CEC Continuing Education Center

DesignNews

Best Practices for Designing Real-Time Embedded Systems

DAY 3 : It’s All About the Data

Sponsored by

Digi-Key ELECTRONICS

informa markets

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Course Sessions

- System Level Design Philosophy
- Designing a Hardware-less System
- It's All About the Data
- Testing Your Way to Design Success
- The Best Practices Lightning Round

1

Following the Data

At the end of the day, an embedded system is just managing dataflow.

Following the Data

5 Types of Data Mechanisms to Consider

- Producers
- Consumers
  - Processors
  - Storage
- Transfer Mechanisms

Following the Data

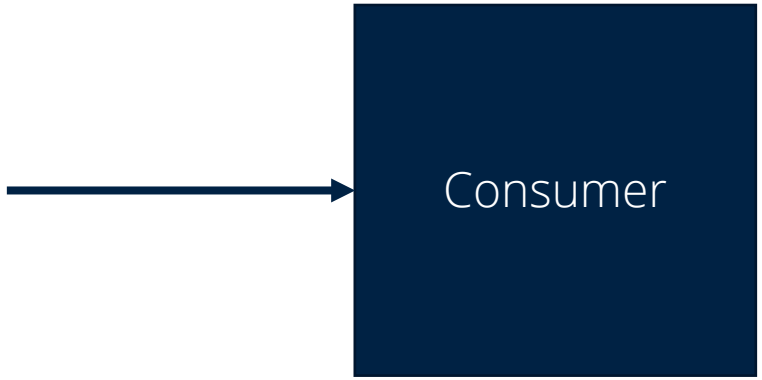
Producers Generate the Data:

- Input blocks to the system
- Analog sensors (ADC's)
- Digital Sensors (SPI, I2C, etc)
- Communication devices (Wi-Fi, USB, etc)
- Test Modules
- Application modules

Following the Data

Consumers use the Data:

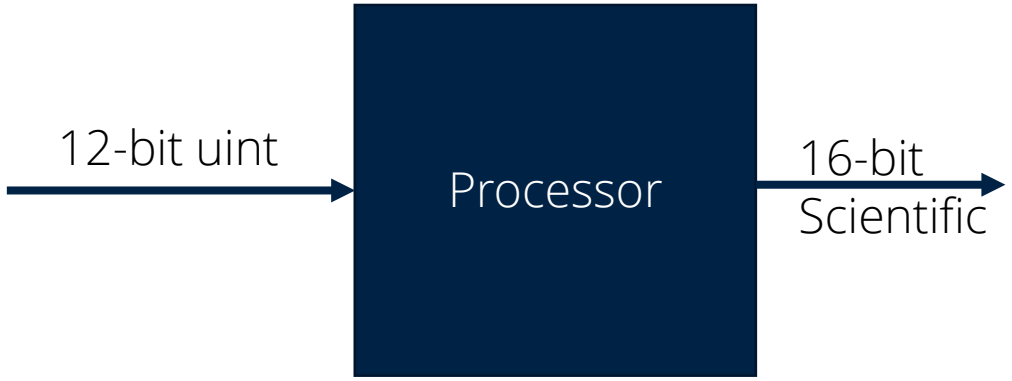
- Output Blocks from the system
- Displays
- Communication devices (Wi-Fi, USB, etc)
- Test Modules
- Application modules



Following the Data

Processors Convert the Data:

- Take in data and may produce new data
- Communication devices (Wi-Fi, USB, etc)
  - Packet data into structured data
- Test Modules
- Application modules
  - 12-bit Adc to scientific scaled units



Following the Data

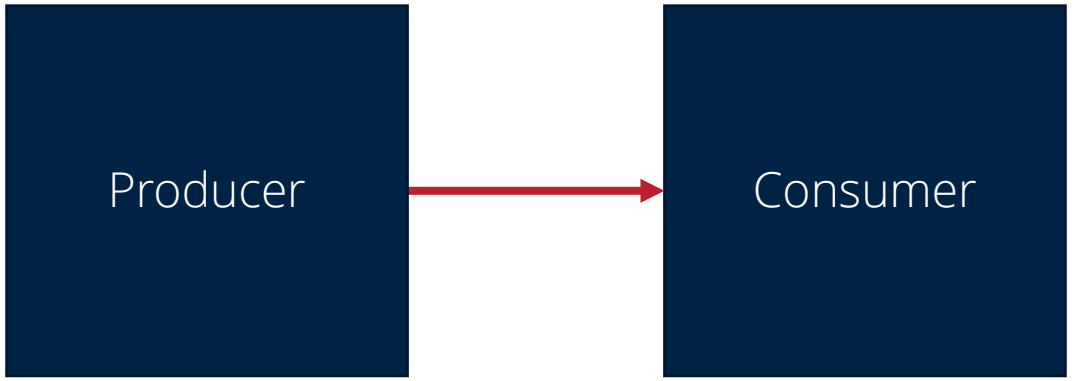
Storage manages data:

- RAM (Volatile)
- Flash (Non-volatile)
- Internal / External
- Encrypted vs non-encrypted

Following the Data

Transfer Mechanisms Move the Data:

- "Middle" modules
- CPU Registers
- Direct Memory Access (DMA)
- Queues
- Storage



Do you currently architect your application based on the data?

- Yes
- No
- No, but now I plan to

2

Decomposing an Application

Application decomposition consists of identifying inputs, outputs and transfer mechanisms.

Decomposing an Application

Step #1 – Identify the final data products (Outputs)

- Display data
- PWM frequency and duty cycle
- GPIO states
- Communication packet
- Log files
- etc

Decomposing an Application

Step #2 – Analyze the final data products (Outputs) requirements

- Units and scaling
- Sample rates
- Data size
- Response times
- (Security)

Decomposing an Application

Step #3 – Identify how data is getting to the output (Transfers)

- Transfer on LVDS
- Serial interface
- Peripheral Register
- DMA
- Interrupt Service Routine (ISR)

Decomposing an Application

Step #4 – Identify Processing Mechanisms

- Application modules
- Data converters
- Analyzers
- Inferences
- DSP algorithms

Decomposing an Application

Step #5 – Identify Inputs

- Application modules
- Peripheral devices
- Communication interfaces
- Data generators
- etc

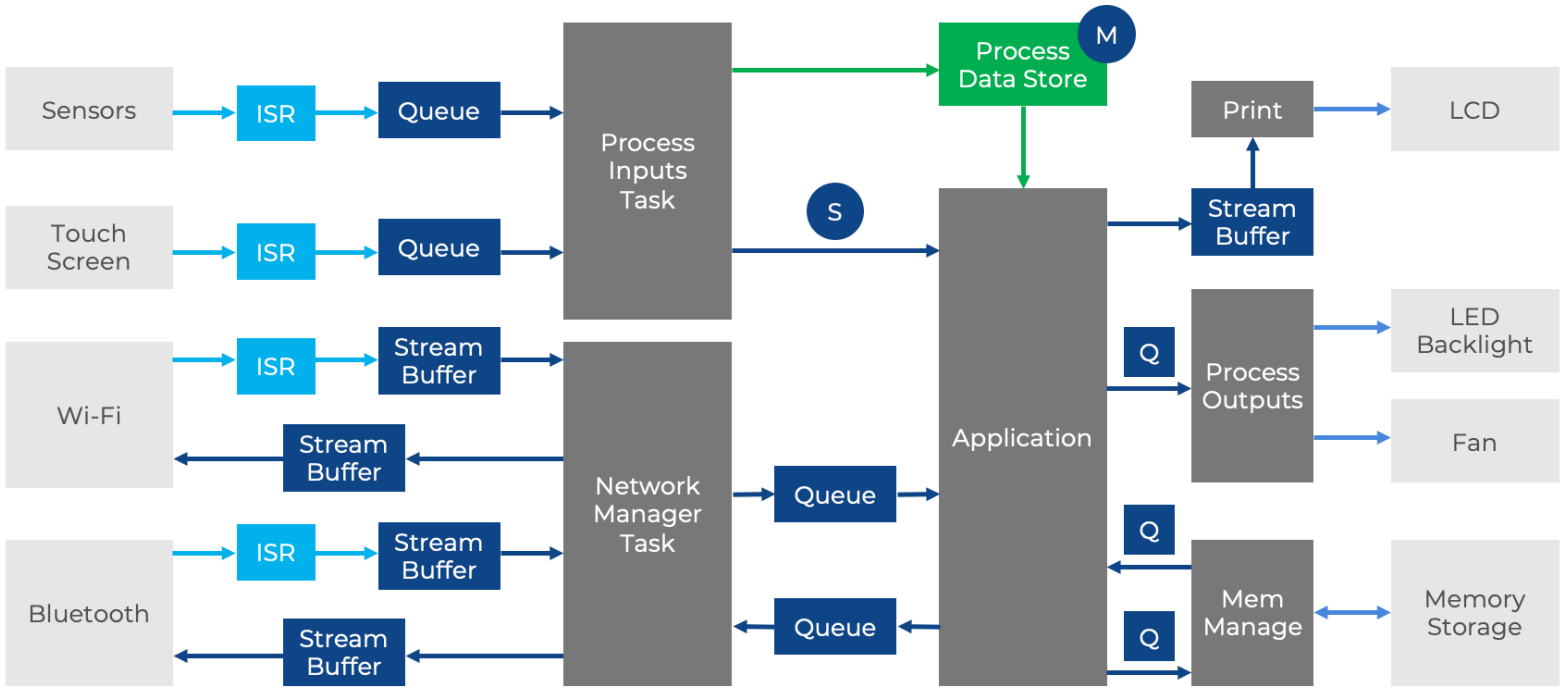
Decomposing an Application

Step #6 – Analyze the input data products requirements

- Units and scaling
- Sample rates
- Data size
- Response times
- (Security)

Decomposing an Application

Step #7 – Develop a Dataflow Diagram



What are the advantages to developing a data flow diagram?

- All data are identified up front
- Can analyze transfer mechanisms and system through-put
- Clear high-level map for software developers to follow
- All the above
- None of the above
- Other

Thank you for attending

Please consider the resources below:

- www.beningo.com
  - Blog, White Papers, Courses
  - Embedded Bytes Newsletter
    - http://bit.ly/1BAHYXm

From www.beningo.com under

- Blog > CEC – Best Practices for Real-Time Embedded Systems

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key ELECTRONICS

informa markets