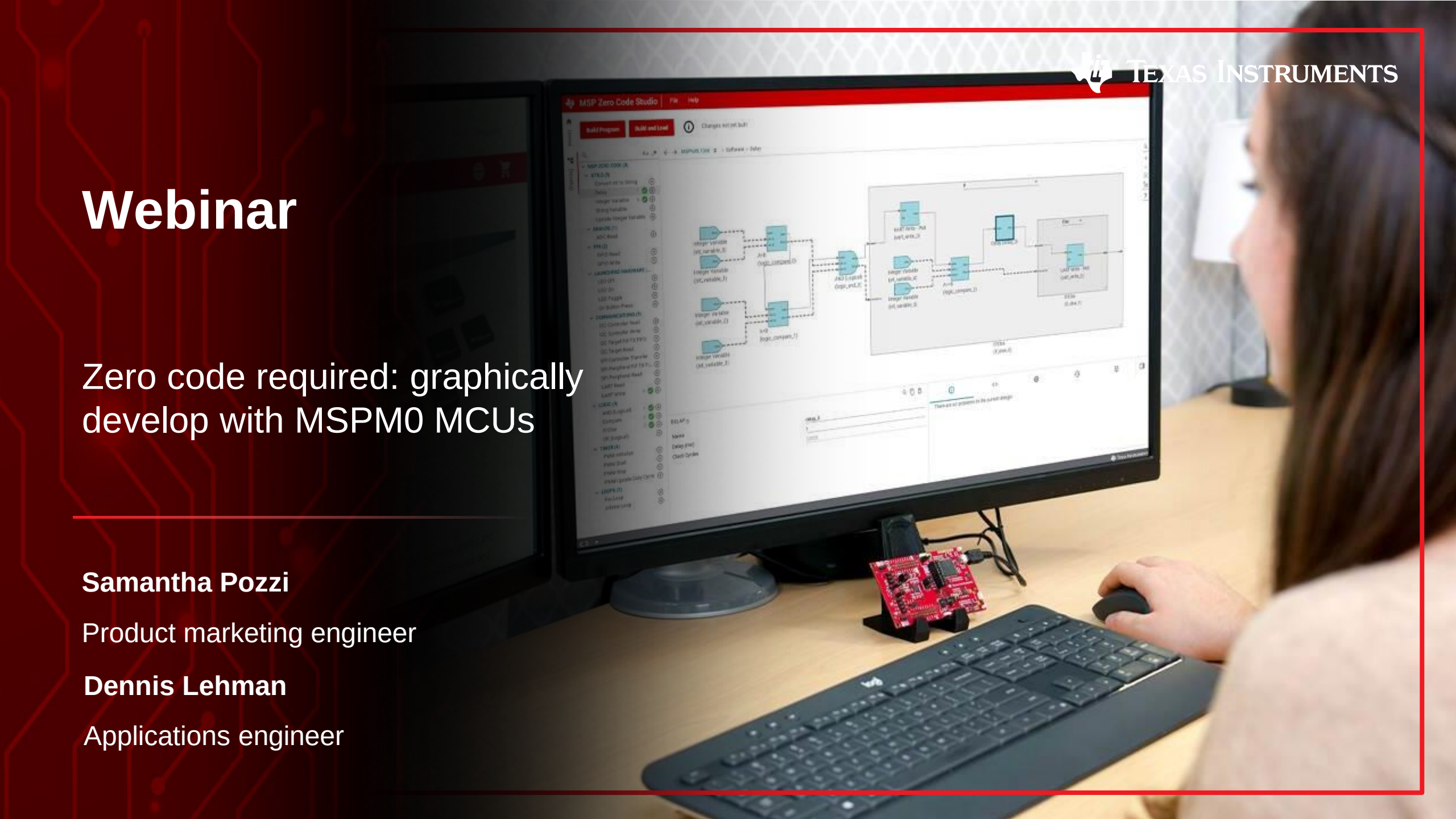
Webinar

Zero code required: graphically develop with MSPM0 MCUs



Samantha Pozzi

Product marketing engineer

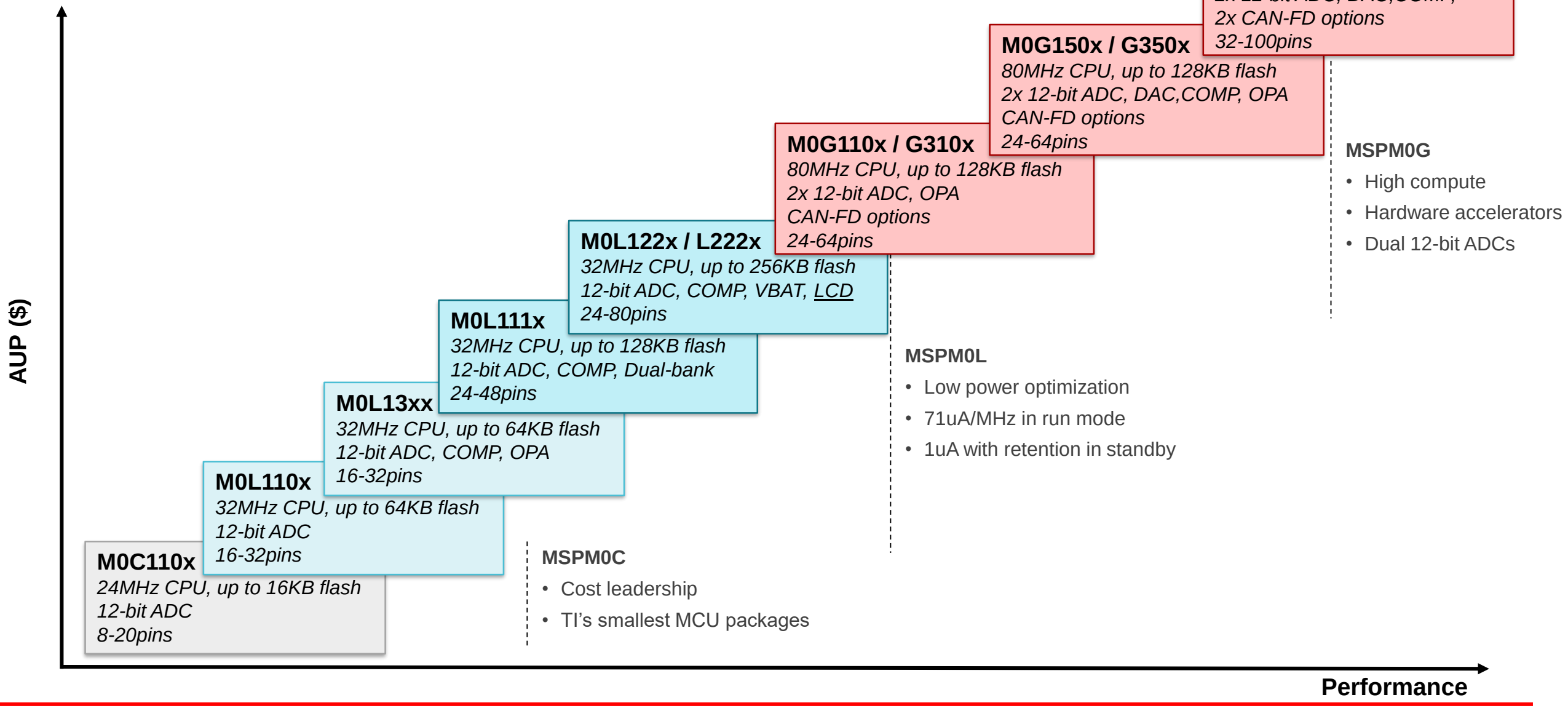
Dennis Lehman

Applications engineer

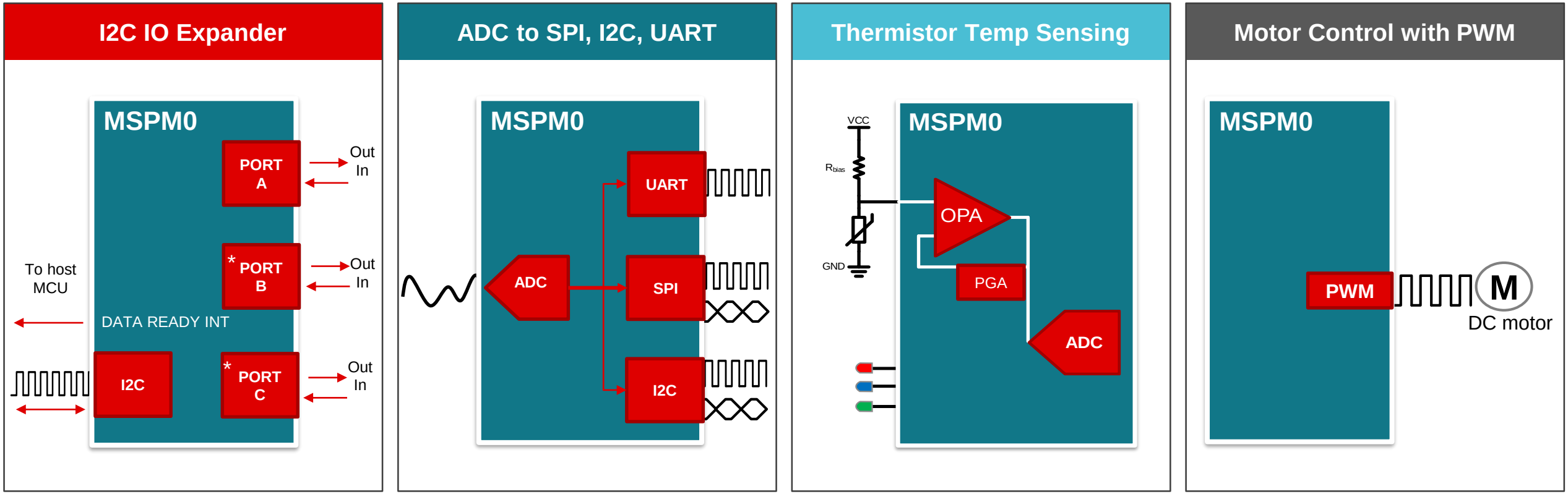
Agenda

- MSPM0 general purpose MCU portfolio
- Traditional code development for creating applications for MCUs
- Benefits of using MSP Zero Code Studio
- Overview of tool functionality and loaded examples
- Example use cases and applications
- Application code demos in MSP Zero Code Studio
  - **Demo #1**: Basic LED toggle
  - **Demo #2**: Conditional logic example
  - **Demo #3:** Motor control with PWM

MSPM0 microcontrollers



MSPM0+ MCU | Broad range of applications



- As general purpose MCUs, the MSPM0 portfolio has potential and flexibility to be designed into a wide range of applications and end equipments
- MCU application is largely defined by the firmware

Traditional code development | Embedded software

MSPM0 SDK: Optimized, robust, and flexible embedded software

Operating Systems

RTOS		Schedulers
FreeRTOS	Zephyr	Light task scheduler

Libraries and Subsystems

Data	Communication	Sensing	Control
EEPROM emulation	LIN library	Battery management	Brushed DC
Safety	CAN library	Smoke detection	Stepper
IEC 60730	SMBus	Electricity metering	Sensored trap
Security	PMBus	Occupancy detect	Sensorless FOC
Secure boot	DALI	Pulse oximeter	**Processing**
Logic	OneWire	Blood pressure	IQ-MATH
Power sequencing	SENT	Temperature	FIR/IIR filters
Digital muxing	UART to I2C bridge	Analog to serial	CMSIS-DSP

Device Support

Hardware Abstraction	Device Boot Features
TI Drivers (high level abstraction)	Boot loaders (UART, I2C, SPI, CAN)
DriverLib (low level abstraction)	Flexible debug security modes

Comprehensive Examples

Peripheral driver example projects
Digestible driver examples to demonstrate basic functionality

Subsystem example projects
Connect modules together to implement simple functions

Library example projects
Demos for using the libraries provided in the software development kit

RTOS example projects
Demos for using drivers in the context of a real-time operating system

Third Party Support

Automotive software
Solutions for LIN and CAN protocol stacks

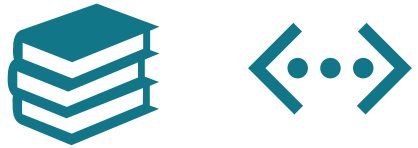
Emulation and production
Programming tools from Elprotronic, Segger, Lauderbach and more

IAR Embedded Workbench

KEIL™ Tools by ARM

Application code is developed in CCS or 3P environment

- Most flexibility for application designs
- Enables the development of complex applications



Resources for development

- Accelerate development with readily available application based resources
- MSPM0 SDK includes software libraries, middleware, example codes, drivers, subsystems, and more!

Barrier to Entry

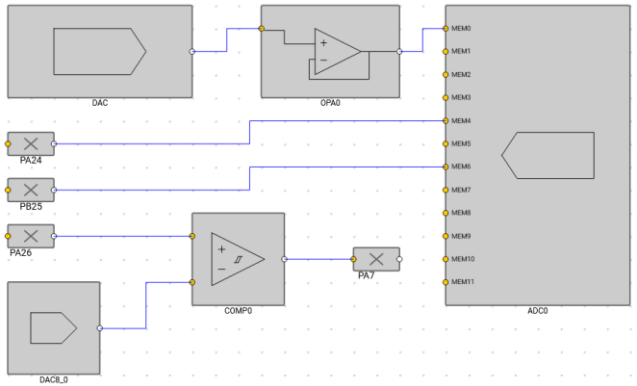
- Ideal for experienced developers or those with time/resources to learn a new environment

Traditional code development | Graphical tools

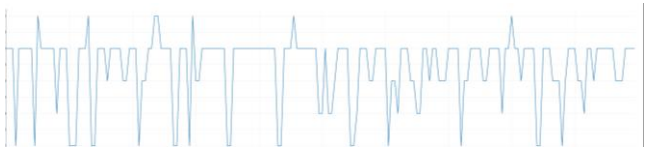
Analog design

ti.com/tool/mspm0-analog-configurator

Drag-and-drop ADCs, DACs, CMPs, amps



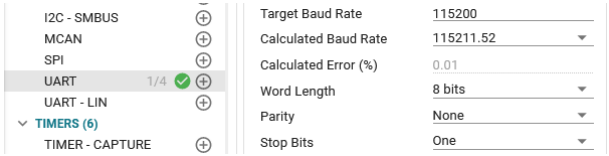
View & log data converter outputs in real time



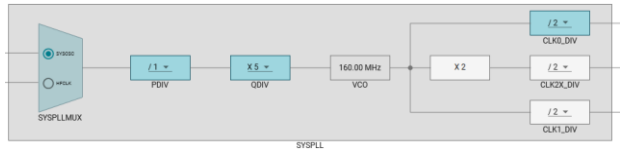
System configuration

ti.com/tool/sysconfig

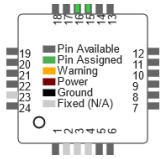
Select and set up peripherals

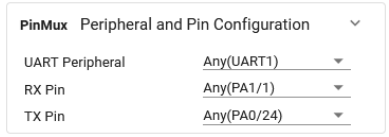


Configure & optimize clock system



Solve pin multiplexing challenges





Speedy development

- Users can visually configure peripherals and pins as well as enable analog signal chain using a MSPM0 device
- Accessible to a wider audience as little experience / training required
- Standard tools for the industry

Limited design capabilities

- User need to continue application code creation outside of the visual development tools

MSP Zero Code Studio | Graphical configuration tool

Place building blocks

Drag-and-drop visual interface:

1. Add a wide array of peripherals: digital, analog, and serial port blocks
2. Add mapping (processing) blocks: logic, loop blocks
3. Connect with sequence & data lines

Guided experience

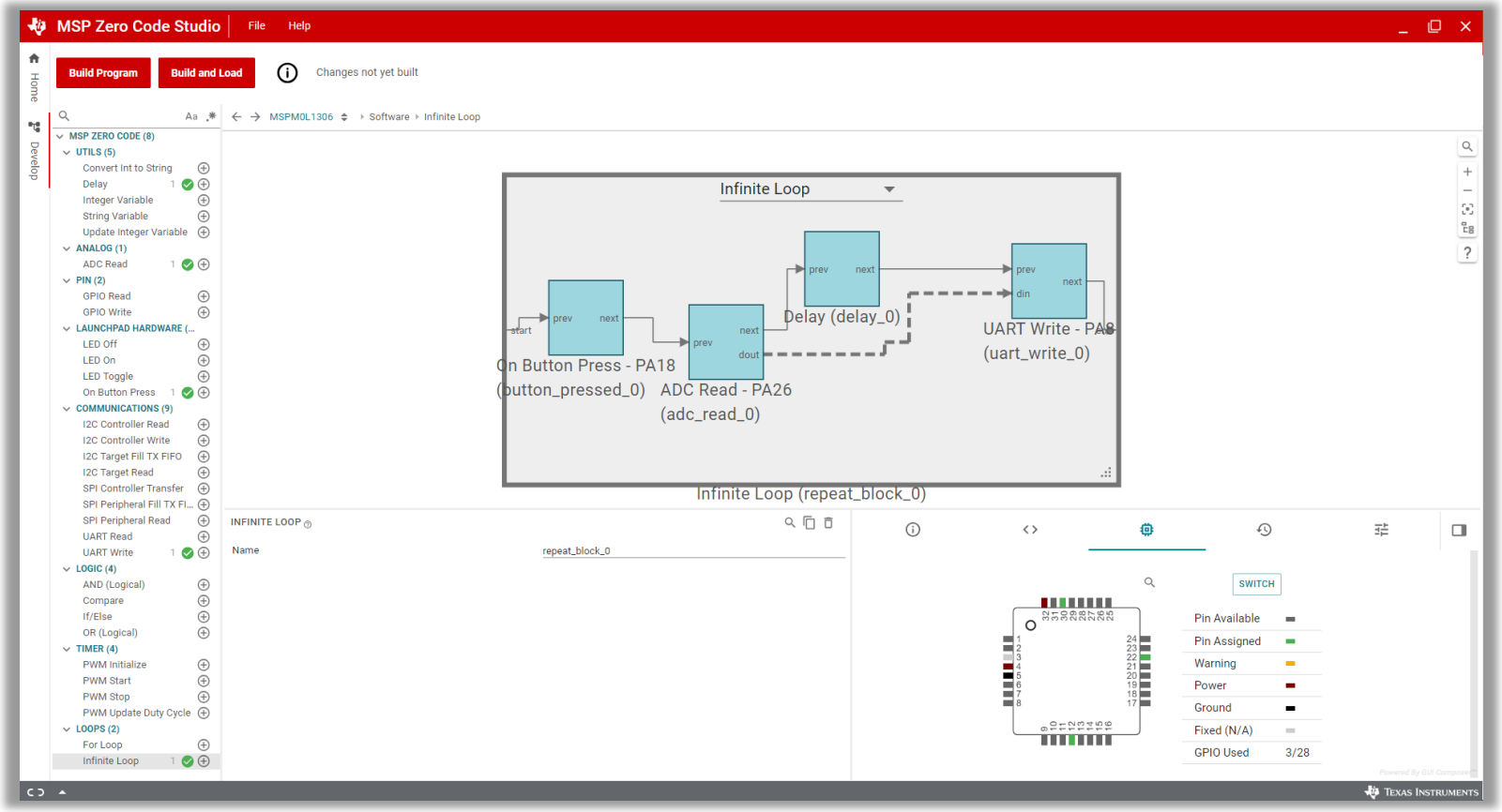
Choose an demo /empty project to start

1. Follow on-screen prompts to start design

Deploy to device

Run application on a device:

1. Tool generates exportable ‘C’ code
2. Code builds behind the scenes
3. Tool downloads project to target



MSP Zero Code Studio | Accelerate software development

TI's newest graphical development environment

ti.com/tool/MSP-ZERO-CODE-STUDIO

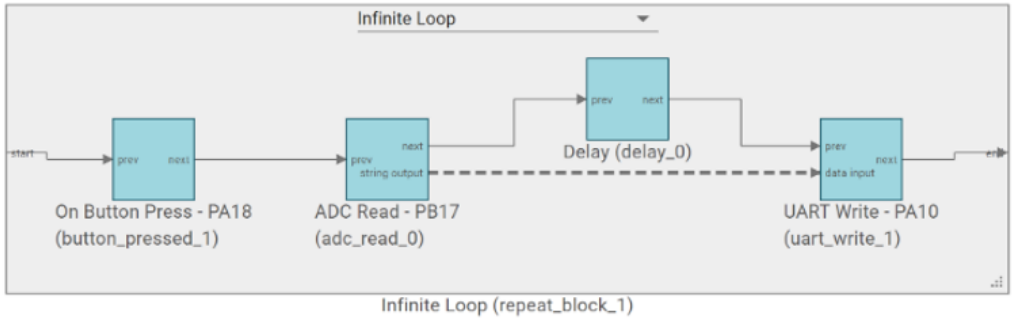
Welcome to MSP Zero Code Studio

- Configure, develop, and run MCU applications
- Compile and download your application to your MSPM0 device
- Support for all MSPM0 microcontrollers and EVMs.

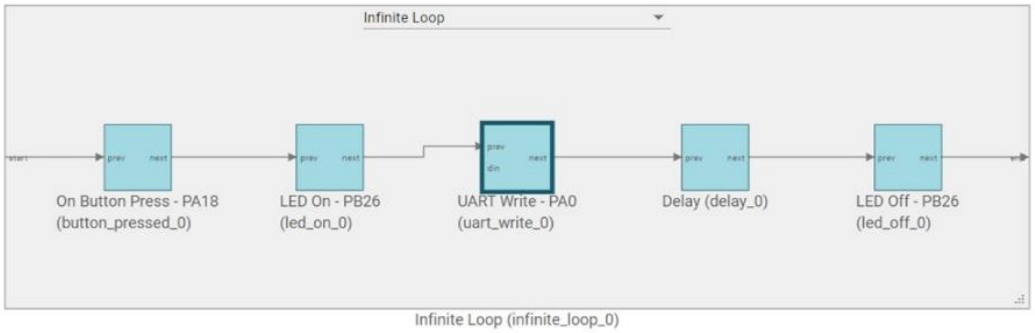
Design MCU applications in minutes

- All experience levels supported
- Blocks added by drag-and-drop action
- Example applications provided
- Zero coding, no IDE and no TRM study needed

ADC Sample to UART write Example



UART Hello World Example



MSP Zero Code Studio | Configuration options

MSP Zero Code Studio allows users a wide array of configuration options, including:

Device Peripherals

Configuration option	M0G	M0L	M0C
PWM	✓	✓	✓
GPIO	✓	✓	✓
I2C	✓	✓	✓
SPI	✓	✓	✓
UART	✓	✓	✓
ADC	✓	✓	✓
COMP	✓	✓	
OPA	✓	✓	

Logic Concepts

Configuration option	M0G	M0L	M0C
AND (Logical)	✓	✓	✓
Compare	✓	✓	✓
If/Else	✓	✓	✓
OR (Logical)	✓	✓	✓

Loops

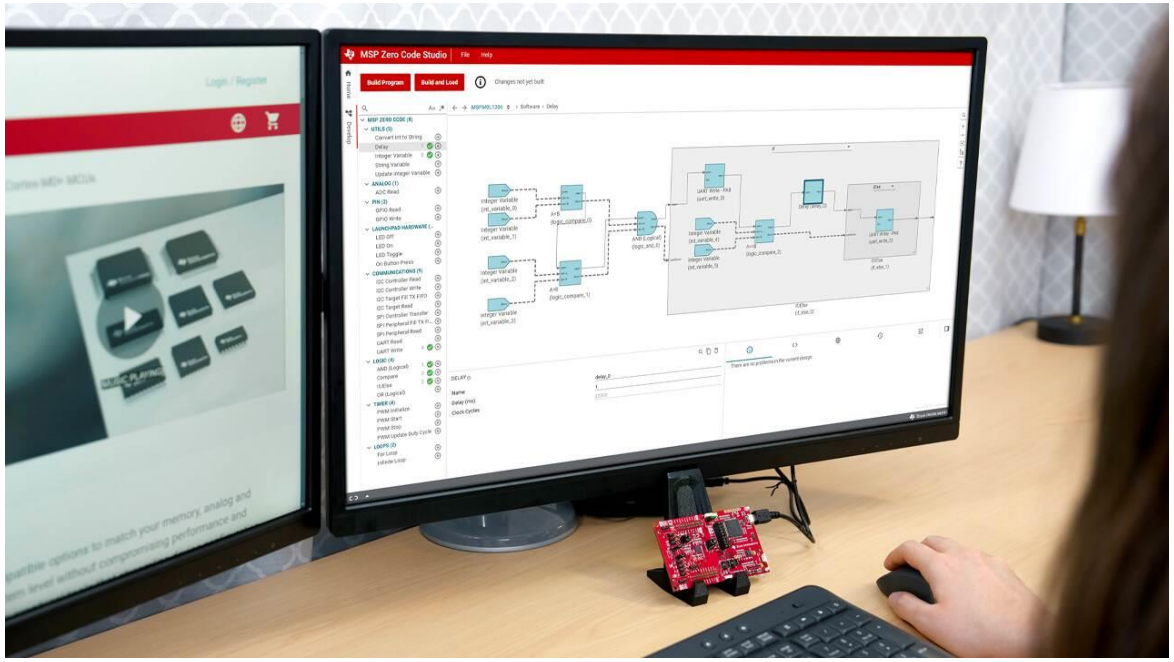
Configuration option	M0G	M0L	M0C
For Loop	✓	✓	✓
Infinite Loop	✓	✓	✓

Variables

Configuration option	M0G	M0L	M0C
Lookup Tables	✓	✓	✓
Integers	✓	✓	✓
Strings	✓	✓	✓

MSP Zero Code Studio | Use cases & applications

- Use cases
  - Quickly evaluate MSP devices
  - Program devices without requiring any software background
  - Single-App solution – no need for installing multiple IDEs or tools.
  - Create simple designs in minutes
- Example applications
  - ADC with I2C interface
  - I2C/SPI to UART
  - PWM Motor Control



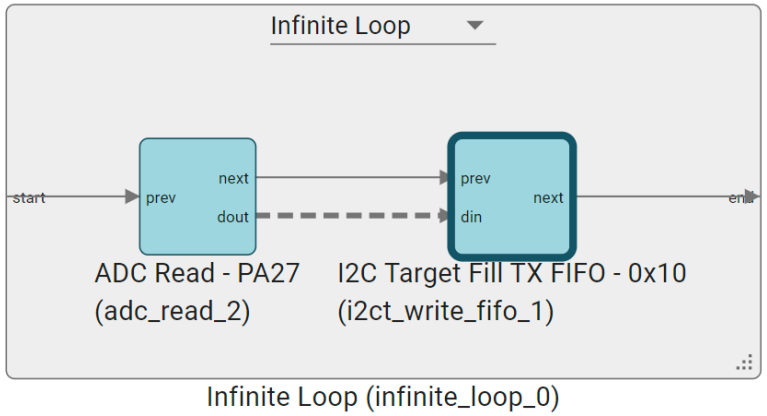
Temperature transmitter

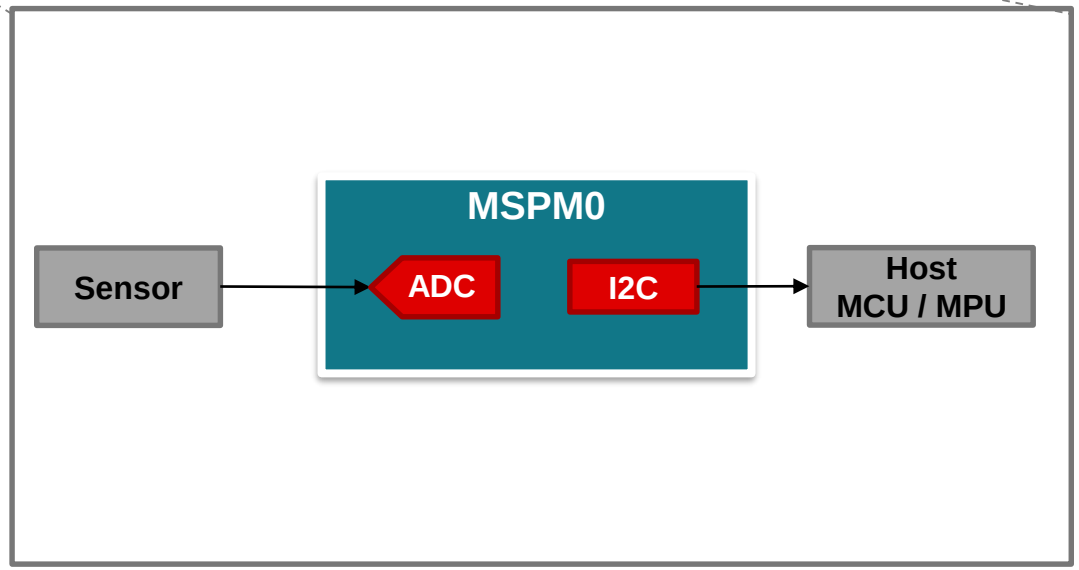
MSP Zero Code Studio configuration

MSPM0 Application

- Integrated ADC converts the reading from the pressure sensor
- I2C channel communicates reading with the host MCU / MPU

Zero Code Configuration





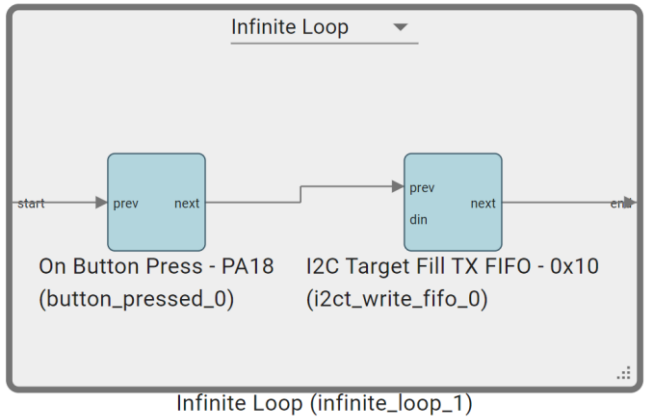
Coffee machine

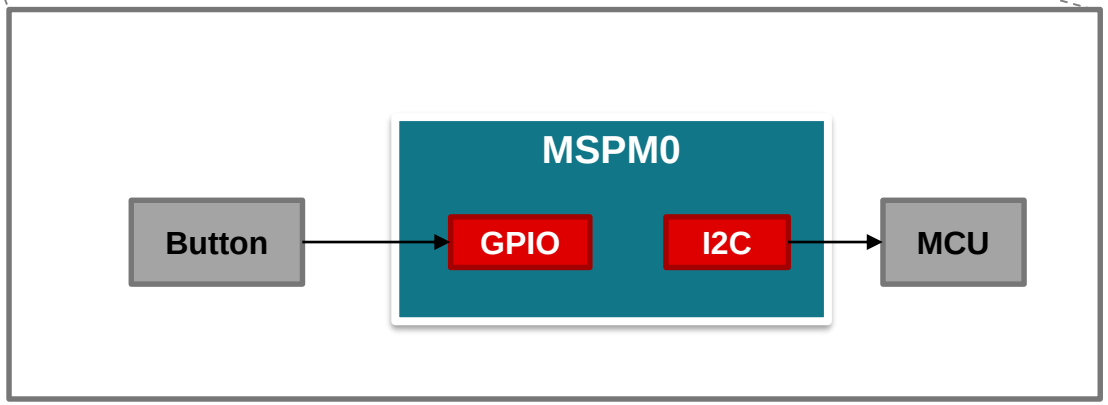
MSP Zero Code Studio configuration

MSPM0 Application

- Button press is read through GPIO, processes and sent via I2C to the host MCU

Zero Code Configuration





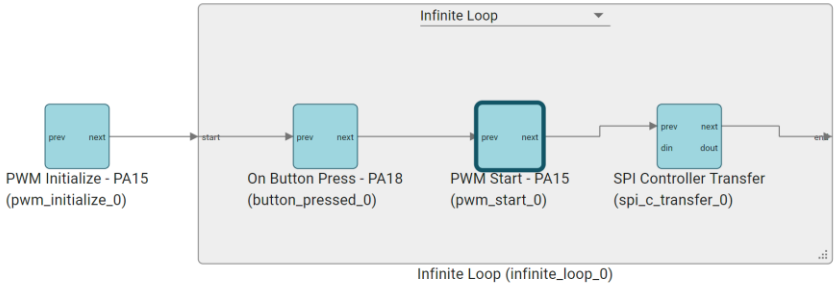
Electronic smart lock

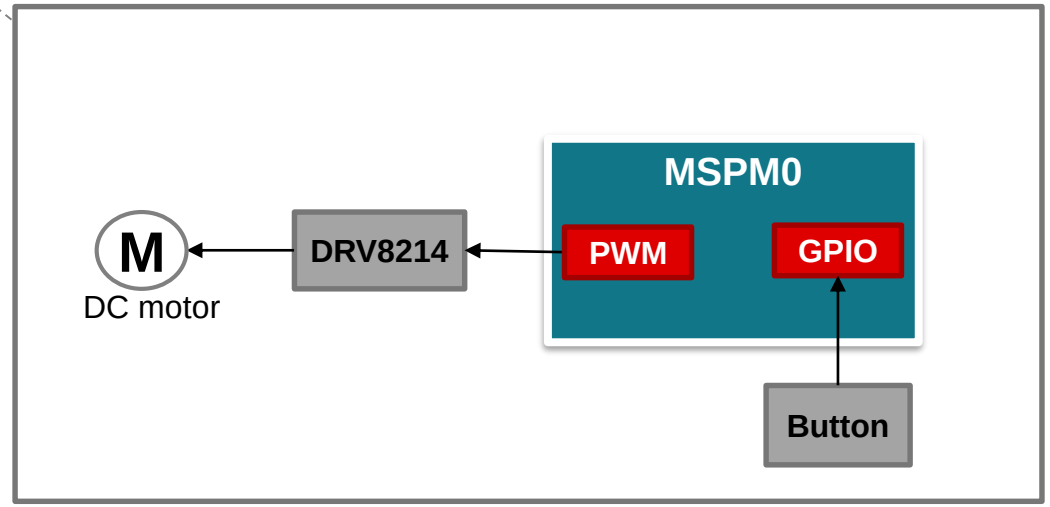
MSP Zero Code Studio configuration

MSPM0 Application

- Button press is read through GPIO
- General-purpose timer generates PWM to send to motor driver (DRV8214)

Zero Code Configuration

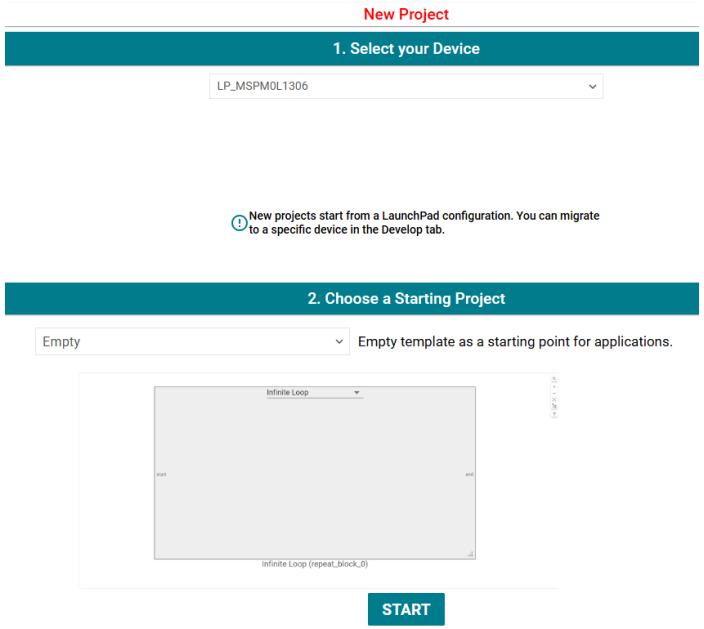




MSP Zero Code Studio | How to use

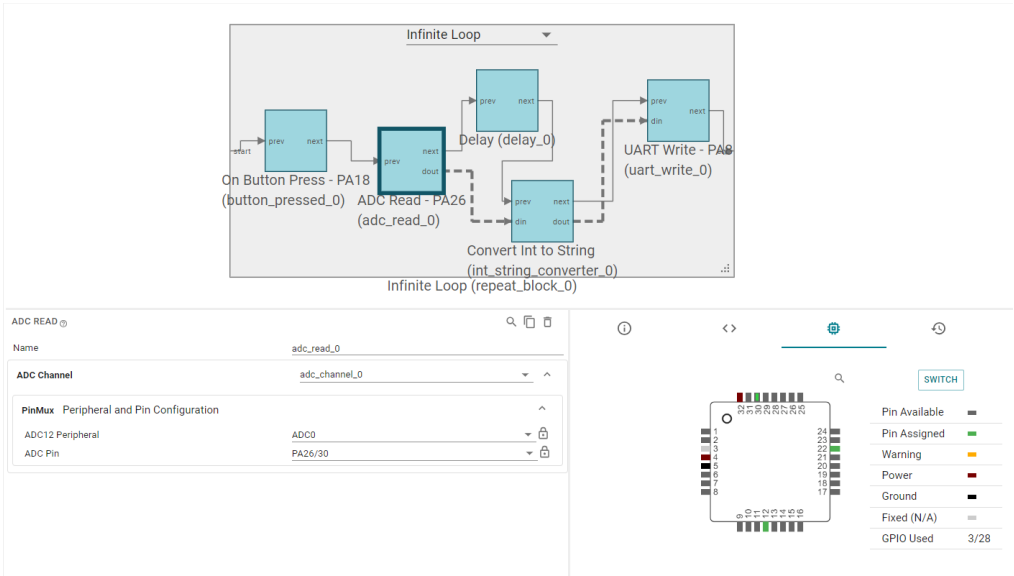
Start process

- Start a new project
- Load exiting project



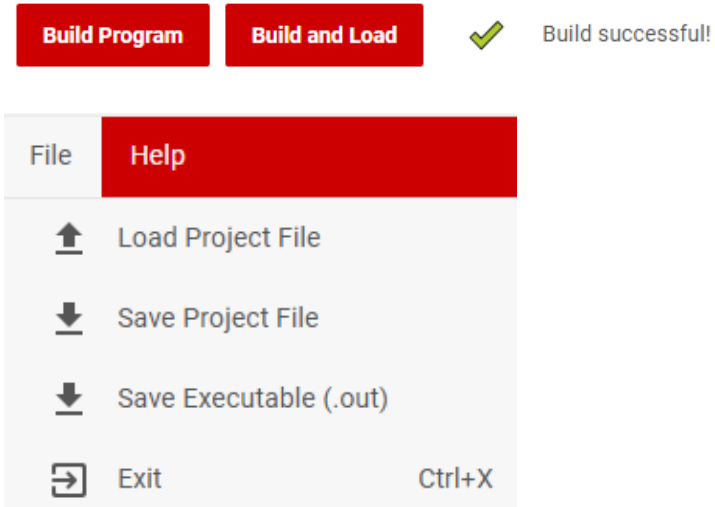
Develop process

- Add/ remove/ configure blocks
- Add blocks into loop
- Sequence / data connection
- Generated files
- Pin usage and device selection



Load process

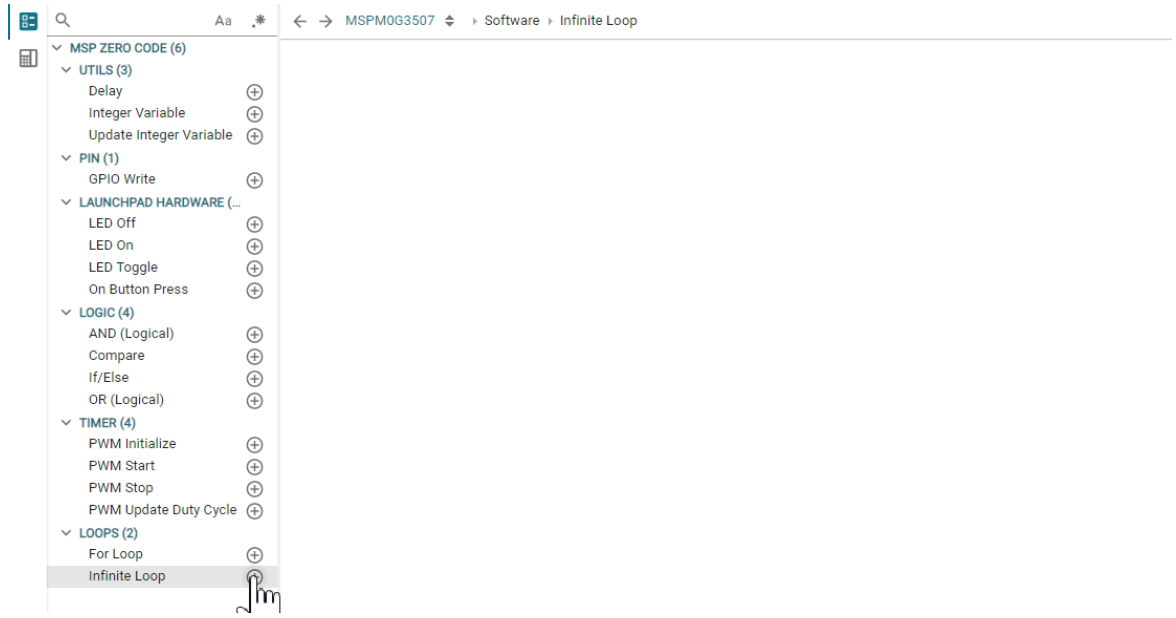
- Only build
- Build and Load
- Save Project file
- Save Executable (.out)



Application code demos in MSP Zero Code Studio

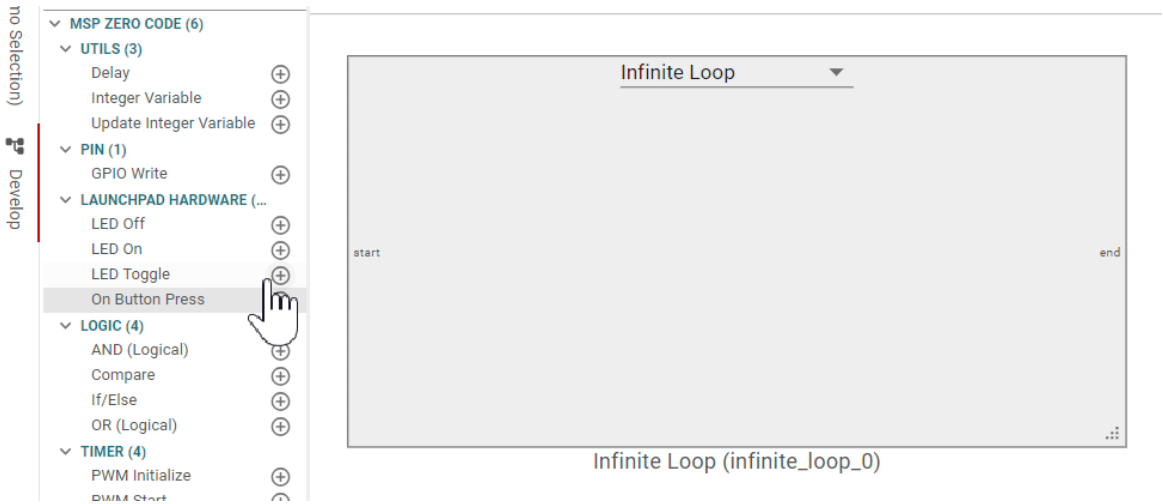
Example 1 – Basic LED toggle

- Learn the basics of **MSP Zero Code Studio** and configure an LED to toggle every time a button is pressed.
  - This example will walk you through how to add and connect blocks, and also toggle the LaunchPad's on-board **LED** every time the LaunchPad **button** is pressed.



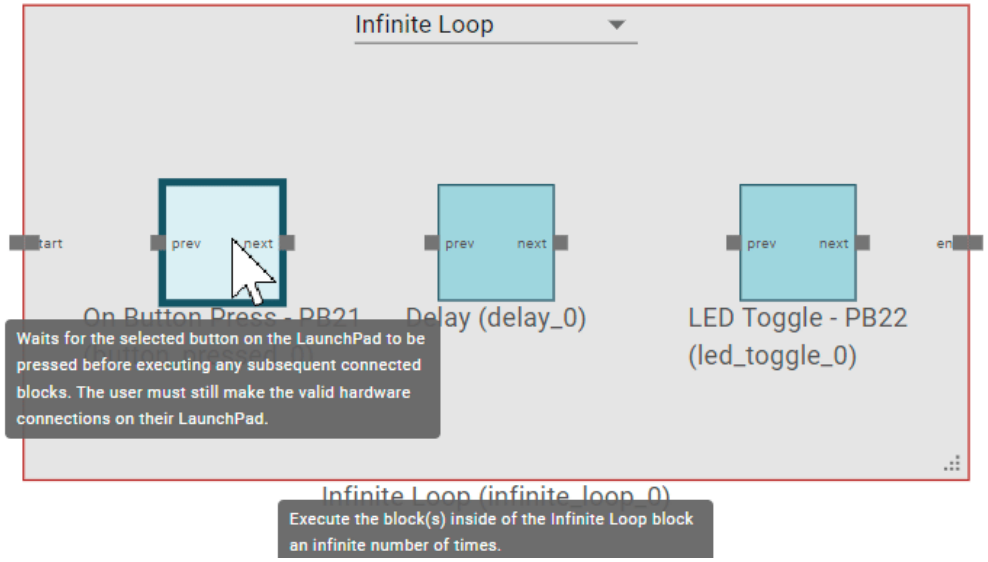
Example 1 – Basic LED toggle

- Step 1: Adding blocks
  - Highlight the **Infinite Loop** block by selecting it in the design - highlighting this block will ensure that the next block is added inside of the loop. Blocks inside of an **Infinite Loop** will continue to be repeated in the order they have been defined.
  - Add the **On Button Press** block by clicking on the (+) button next to its name on the left pane. This block will wait for the user to press the LaunchPad button before executing the next block in its sequence.
  - Add a **Delay** block as well as a **LED Toggle Block**. This block will configure the selected on-board LaunchPad LED to toggle.



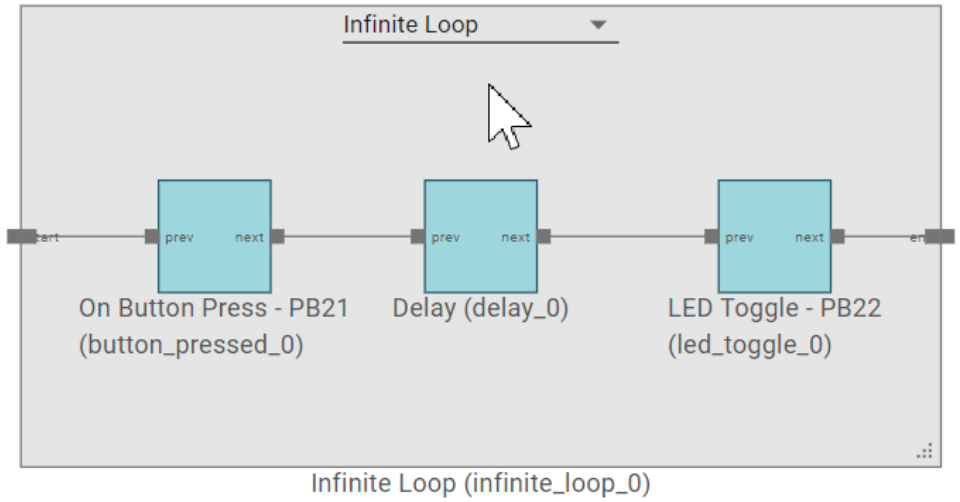
Example 1 – Basic LED toggle

- Step 2: Making connections
  - Connect the **start** port of the **Infinite Loop** block to the **prev** port of the **On Button Press** block.
  - This is called a **sequence connection,** and it allows you to specify in which order blocks will be executed.
  - Sequence connections are represented by a continuous arrow line that connects two blocks.
  - Continue by connecting the **On Button Press** block to the **Delay** block, the **Delay** block to **LED Toggle**, and finally the **LED** toggle to the **end** port of the loop.



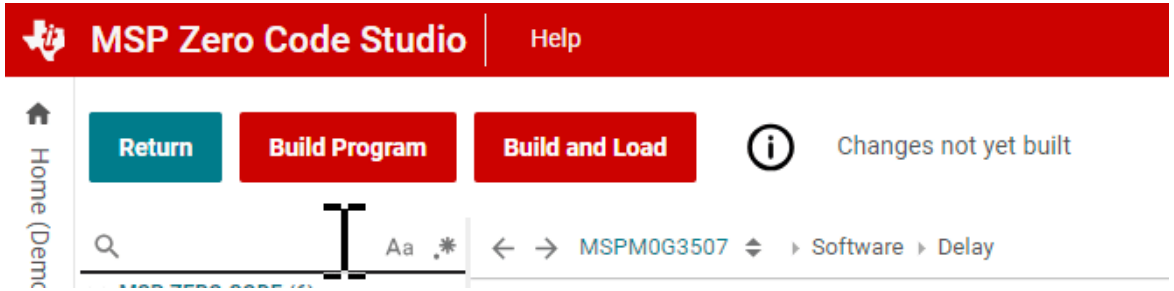
Example 1 – Basic LED toggle

- Step 3: Block configuration
  - Select the **Delay** block on the design, this will open the block configuration pane on the bottom.
  - Set the **Delay (ms)** parameter to 500. This will add a brief delay to the LED Toggle action after the initial button press.
  - Select the **On Button Press** block, on the **Button Configuration** select **PA18** for the Button Pin.
  - You have now created an application that waits for the user to press a button (S1 - PA18) on the LaunchPad, and then delays for 500ms before toggling the LaunchPad's on board LED.



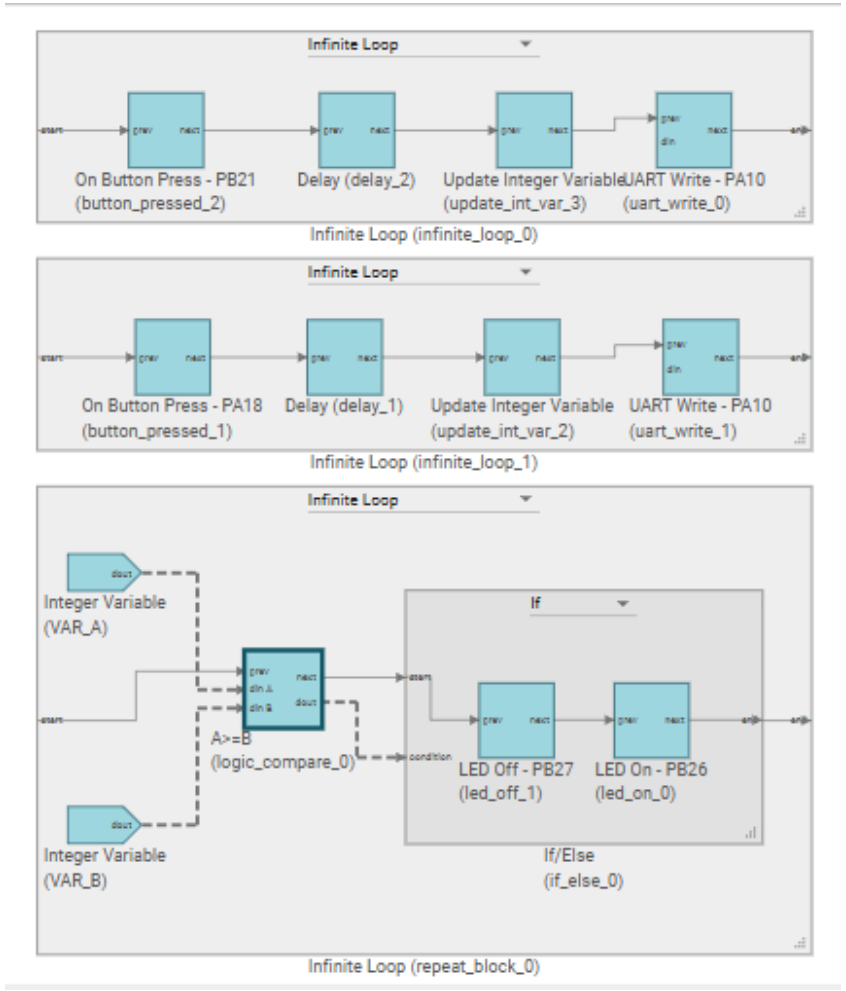
Example 1 – Basic LED toggle

- Step 4: Build and load
  - The final step is to build your program, and then load it to the Launchad.
  - Select the **Build and Load** button and wait until the program is done building. Follow on-screen prompts to load to the connected device.
  - You have succesfully developed your first **Zero Code Studio** program!



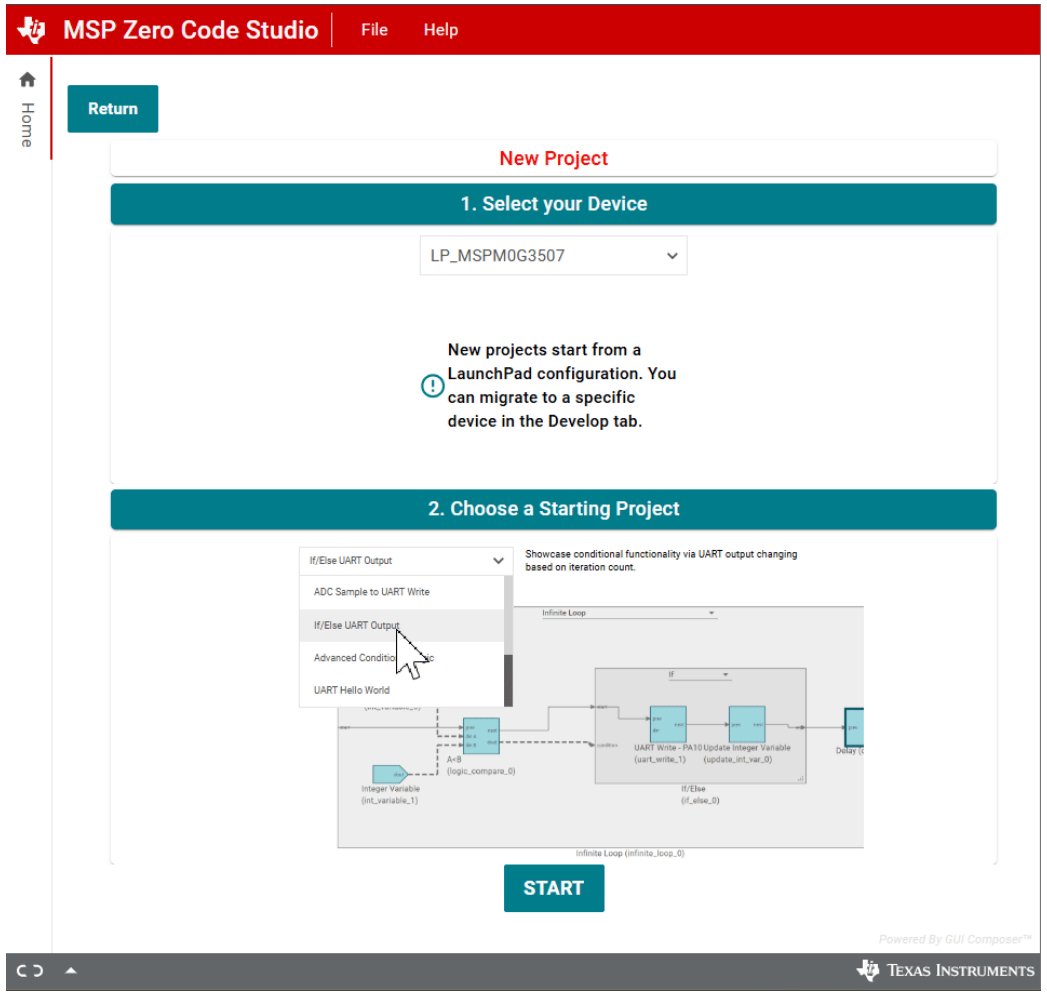
Example 2 – Conditional logic

- For this example, we will configure two variables to be compared.
  - If A>=B, the LED is set to RED
  - If A<B, the LED is set to GREEN
- The value of A is controlled by the user
  - If the user presses LaunchPad button PA18, the value is increased
  - If the user presses LaunchPad button PA21, the value is decreased



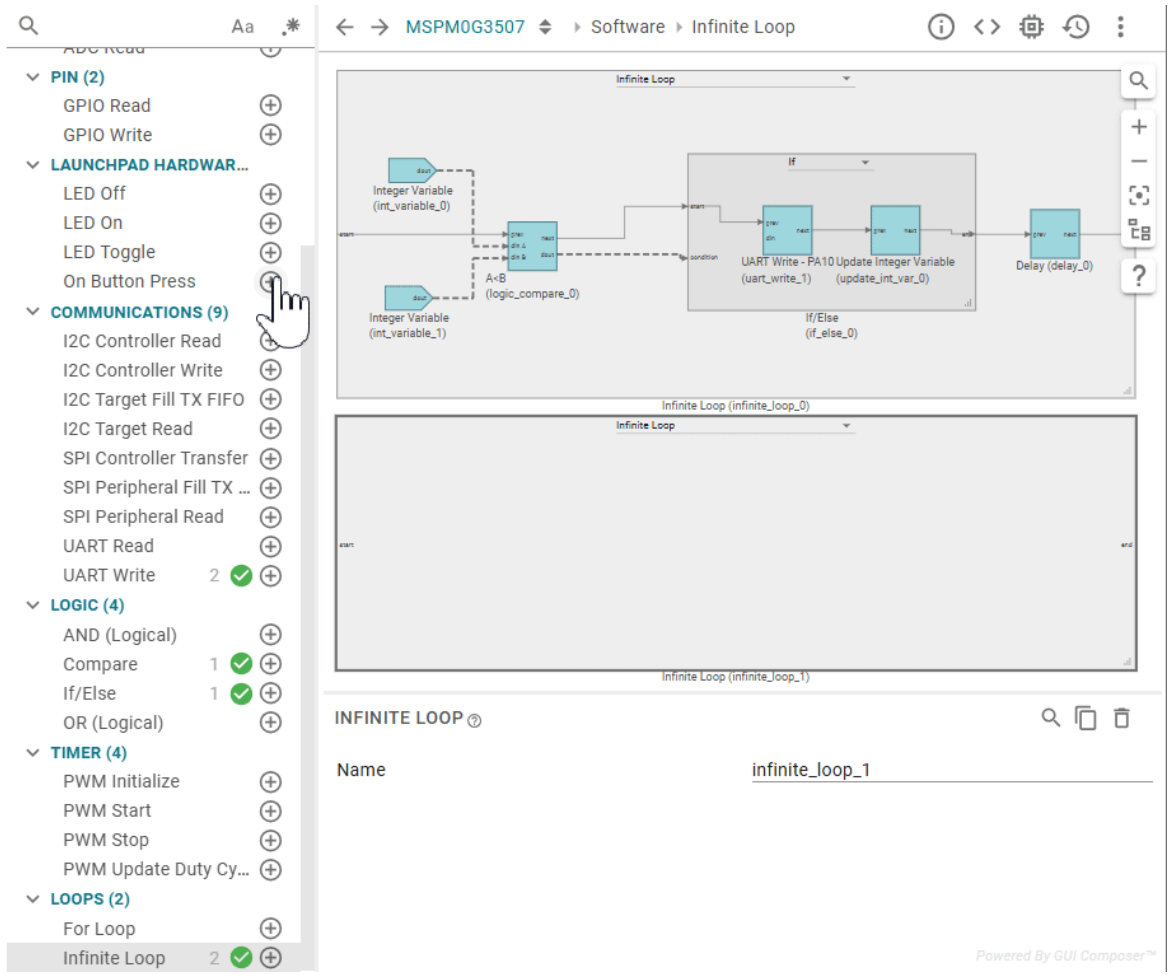
Example 2 – Conditional logic

- Step 1: Starting point
  - When starting a new project, select the *If/Else UART Output* project as a starting point
  - This starting example configures two variables that will be compared and modified. A message is sent via UART with the comparison result.
    - If A>B, A is reset to 0.
    - Otherwise, A is increased by 1.
  - We will modify this example such that the input A is controlled by the user, and so that the on-board LEDs change based on the comparison.



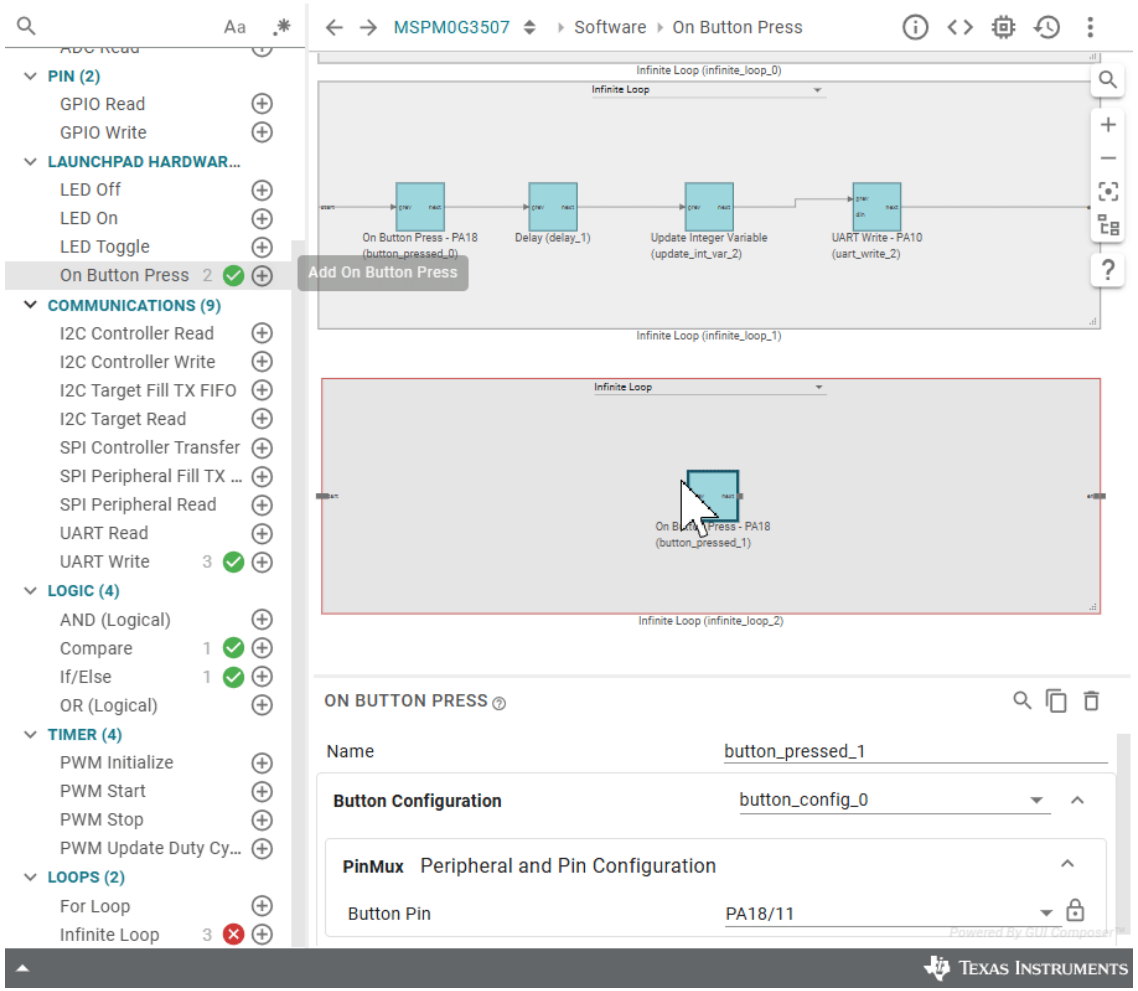
Example 2 – Conditional logic

- Step 2: Configure button (Increase)
  - Add an **Infinite Loop**
  - Connect an **On Button Press** block, set the pin to PA18
  - Connect a **Delay,** set it to 500ms
  - Connect an **Update Integer Variable** block. Make sure that the **Select Variable** field is configured to the first input (int_variable_0). Configure the block to **increment** the value.
  - Connect a **UART Write** block. Set the pin to PA10, and add a message to detail the action “INCREASE”



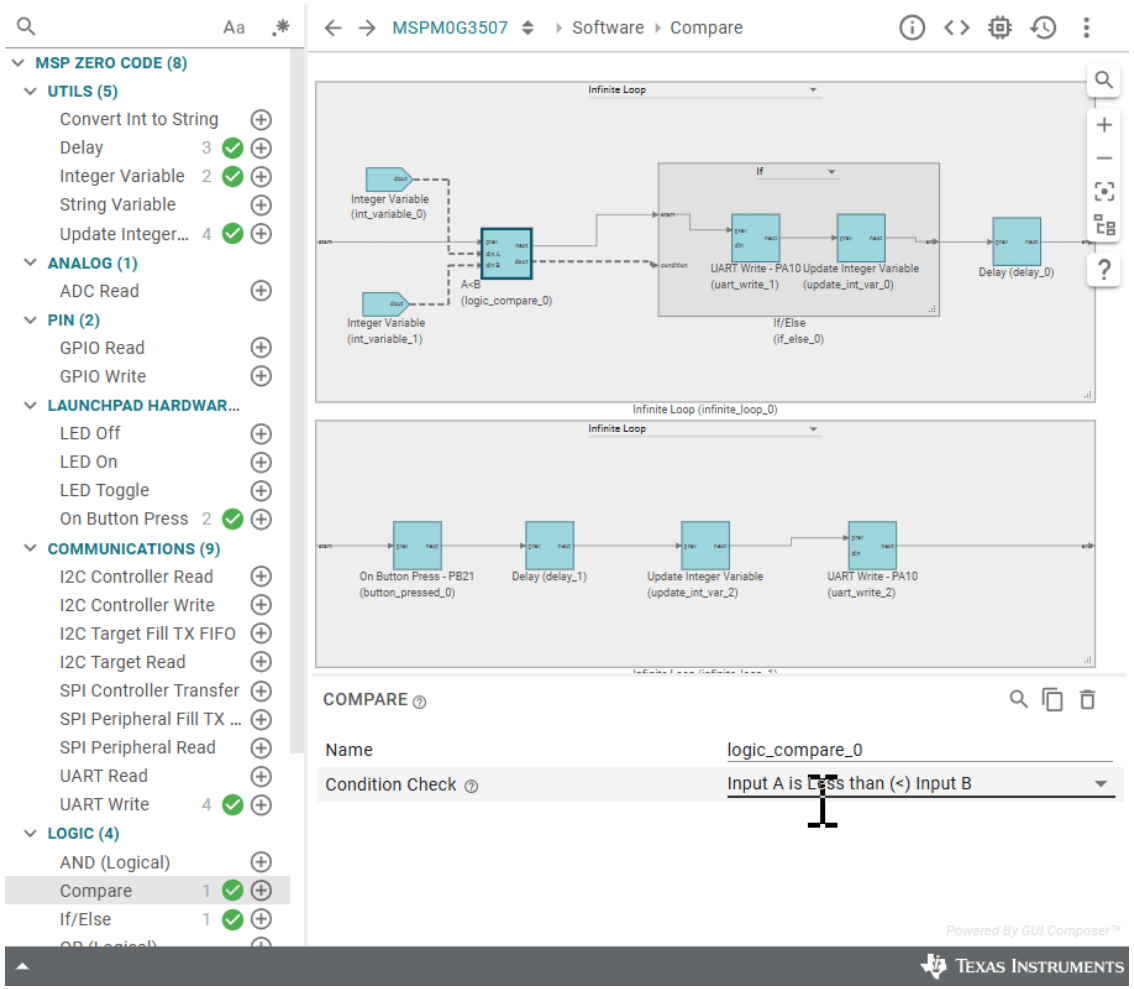
Example 2 – Conditional logic

- Step 3: Configure button (Decrease)
  - Add an **Infinite Loop**
  - Connect an **On Button Press** block. On the *Button Configuration,* select the option to *Create New+.* Set the pin to PA21.
  - Connect a **Delay,** set it to 500ms
  - Connect an **Update Integer Variable** block. Make sure that the **Select Variable** field is configured to the first input (int_variable_0). Configure the block to **decrement** the value.
  - Connect a **UART Write** block. Set the pin to PA10, and add a message to detail the action “DECREASE”



Example 2 – Conditional logic

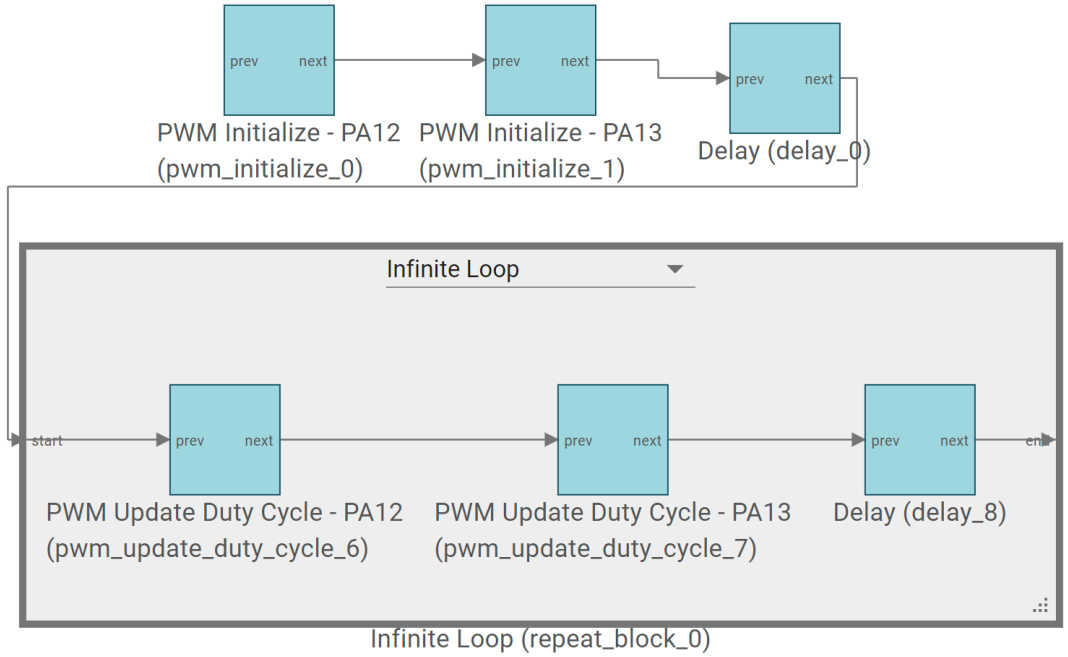
- Step 4: Modify conditions
  - Change the condition of the **Compare** block to **A>=B**.
  - **Green LED:** Select the **If/Else** block, on the *If* context:
    - Remove both of the existing blocks (UART Write, Update Integer Variable).
    - Connect an **LED Off** block. Configure it to PB27.
    - Connect an **LED On** block. On the *LED Configuration* select *Create New+.* Configure the pin to PA26.
  - **RED Led:** Select the **If/Else** block, on the *Else* context:
    - Remove both of the existing blocks (UART Write, Update Integer Variable).
    - Connect an **LED Off** block. On *Pin Configuration,* select the second LED (*led_config_1).*
    - Connect an **LED On** block. On the *LED Configuration* select the first LED (*led_config_0*).
  - Remove the final delay, and connect the **If/Else** to the end of the loop.



Example 3 – Motor control with PWM

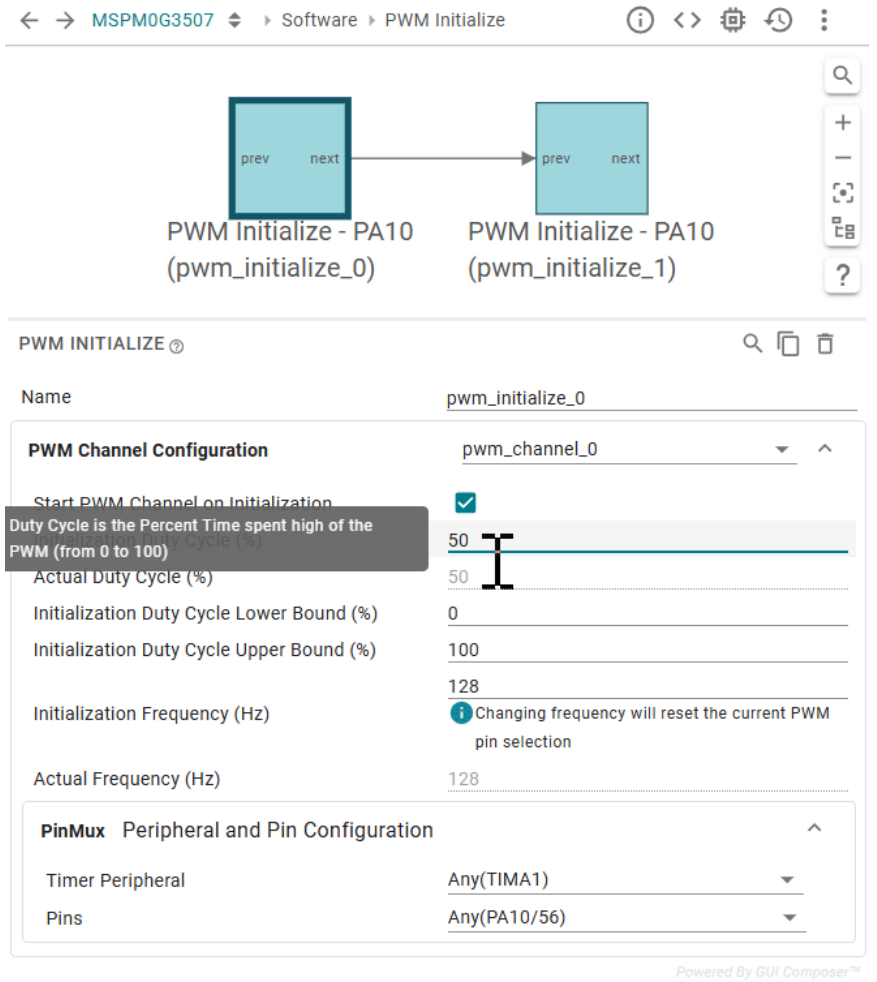
This example serves as a showcase for basic **motor control** application.

- The motors are controlled by configuring the **PWM Init** blocks, as well as using the **PWM Update Duty Cyle** block to change the speed. In this example, PA12 and PA13 are configured to output a PWM Duty Cycle that controls the motors.
- The **Delay** block is used to define some time between steps.



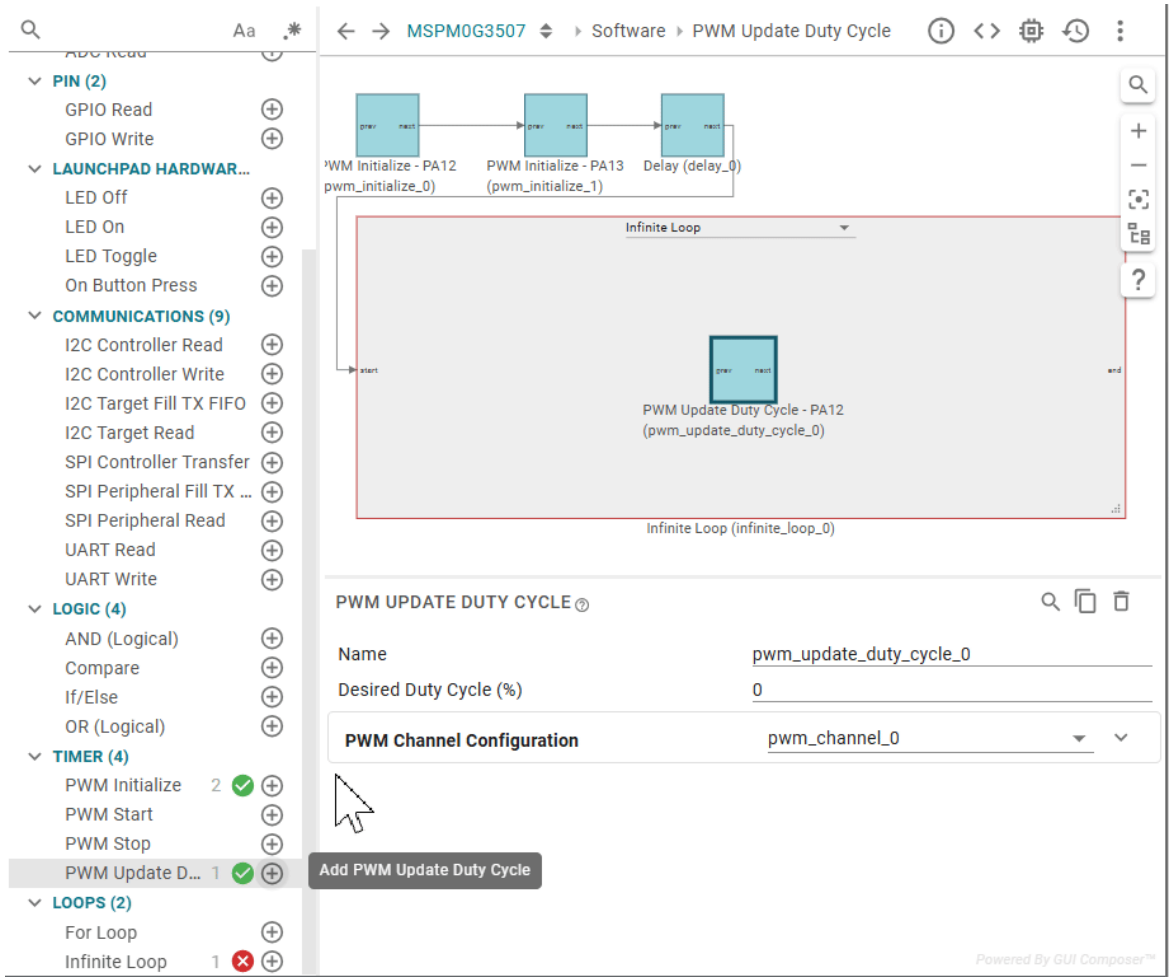
Example 3 – Motor control with PWM

- Step 1 – PWM initialization
  - Add a **PWM Initialize** block
    - Set *initialization duty cycle* to 0%
    - Set *initialization frequency* to 4000 Hz
    - Set the *pin* to PA12
  - Connect a second **PWM Initialize** block
    - On the *PWM Channel Configuration,* select *Create New+.*
    - Repeat the steps for the first block, but set the pin to PA13.
  - Connect a **Delay** block, set it to 100ms



Example 3 – Motor control with PWM

- Step 2 – Update duty cycle
  - Add an **Infinite Loop.**
  - Connect a **PWM Update Duty Cycle** block. On *PWM Channel Configuration,* select *pwm_channel_0.* Set the **Desired Duty Cycle (%)** to 60.
  - Connect a second **PWM Update Duty Cycle** block. On *PWM Channel Configuration,* select *pwm_channel_1.* Set the **Desired Duty Cycle (%)** to 60.
  - Connect a **Delay** block, and set it to 3 seconds (3000ms).



Getting started with MSP Zero Code Studio

Content type	Content title	Links & more details
Tool Page	MSP Zero Code Studio	A visual design environment that enables users to configure, develop and run microcontroller applications in minutes. Zero coding and no IDE required. Access tool here
Application Brief	Accelerate Development of MCU Application Code with MSP Zero Code Studio	An overview of the benefits of designing application code with MSP Zero Code Studio. Access app brief here.
Portfolio Page	Arm Cortex-M0+ MCUs Portfolio	Our comprehensive portfolio of Arm® Cortex®-M0+ MCUs deliver the sensing and processing features you need. Access portfolio page here
Design & Development Page	Arm® Cortex ®-M0+ MCUs design & development	Get started quickly with intuitive and user-friendly hardware, software and development tools. Access design & development page here
Subsystem Tool Page	Arm® Cortex ®-M0+ MCUs subsystems	Solve common MCU design challenges with TI MSPM0 subsystems. Access subsystem page here.
E-book	Analog Engineer’s Circuit Cookbook: MSPM0+ MCUs	The Analog Engineer’s Circuit Cookbook: M0+ MCUs provides subsystem examples that users can quickly adapt to meet their specific system needs. Access cookbook here.

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 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