PSOC™ digital peripherals

Customer training workshop (CTW)

October 2025

Pre-requisites

- Install or update the following on your PC:
  - ModusToolbox™ Tools Package v3.5.0
  - ModusToolbox™ Eclipse IDE 3.5
  - ModusToolbox™ Programming Tools 1.4

Table of contents

Table of contents

PSOC™ project creation

Table of contents

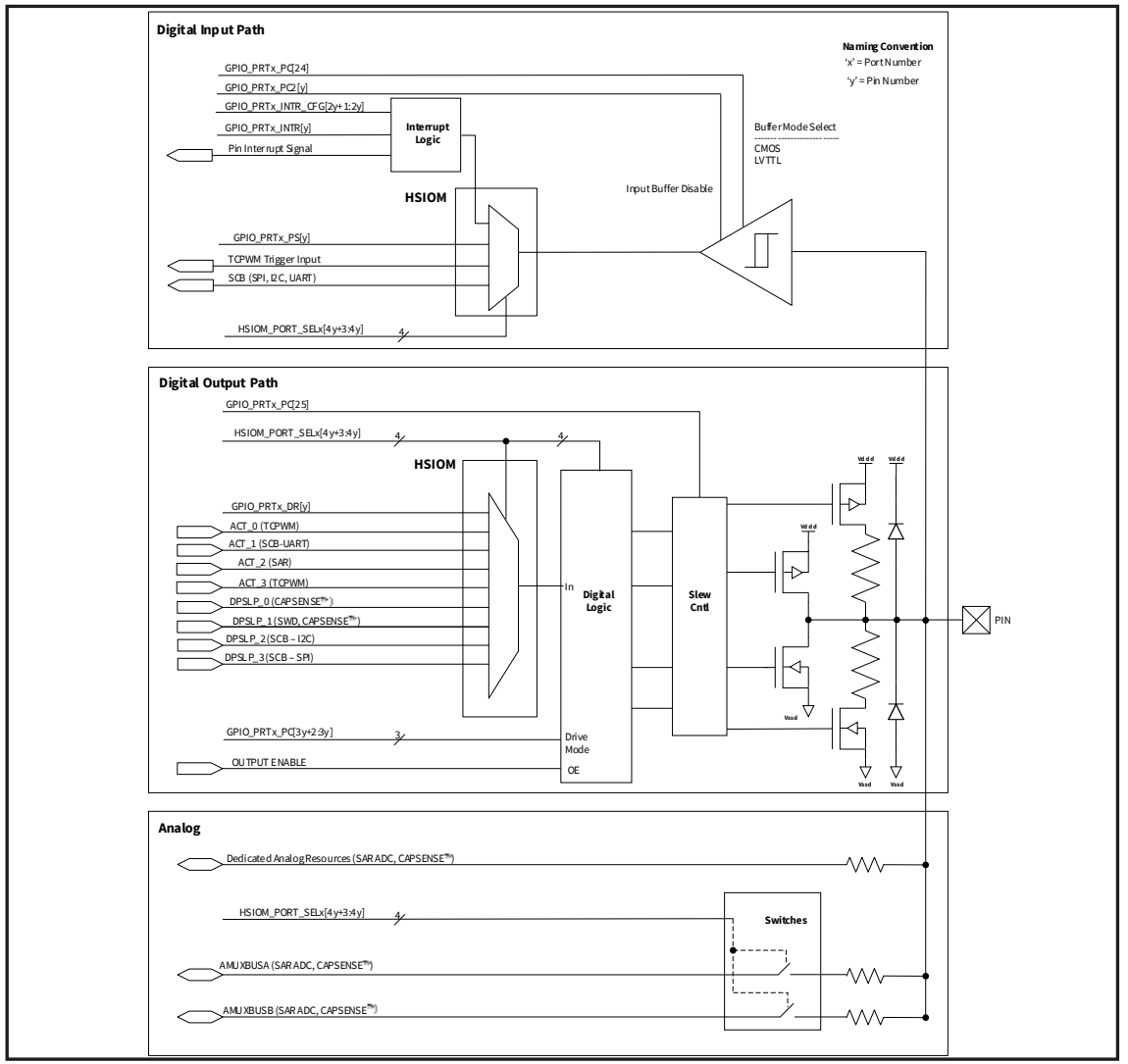
PSOC™ general-purpose I/O (GPIO) control

PSOC™ GPIO control lab agenda

- Configure a PSOC™ GPIO as a strong drive output for LED control using ModusToolbox™ and PDL APIs
- Configure a PSOC™ GPIO as a strong drive output for LED control using only PDL APIs
- Configure a PSOC™ GPIO as an input for reading the status of a user button using ModusToolbox™ and PDL APIs
- Configure a PSOC™ GPIO as an interrupt source using ModusToolbox™ and PDL APIs

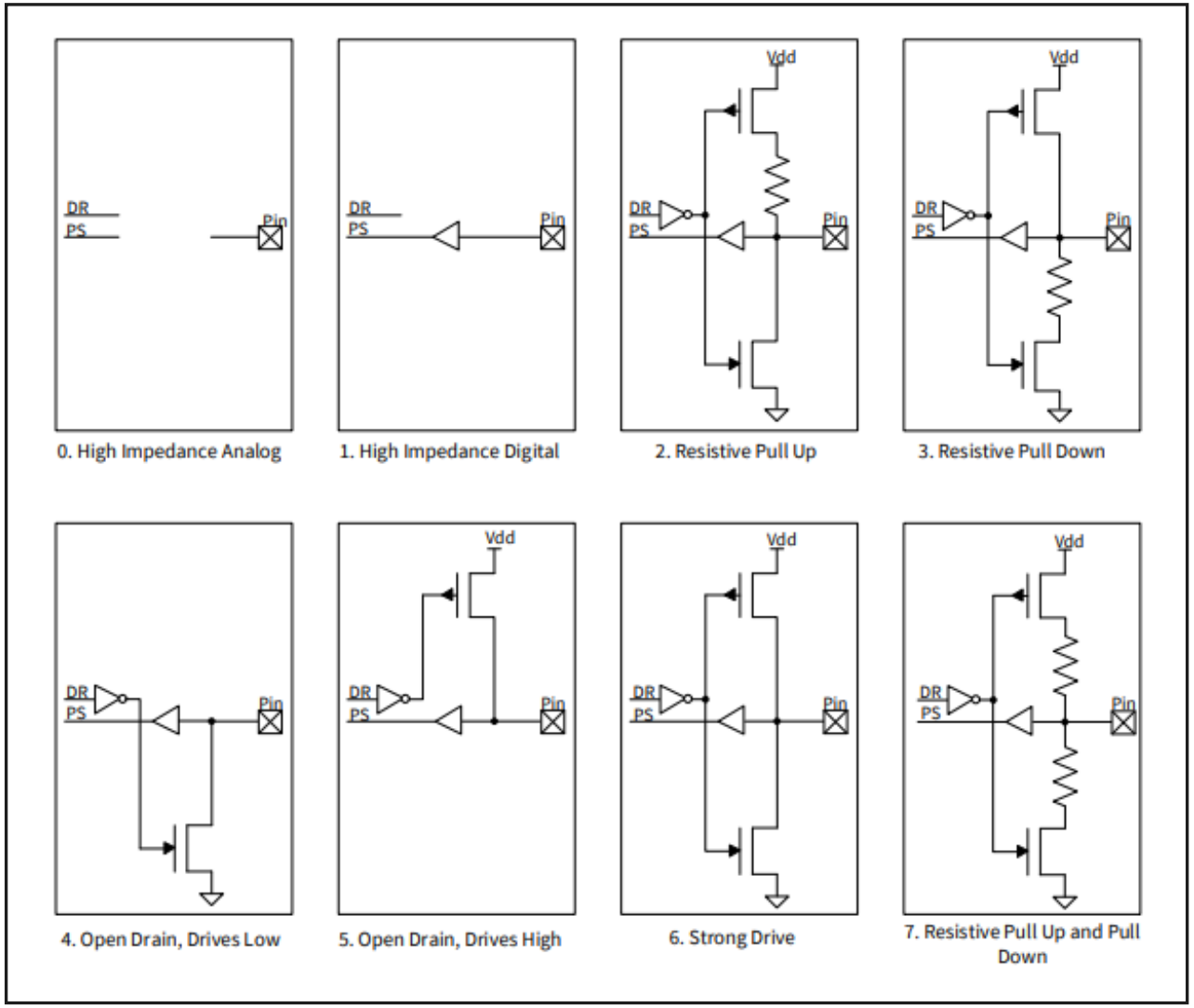
PSOC™ general purpose I/O (GPIO) block diagram

- Digital input path
  - Buffer mode select
  - Input buffer disable
  - High-speed IO matrix (HSIOM)
  - Interrupt logic
- Digital output path
  - Drive mode
  - Slew control
  - Output enable
  - HSIOM
- Analog path
  - Dedicated analog resources
  - HSIOM
  - Analog Mux (AMUX)



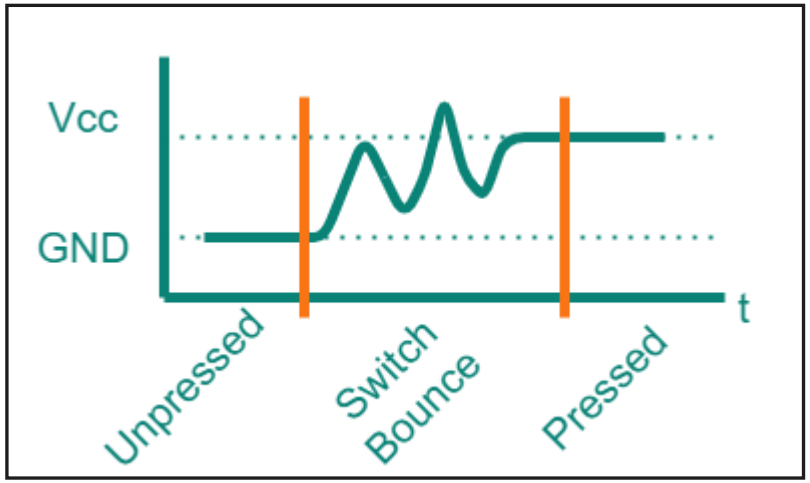
PSOC™ general purpose I/O (GPIO)

- Drive modes
  - High impedance (High-Z) analog
  - High impedance (High-Z) digital
  - Resistive pull-up, pull-down, pull-up, and pull-down
  - Open-drain, drives low or drives high
  - Strong drive
- Input buffer mode
  - CMOS
  - LVTTL
- Slew rate



Considerations when using GPIOs as inputs and interrupts

- Driving circuit
  - **Strong drive**: configure input as a digital high-Z input
  - **Open-drain**: configure input as a resistive pull-up/down based on circuit polarity
  - **TTL compliance**: adjust input thresholds as needed
- Switch bounce
- Interrupt modes
  - Rising
  - Falling
  - Both
- Interrupt priority



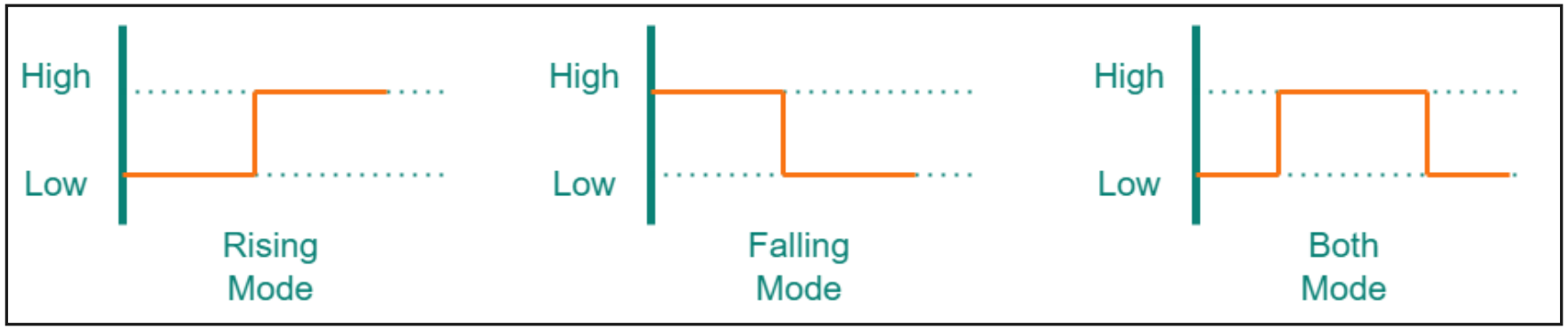
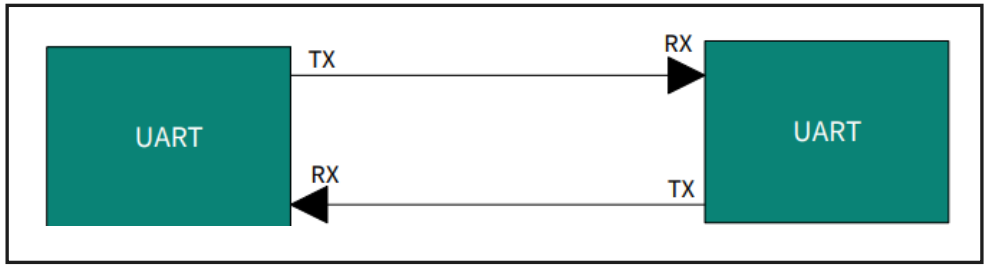


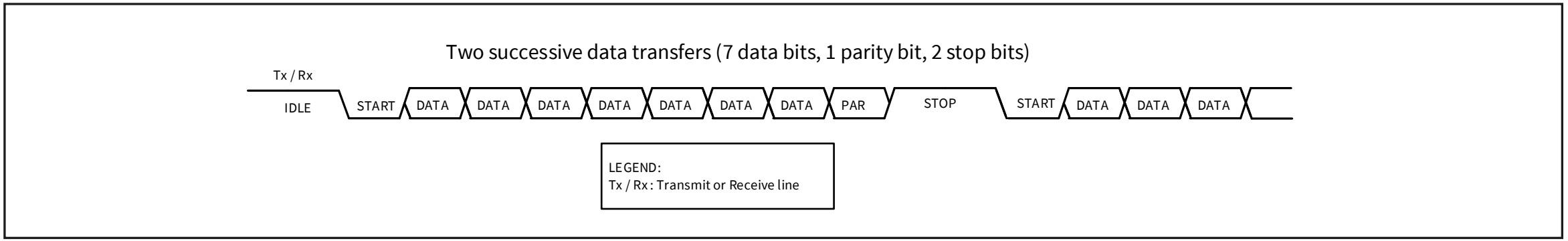
Table of contents

PSOC™ serial communication blocks (SCB) with UART

PSOC™ serial communication block (SCB) with UART

- **Com mode:** Standard, Smartcard reader, IrDA, and LIN
- Configurable baud rate
- **Bit order**: MSB or LSB first
- **Parity**: Odd, even, none
- **Stop bits**: 1 to 3
- Advanced features like RS-485 support and flow control (RTS and CTS)





PSOC™ SCB UART lab agenda

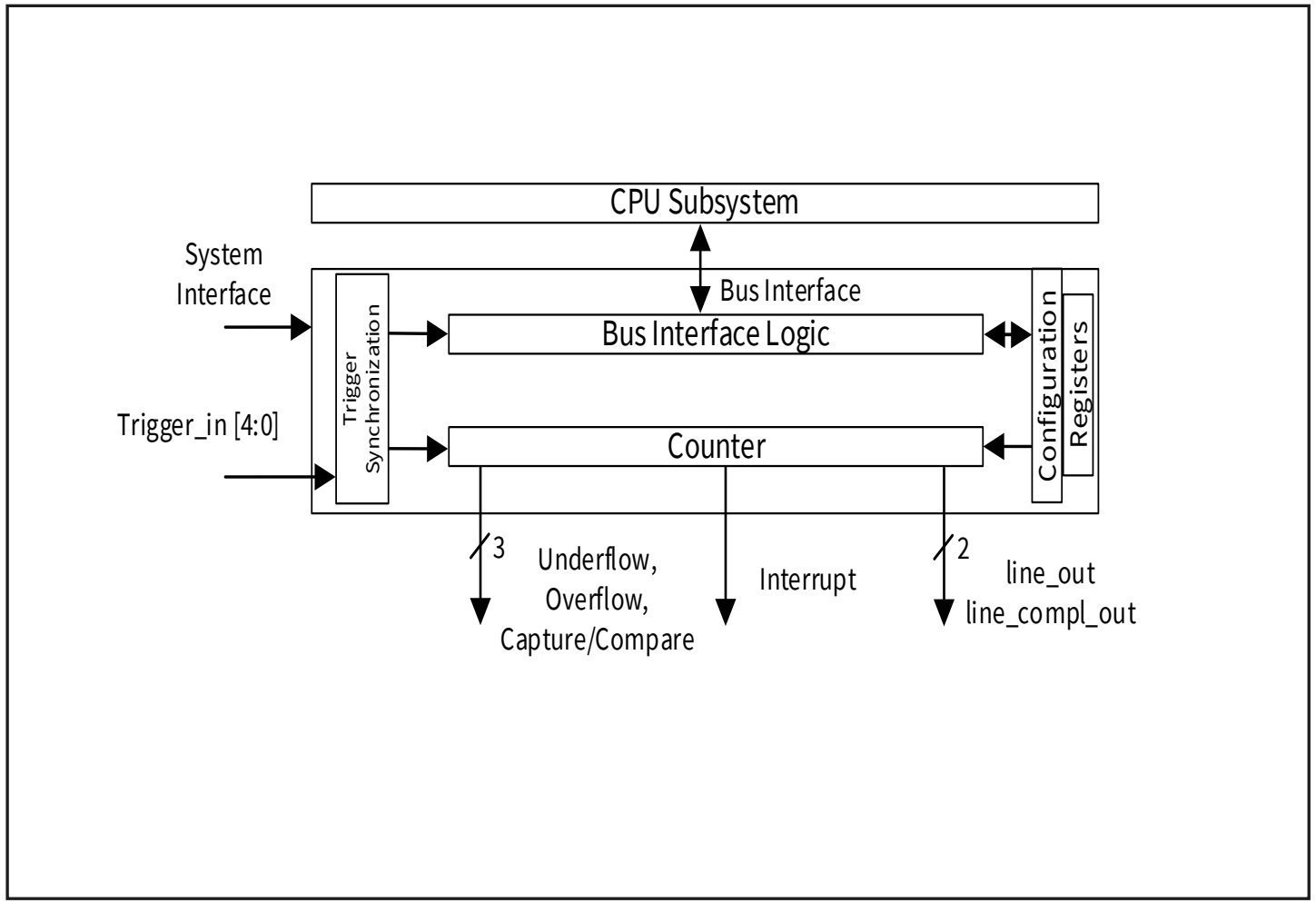
- Configure a PSOC™ SCB for UART mode and send data using ModusToolbox™ and PDL APIs
- Use a PSOC™ SCB in UART mode to receive data with a polled method
- Use a PSOC™ SCB in UART mode to receive data with an interrupt method
- Enable standard C file printf using the retarget-IO library available in the ModusToolbox™ library manager

Table of contents

PSOC™ Timer, Counter, PWM (TCPWM)

PSOC™ Timer, Counter, PWM (TCPWM)

- Operational modes
  - Timer-counter with compare
  - Timer-counter with capture
  - Quadrature decoding
  - Pulse width modulation
  - Pseudo-random PWM
  - PWM with dead time
- Up, down, and up/down counting modes
- Double buffering compare/capture and period values
- Underflow, overflow, and capture/compare output signals
- Interrupts on count and capture/compare

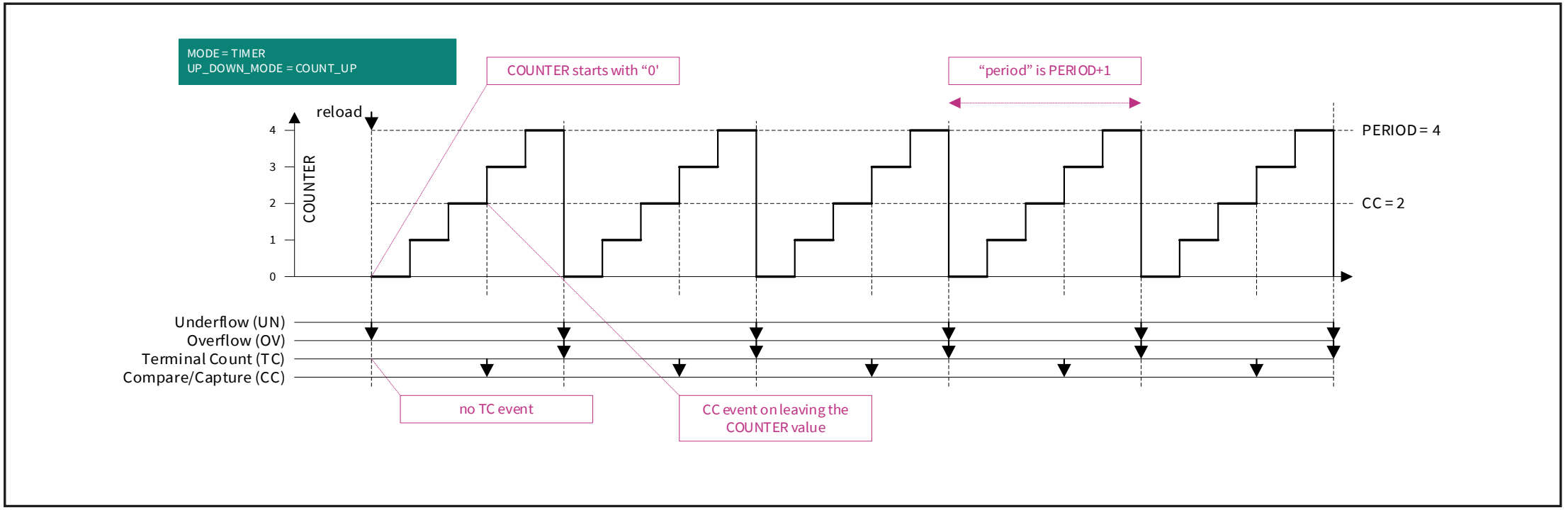


PSOC™ Timer, Counter, PWM (TCPWM) lab agenda

- Configure a PSOC™ TCPWM as a Timer using ModusToolbox™
- Configure a PSOC™ TCPWM for PWM output to control an LED using ModusToolbox™
- Configure a PSOC™ TCPWM as an event counter to keep track of the frequency of a signal using ModusToolbox™
- Configure a PSOC™ TCPWM as an input capture to keep track of the duty cycle of a signal using ModusToolbox™

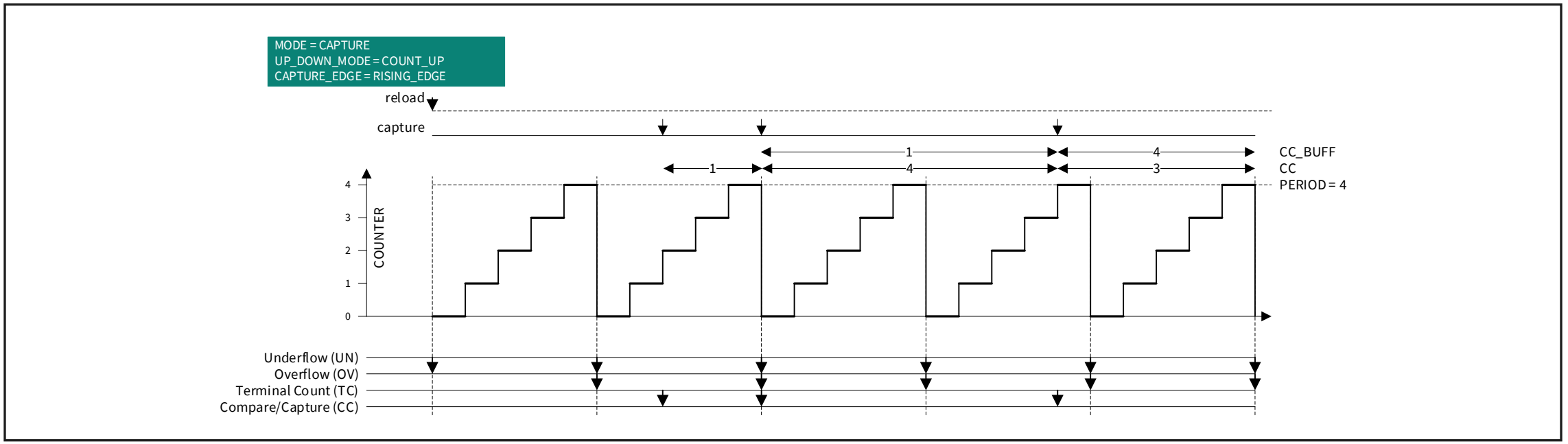
PSOC™ timer counter

- Increments/decrements COUNTER from 0 to PERIOD
- COUNTER is compared with capture/compare (CC)



PSOC™ PWM

- The TCPWM **line_out** toggles high on an overflow event (**OV**)
- The TCPWM **line_out** toggles low when the **COUNTER** matches the **CC** (**cc_match** event)
- The TCPWM **line_compl_out** is the inverse of **line_out**



PSOC™ timer capture

- Increments/decrements COUNTER from 0 to PERIOD
- COUNT is copied to CC on PERIOD rollover; CC is copied to CC_BUFF on new Capture event

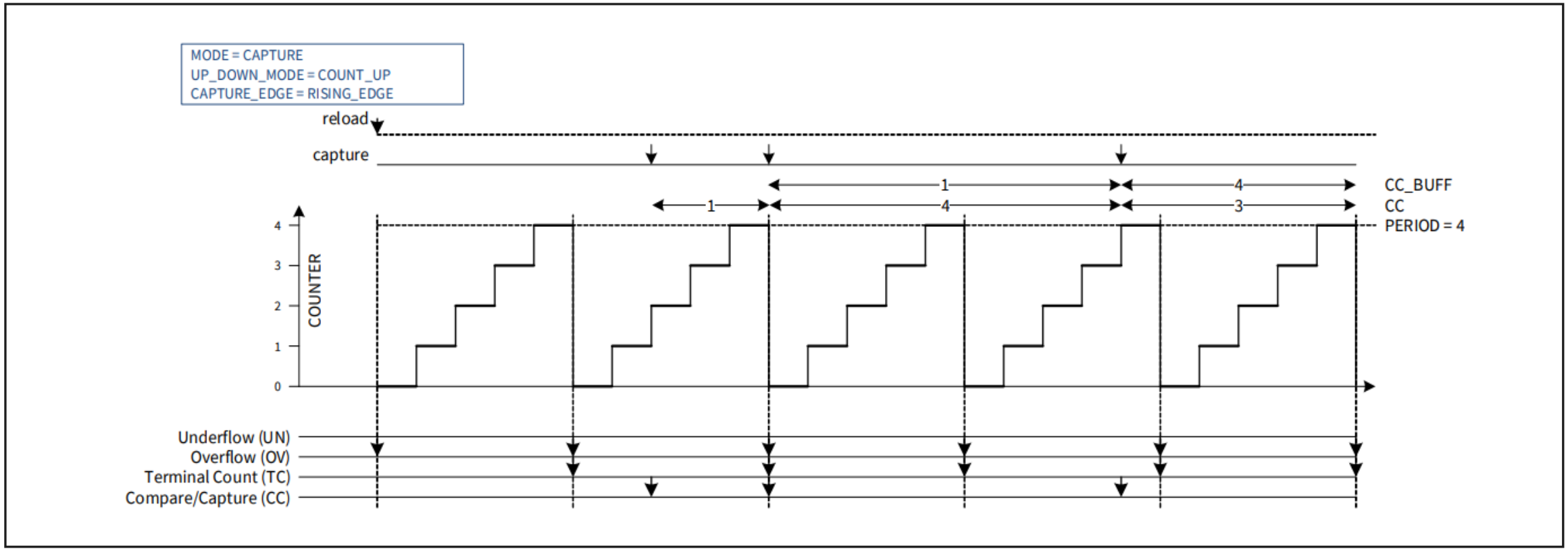


Table of contents

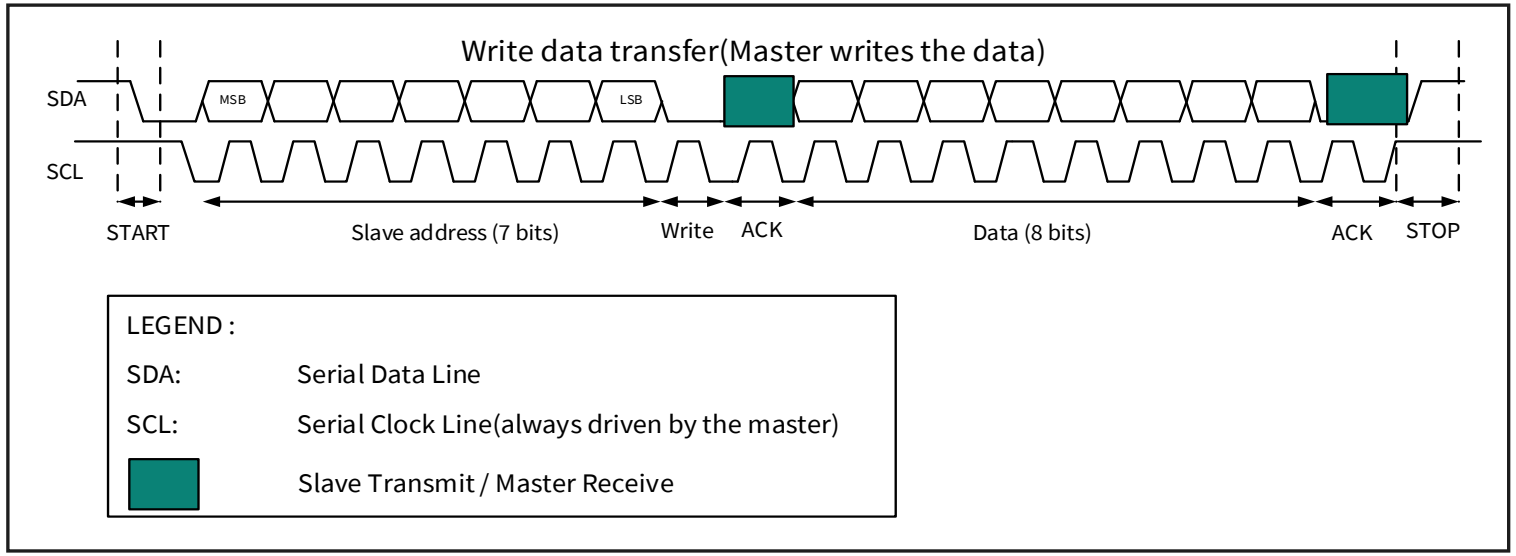
PSOC™ serial communication blocks (SCB) with EZI2C

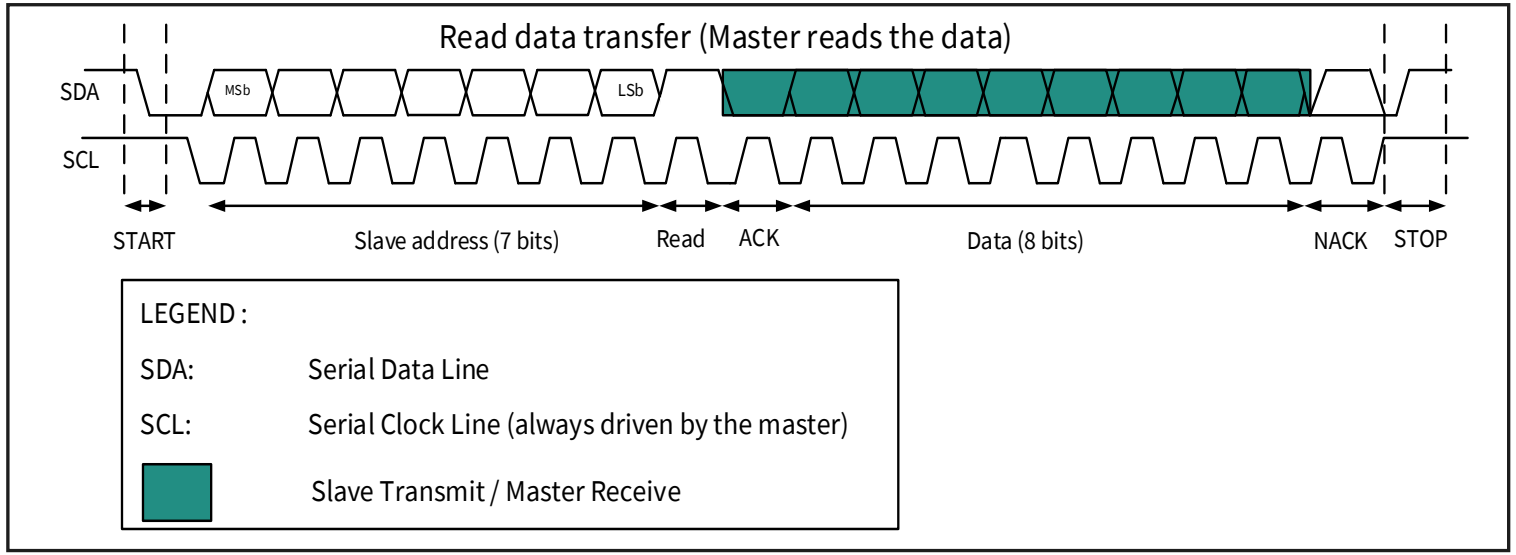
PSOC™ SCB with EZI2C lab agenda

- Configure a PSOC™ SCB for I2C slave mode with the EZI2C driver using ModusToolbox™
  - Respond to an I2C master with static data
- Use a PSOC™ SCB for I2C slave mode to respond to an I2C master with up time data using the EZI2C driver
- Use a PSOC™ SCB for I2C slave mode to receive data from an I2C master to control an LED using the EZI2C driver

PSOC™ SCB with I2C

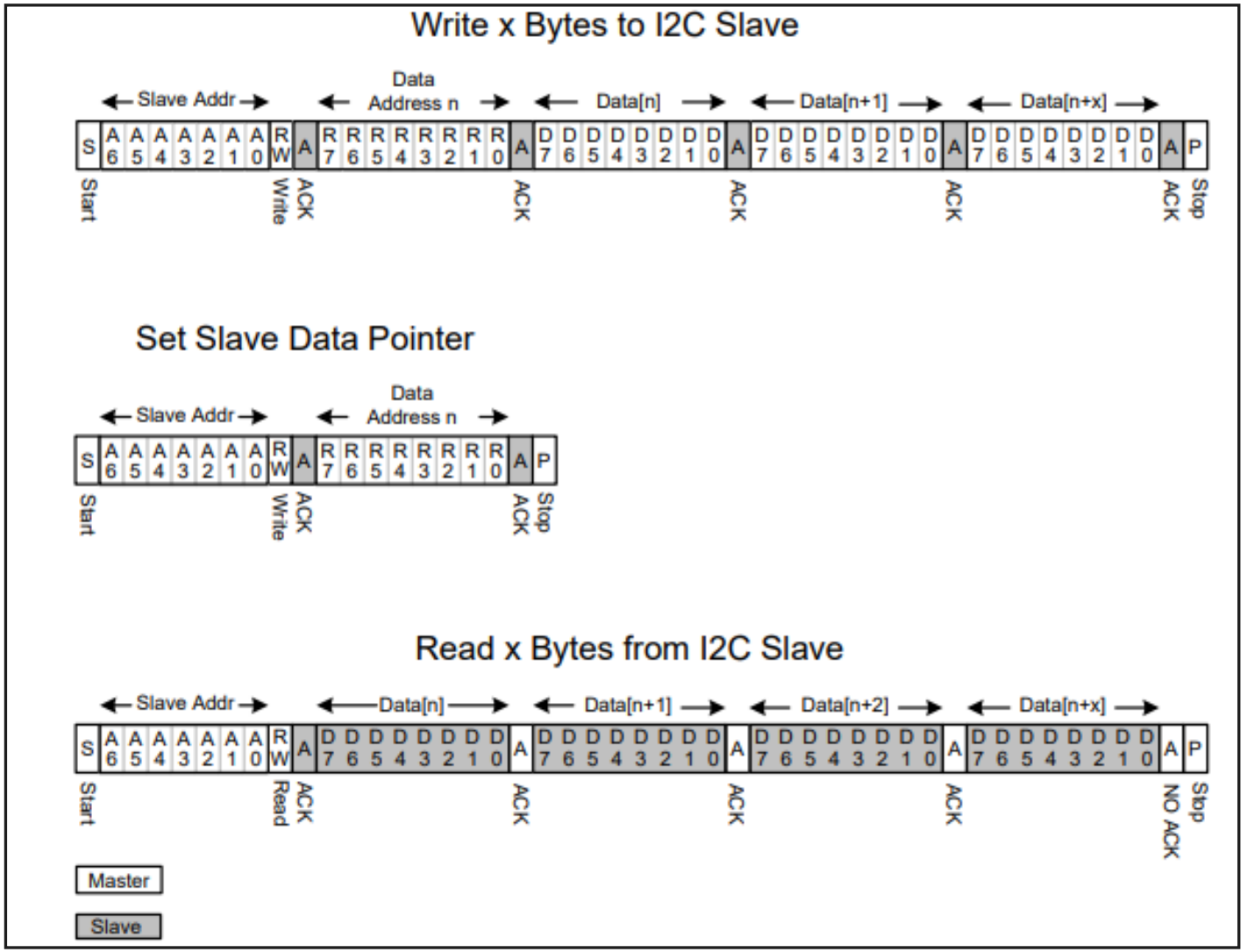
- Configurable I2C slave addresses
- Configurable baud rate
- Supports RX and TX interrupts
- Multi-Master bus arbitration is supported





PSOC™ with and the EZI2C software driver

- EZI2C enables a common EEPROM interface with 8- or 16-bit sub-address sizing
- Allows random access to a block of memory on the EZI2C slave



PSOC™ SCB configured for serial peripheral interface (SPI)

PSOC™ SCB with serial peripheral interface (SPI) lab agenda

- Configure a PSOC™ SCB for SPI master mode using ModusToolbox™
  - Using a second SCB for UART to print sent and received data

- Configure a PSOC™ SCB for SPI slave mode using ModusToolbox™
  - Using a second SCB for UART to print sent and received data

PSOC™ SCB with serial peripheral interface (SPI) and the high-level driver

- High configurability
  - Baud rate
  - Master or slave mode
  - Clock Polarity (CPOL) and Clock Phase (CPHA)
  - Multiple slave select lines (SS)
- Multiple SPI modes supported:
  - Motorola
  - Texas Instruments
  - National Semiconductor (half-duplex)

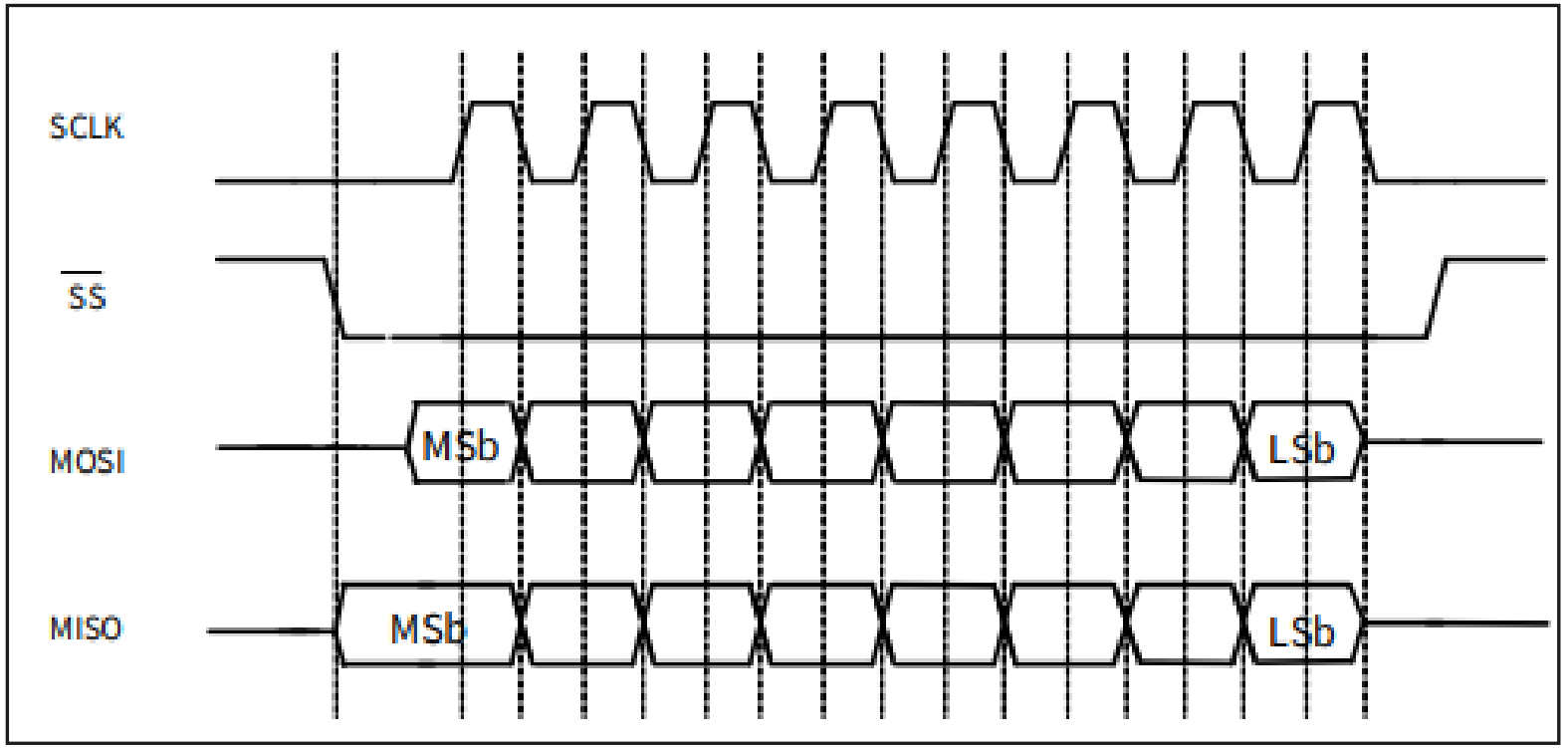


Table of contents

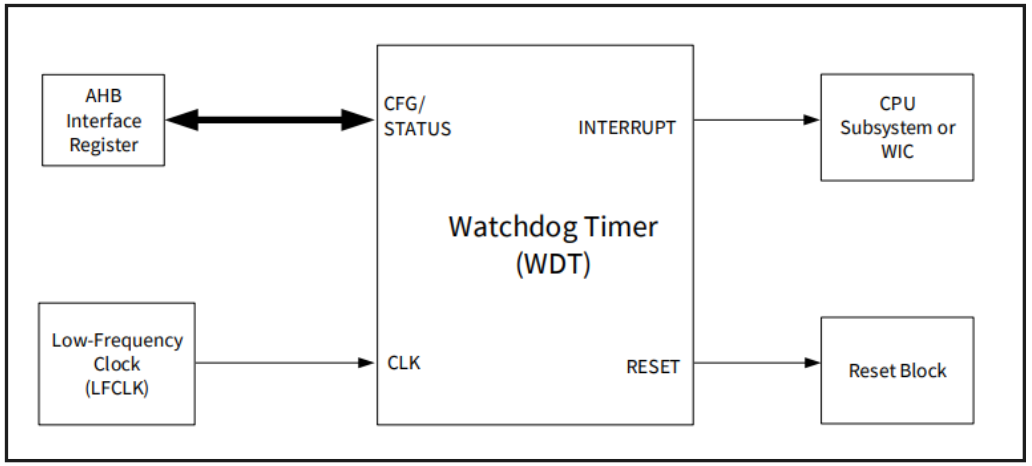
PSOC™ watchdog timer (WDT)

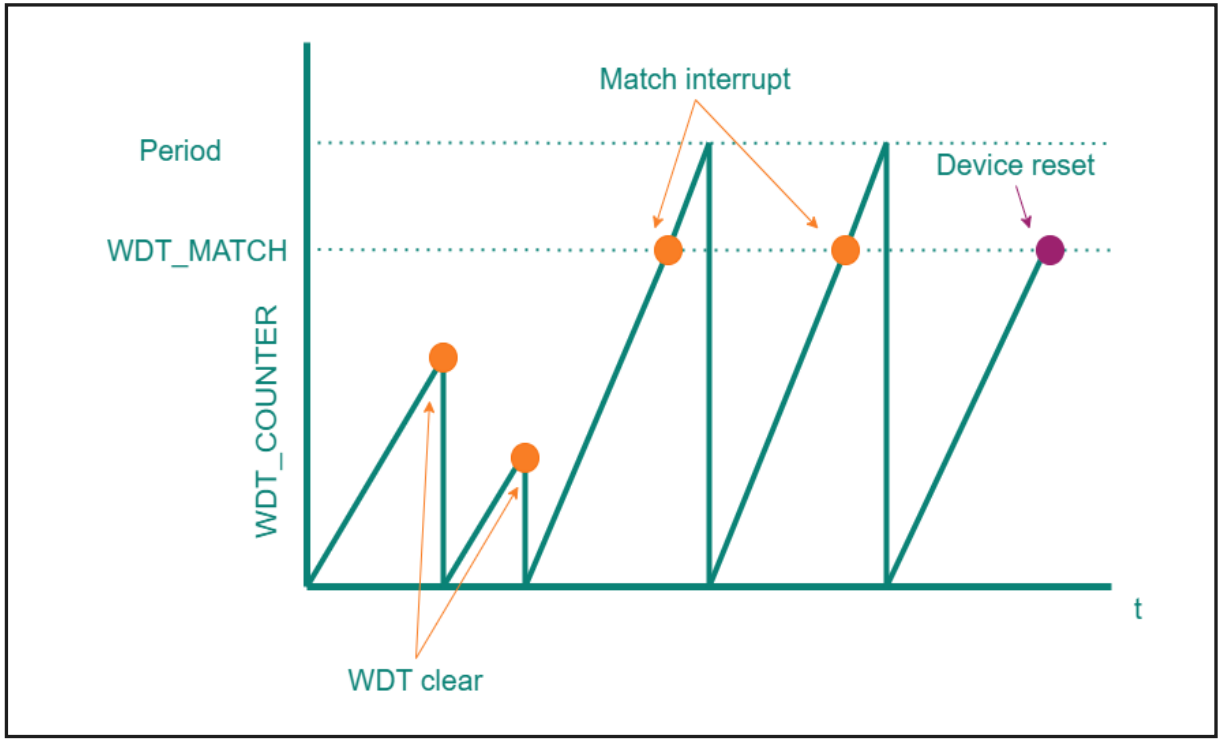
PSOC™ watchdog timer (WDT) lab agenda

- Configure a PSOC™ watchdog timer (WDT) to reset the MCU when the timer is not cleared
  - A button press will trigger an infinite while loop, mimicking a software fault

PSOC™ watchdog timer (WDT)

- Supports interrupt and system reset on timer expiration
- Sourced from LFCLK to support operation in hibernate mode
- The WDT asserts a system reset to the device on the third WDT match event, unless it is periodically serviced in firmware





Revision history

Document revision	Date	Description of change
**	2025-10-31	New spec.

Thank you!
Stay tuned for more trainings…

Join with us on our mission to drive decarbonization and digitalization Together.


Infineon