The Debug Mindset
TI Precision Labs – Microcontrollers

Presented by Amruta Deole
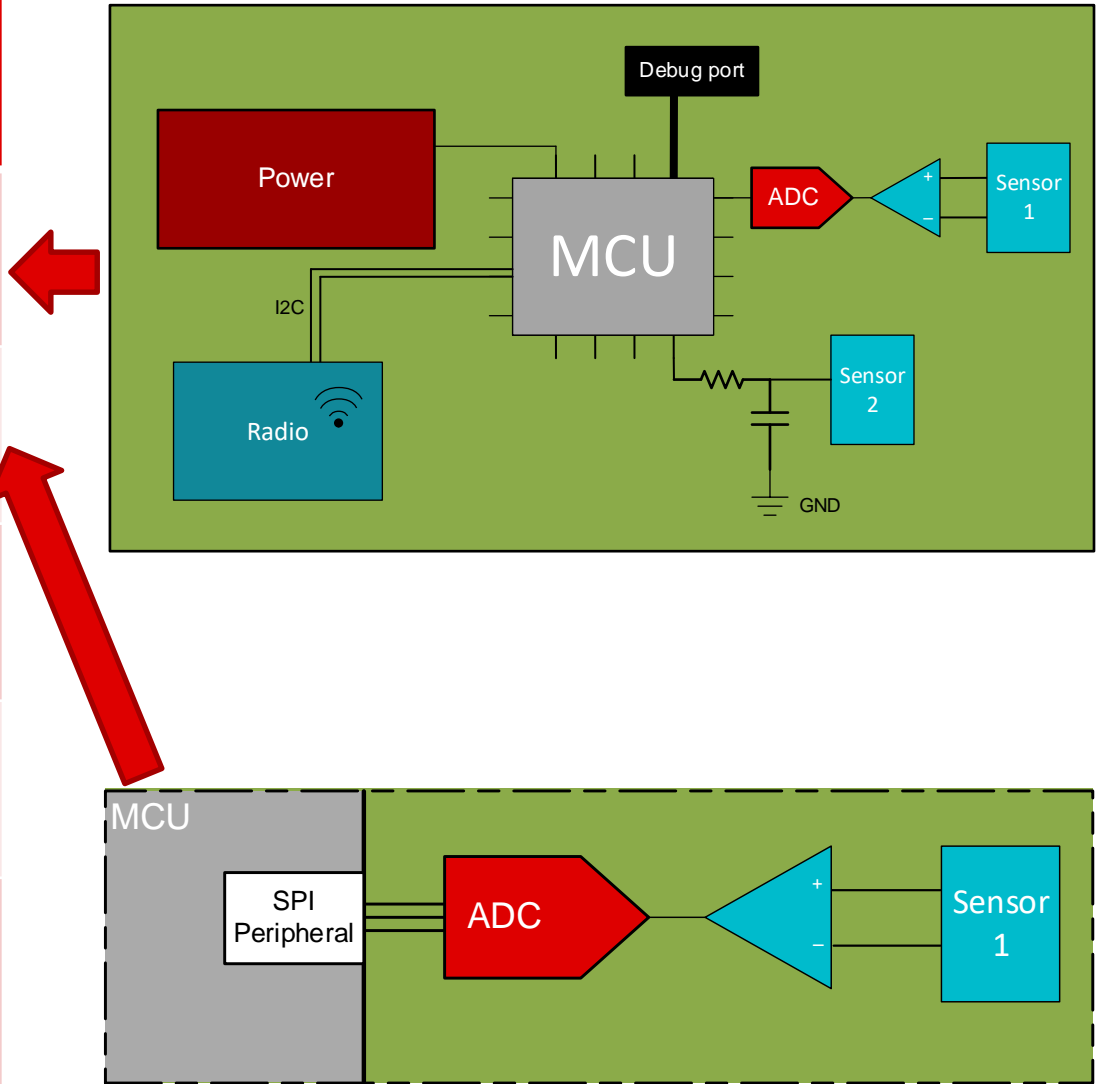
Prepared by Jace Hall

TEXAS INSTRUMENTS

Mindsets needed to debug problems

Break problem into smaller parts	Known good states
Test and iterate	Check assumptions

Talk it out

Check assumptions
1. ~~Power?~~ ✓
2. ~~Sensor 1 ?~~ ✓
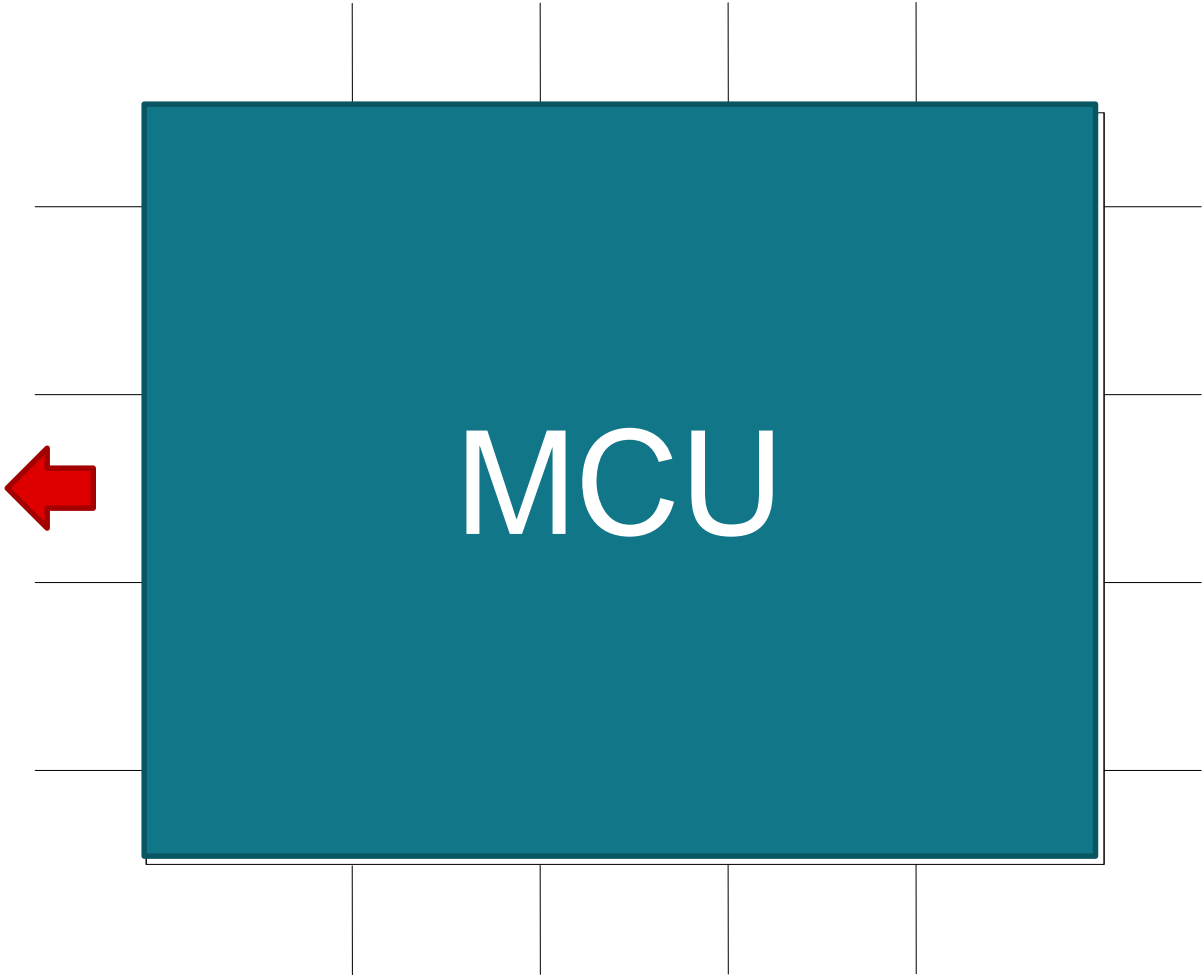3. ~~Register settings ?~~ ✓
4. Firmware

Terminology definitions

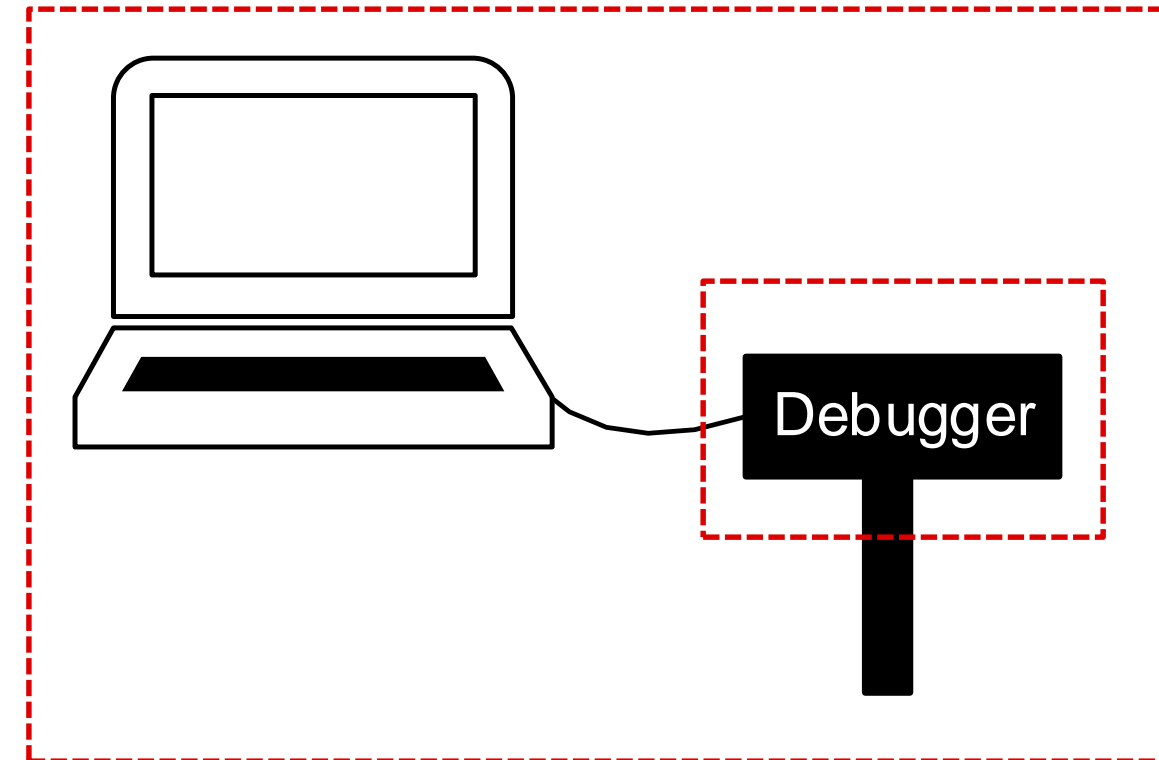| Common Term	Definition
Embedded System	This is the culmination of your microcontroller, external sub-systems, PCB board, and software needed for your application
Sub-system	A collection of circuits and software that perform a specific functionality for the overall embedded system
Microcontroller / MCU	This is typically the heart of your system or sub-system and where hardware and software meet to do an application
Peripheral	An MCU internal component that does specific functions, such as I2C communication or an RTC
Debug Chain	The hardware and software components necessary for debugging an embedded system on a PC. This includes JTAG connection, debugger, and your code IDE such as CCS
Debugger / Programmer	A piece of hardware that connects from a PC to our embedded system that can both program your MCU and read internal registers for the purposes of debugging

Terminology definitions

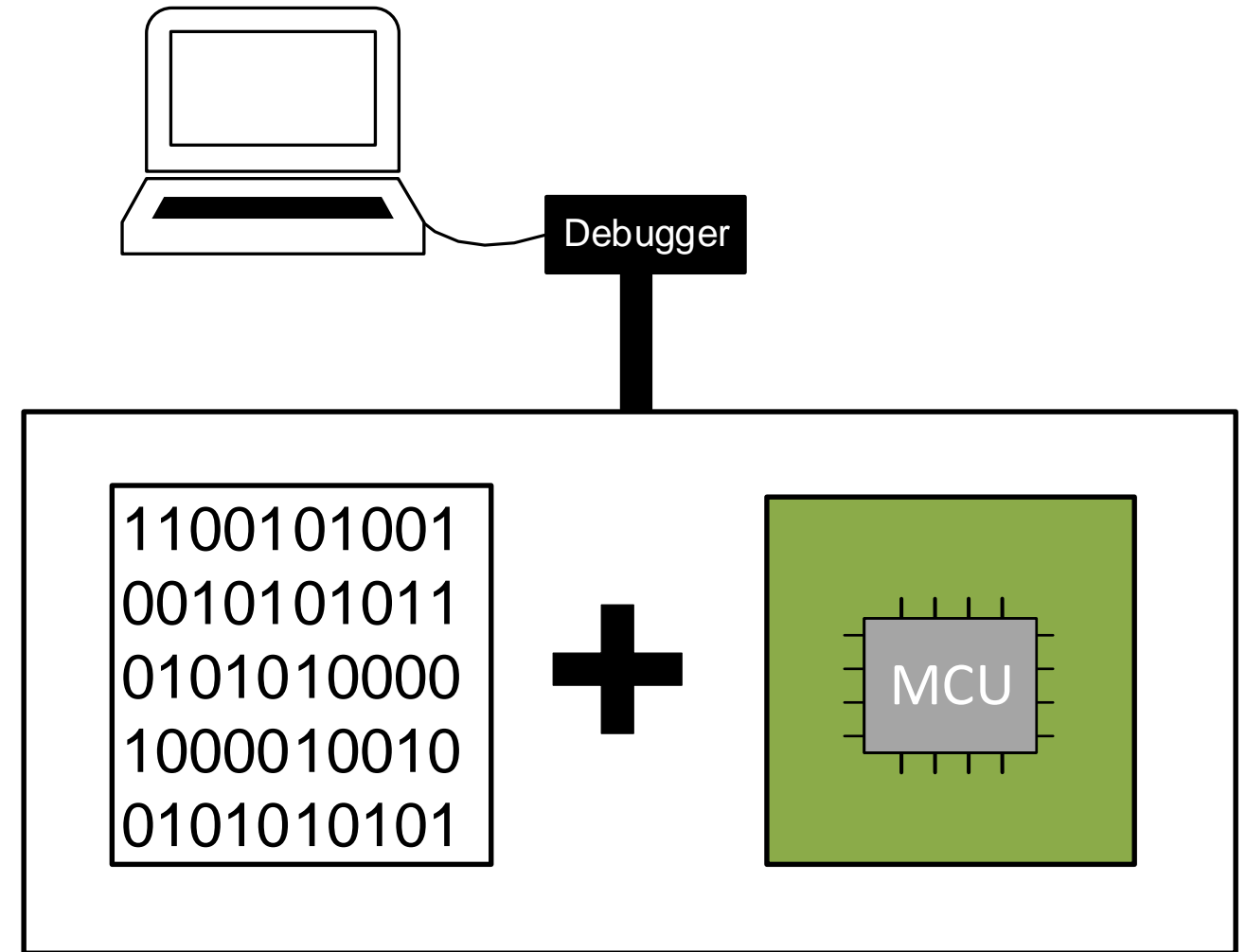| Common Term	Definition
Embedded System	This is the culmination of your microcontroller, external sub-systems, PCB board, and software needed for your application
Sub-system	A collection of circuits and software that perform a specific functionality for the overall embedded system
Microcontroller / MCU	This is typically the heart of your system or sub-system and where hardware and software meet to do an application
Peripheral	An MCU internal component that does specific functions, such as I2C communication or an RTC
Debug Chain	The hardware and software components necessary for debugging an embedded system on a PC. This includes JTAG connection, debugger, and your code IDE such as CCS
Debugger / Programmer	A piece of hardware that connects from a PC to our embedded system that can both program your MCU and read internal registers for the purposes of debugging

MCU

Terminology definitions

| Common Term	Definition
Embedded System	This is the culmination of your microcontroller, external sub-systems, PCB board, and software needed for your application
Sub-system	A collection of circuits and software that perform a specific functionality for the overall embedded system
Microcontroller / MCU	This is typically the heart of your system or sub-system and where hardware and software meet to do an application
Peripheral	An MCU internal component that does specific functions, such as I2C communication or an RTC
Debug Chain	The hardware and software components necessary for debugging an embedded system on a PC. This includes JTAG connection, debugger, and your code IDE such as CCS
Debugger / Programmer	A piece of hardware that connects from a PC to our embedded system that can both program your MCU and read internal registers for the purposes of debugging

Debugger

TEXAS INSTRUMENTS

Break up the problem

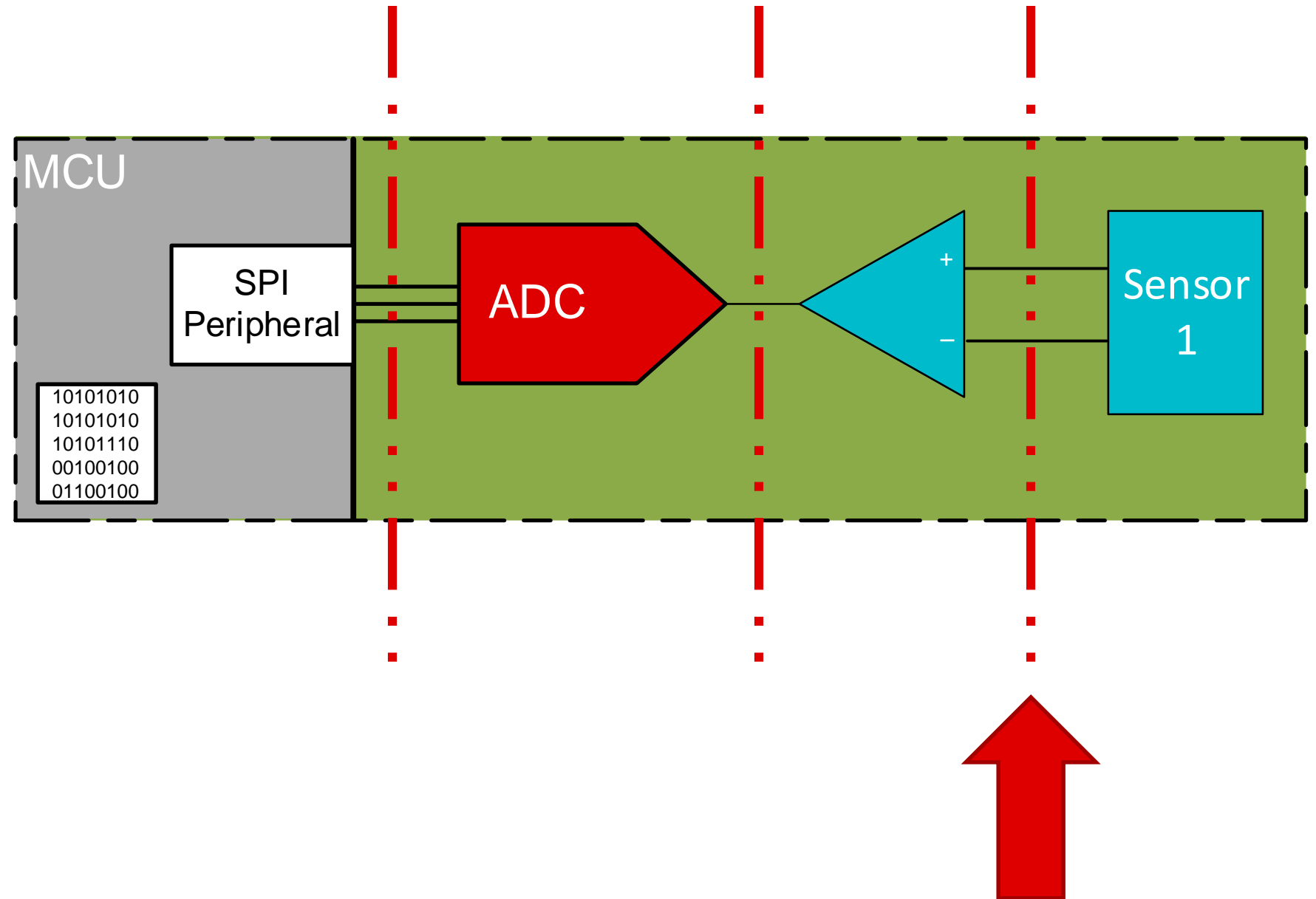- Software and Hardware components

Break up the problem

- Software and Hardware components

- Sub-system division
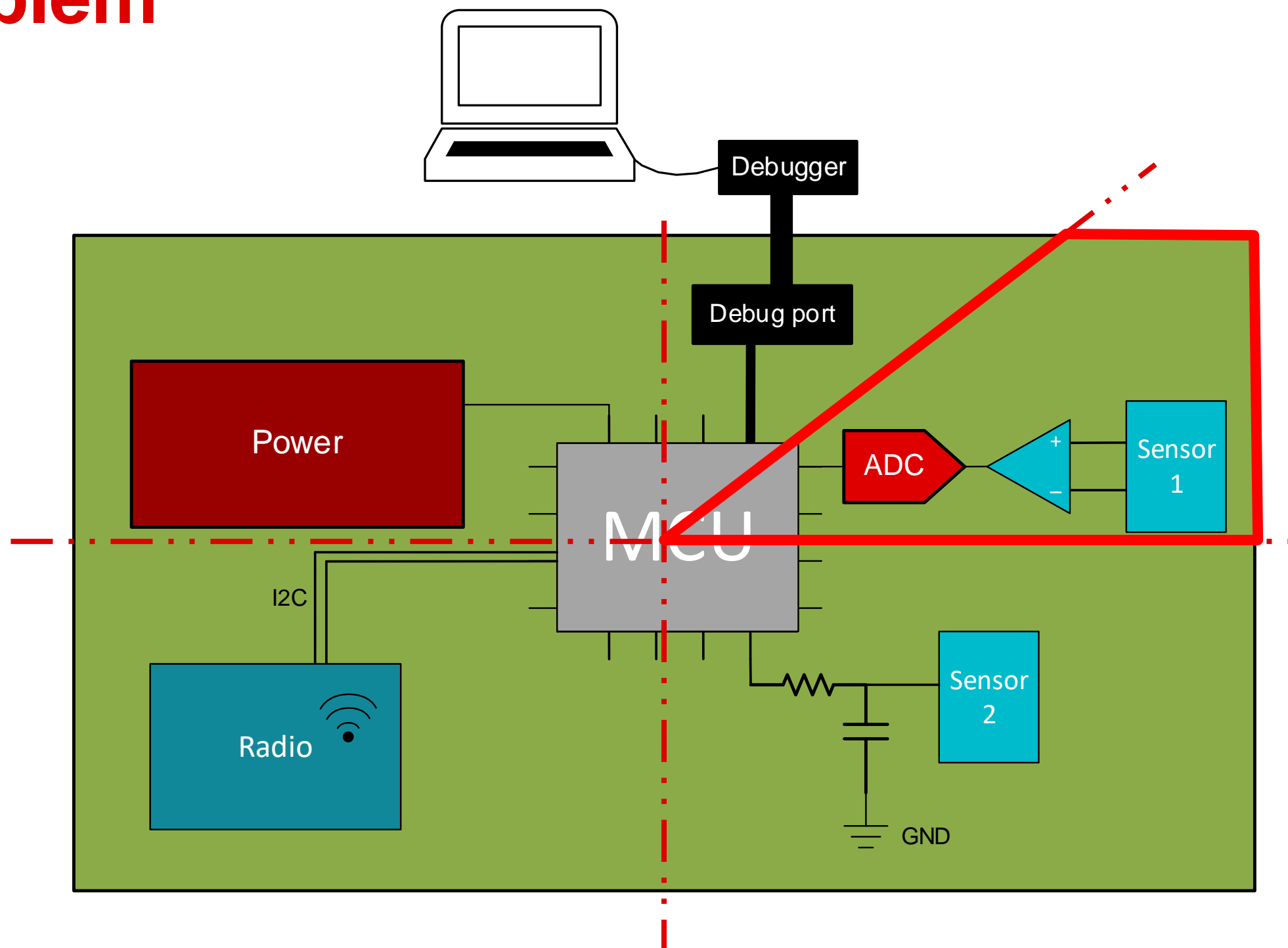
Break up the problem

- Software and Hardware components

- Sub-system division

- Further division

Known good states

- ## Software:
  - Disable added features to confirm a working state
  - Disable other sub-systems' code
  - Start from MCU code examples, modify to fit your application

- ## Hardware:
  - Use simple code examples to test external hardware
  - If HW is modified, use a known good board to check your debug efforts
  - If a HW component is suspect, replace with a known good replacement
  - Remove HW features of board to isolate debug efforts
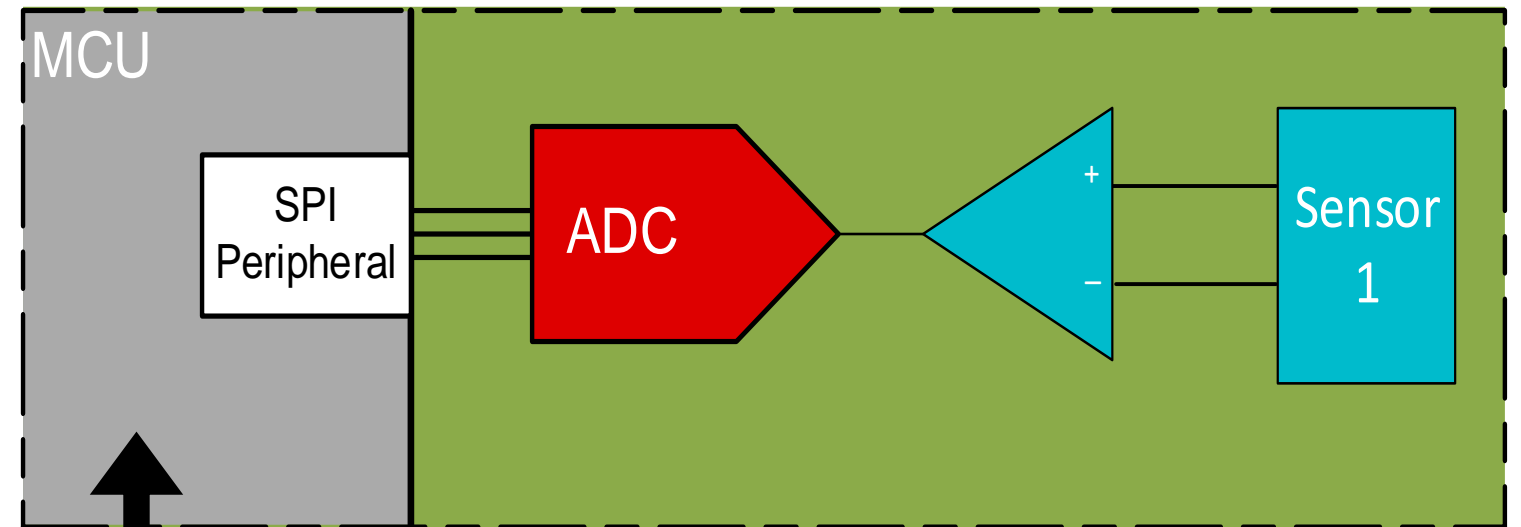

Code:
- Initialize MCU and Clocks
- Initialize SPI
- Configure external ADC over SPI
- Take data from ADC over SPI
- Process data
- Take action on data
- Wait for next transaction and loop

Test and iterate

Test
…small changes

Learn
…resulting behavior

Iterate
… to establish known good states
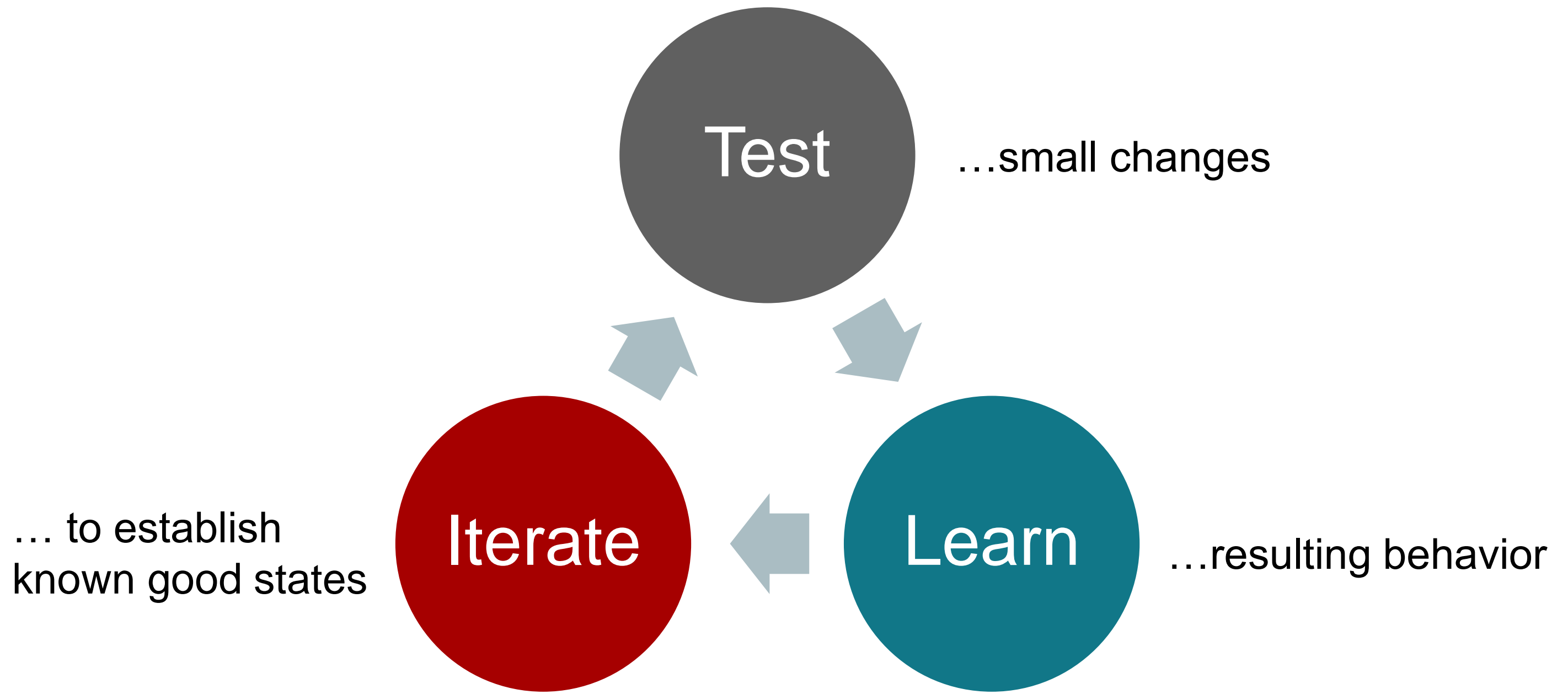
Check assumptions

- Issue scope
- Expected behaviors
- Untested "known goods"

- Power
- Debugger interference
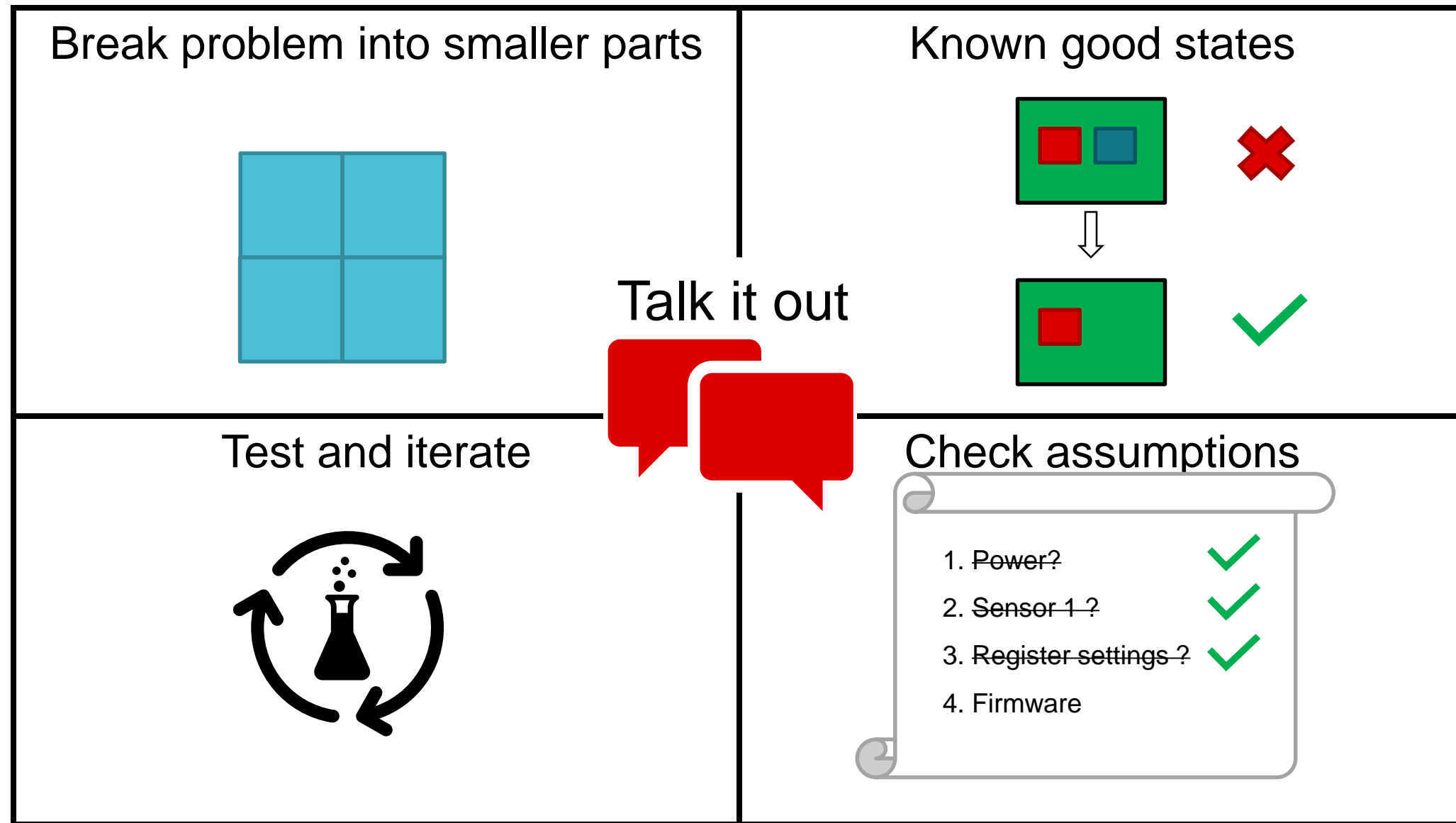- Adjacent sub-system interference
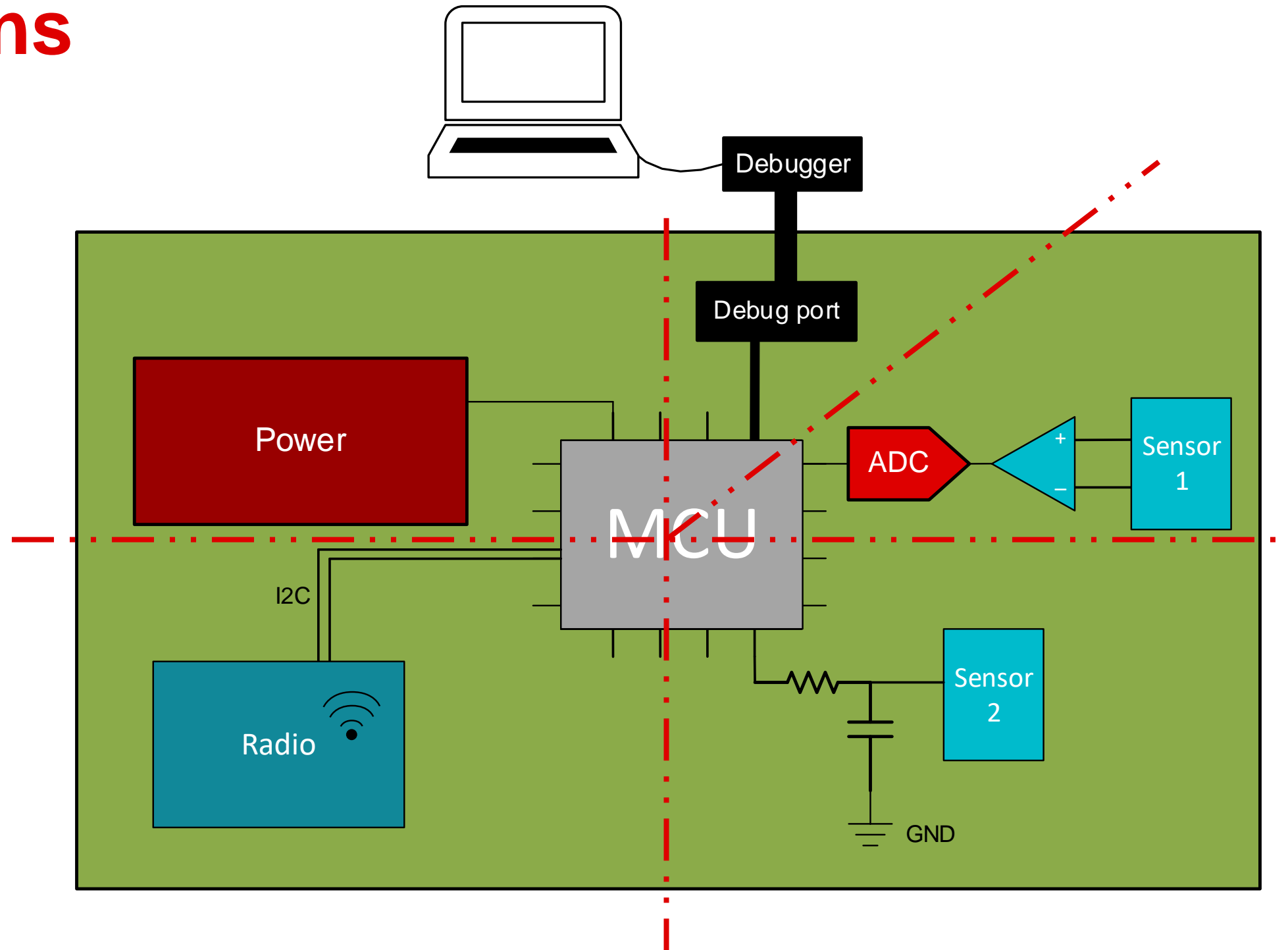
Talk about your problem

- Seek help

- Get a different perspective

- The Rubber Duck Method

- Explain or teach your problem to gain new insight
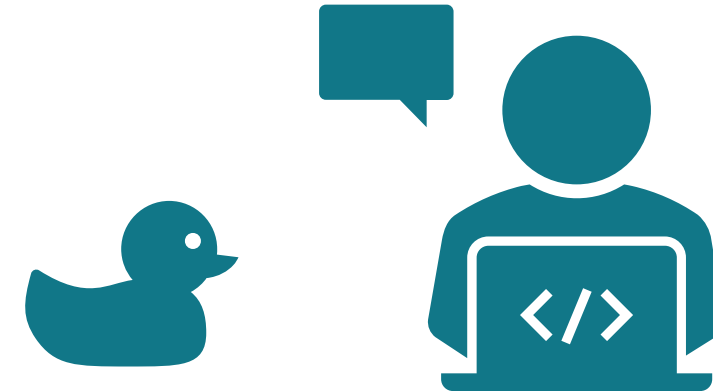
Recap

To find more <insert product name> technical resources and search products, visit ti.com/<insert product portal>