CEC Continuing Education Center

DesignNews

Embedded Software Design Techniques

DAY 2 : Designing RTOS-based Applications

Sponsored by

Digi-Key ELECTRONICS

BENINGO EMBEDDED GROUP

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 ‘Group Chat’ by maximizing the chat widget in your dock.
- Submit questions for the lecturer using the Q&A widget. They will follow-up after the lecture portion concludes.

Course Sessions

- Software Architectures 101
- **Designing RTOS-based Applications**
- Architecture Verification Techniques
- Designing Quality into Embedded Systems
- Software Configuration Management Techniques

1

Task Decomposition

How do I break my application up?

Task Decomposition – The Outside-In Approach

1. Identify the major components
2. Draw a high-level block diagram
3. Label the inputs
4. Label the outputs
5. Identify the first-tier tasks
6. Determine concurrency levels and dependencies
7. Identify second tier tasks (application only tasks)

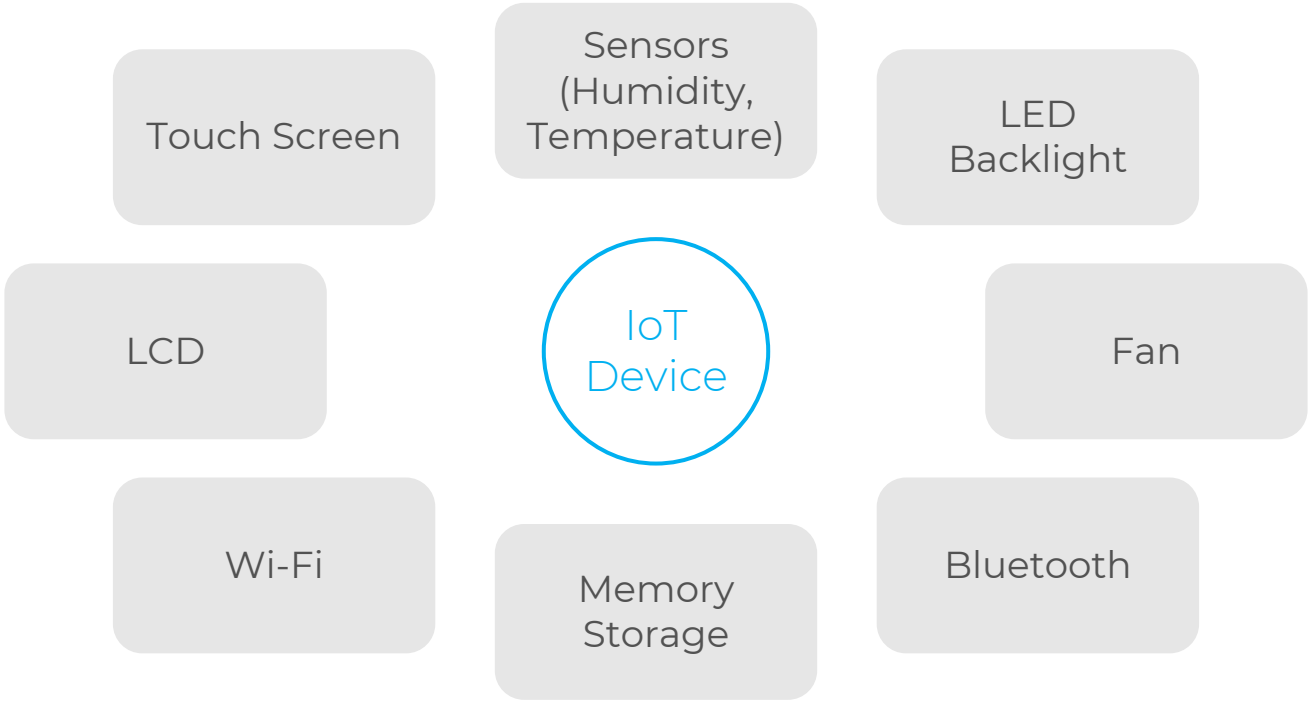
Task Decomposition – The Outside-In Approach

1. Identify the major components

- IoT Device
- LCD
- Touch Screen
- LED Backlight
- Wi-Fi
- Memory Storage
- Sensors (Humidity, Temperature, current, etc.)
- Fan
- Bluetooth

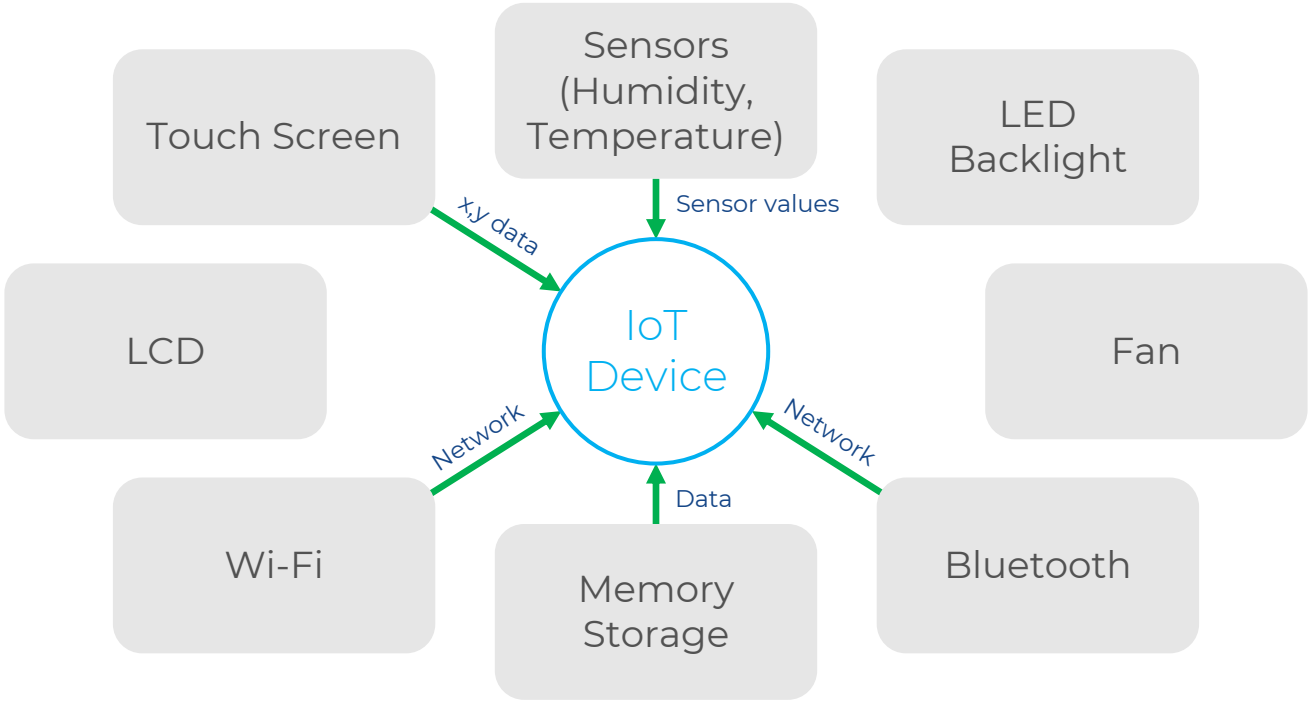
Task Decomposition – The Outside-In Approach

2. Develop a high-level system diagram



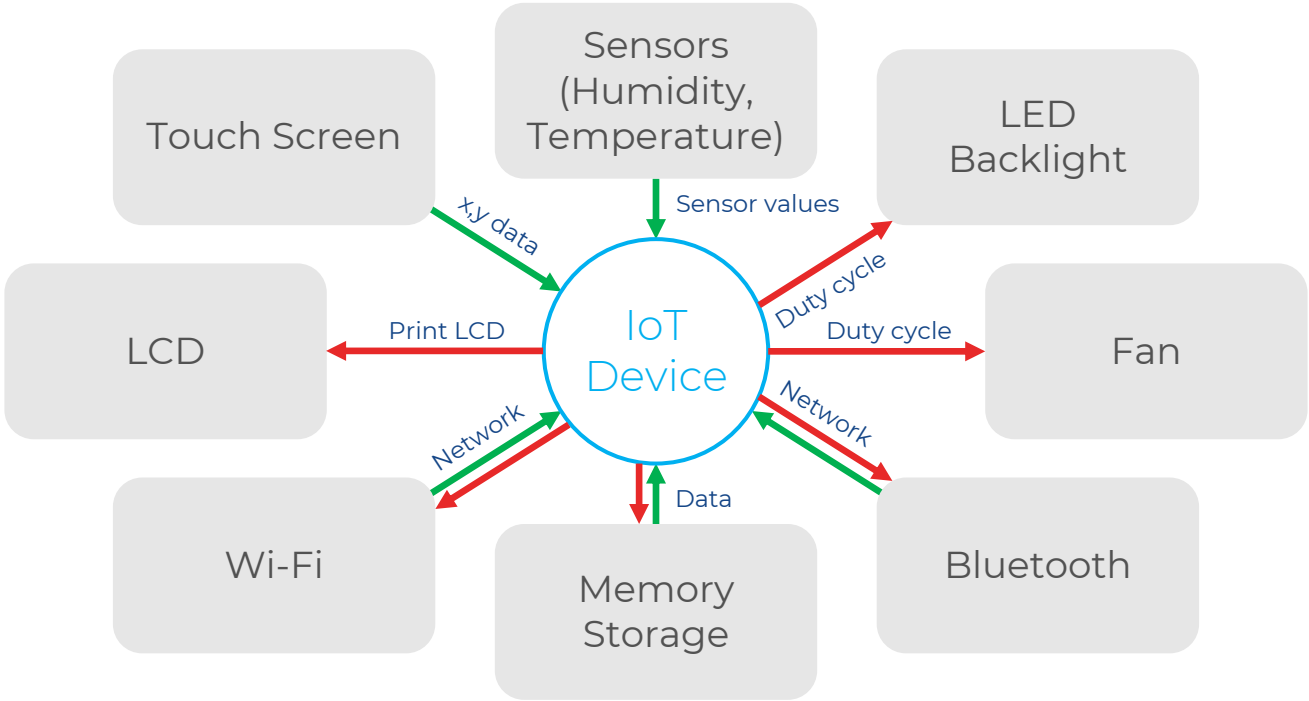
Task Decomposition – The Outside-In Approach

3. Label the inputs



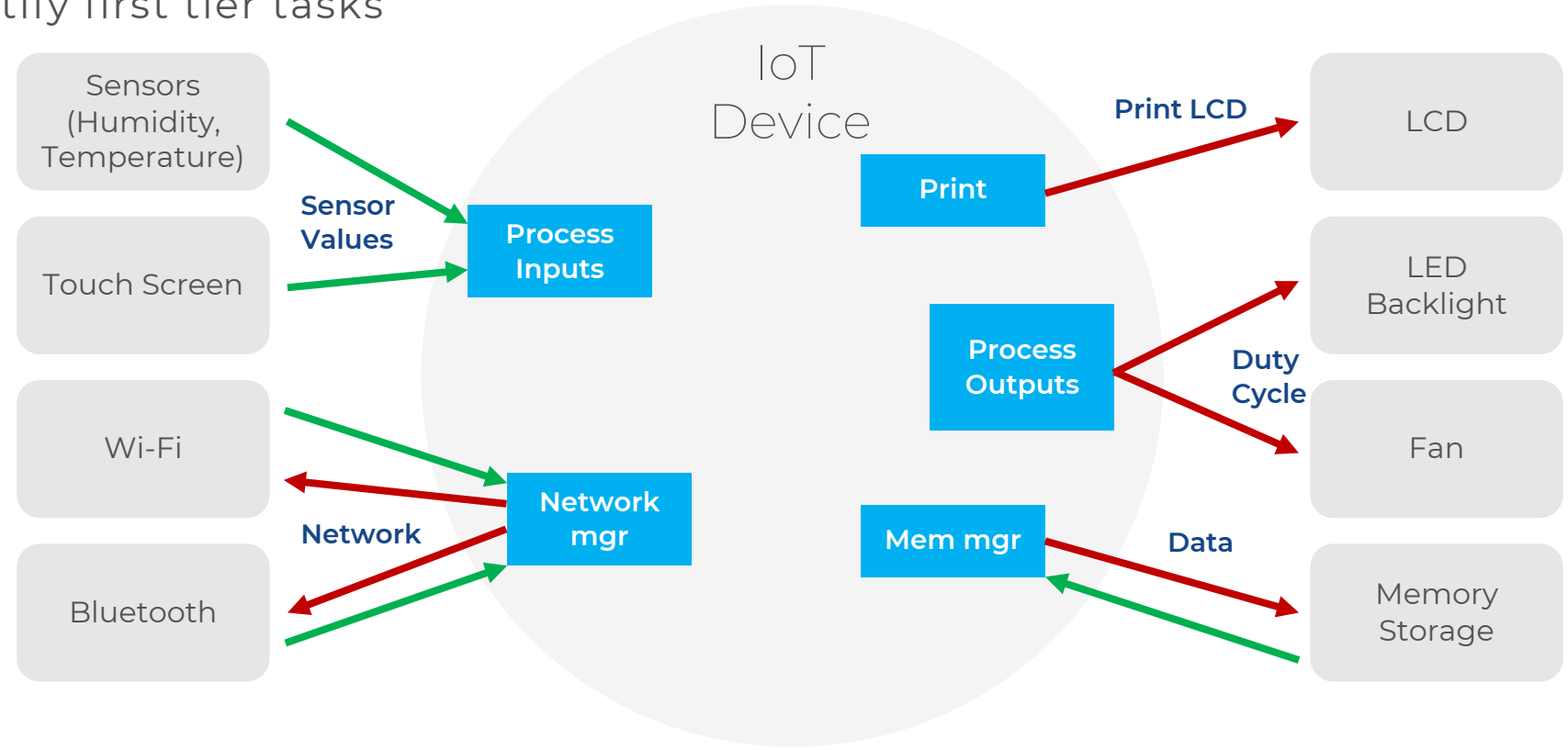
Task Decomposition – The Outside-In Approach

4. Label the outputs



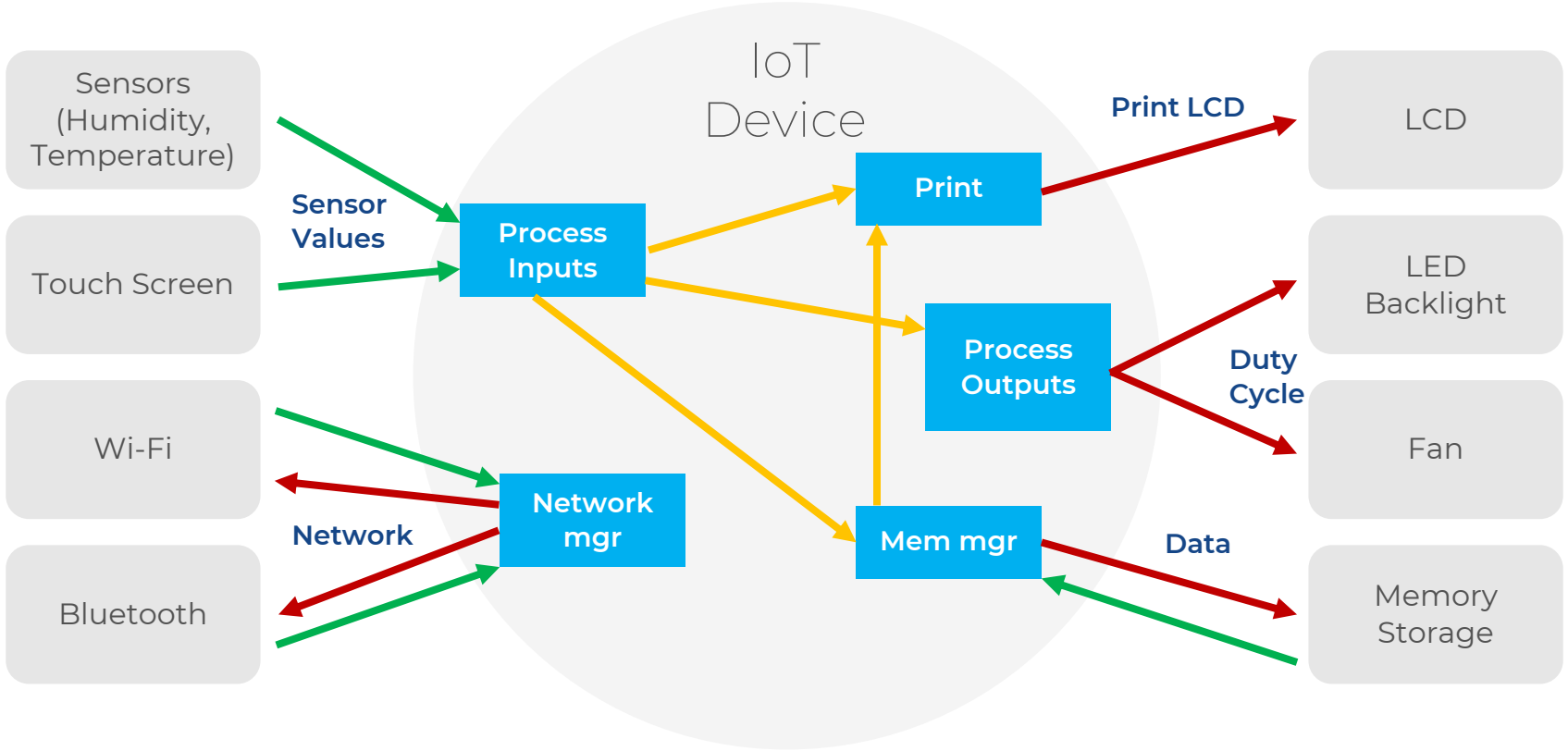
Task Decomposition – The Outside-In Approach

5. Identify first tier tasks



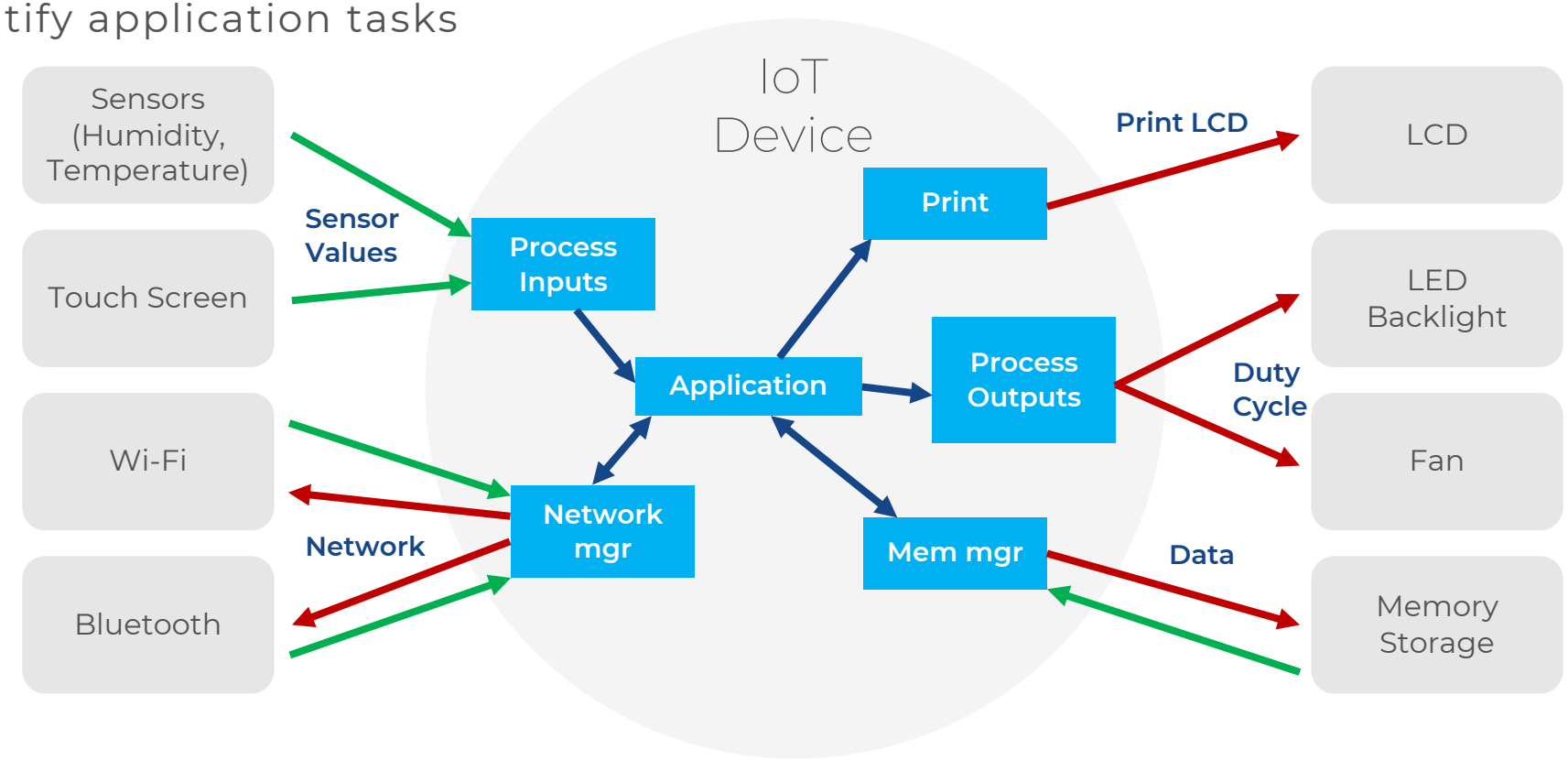
Task Decomposition – The Outside-In Approach

6. Determine concurrency and dependencies



Task Decomposition – The Outside-In Approach

7. identify application tasks



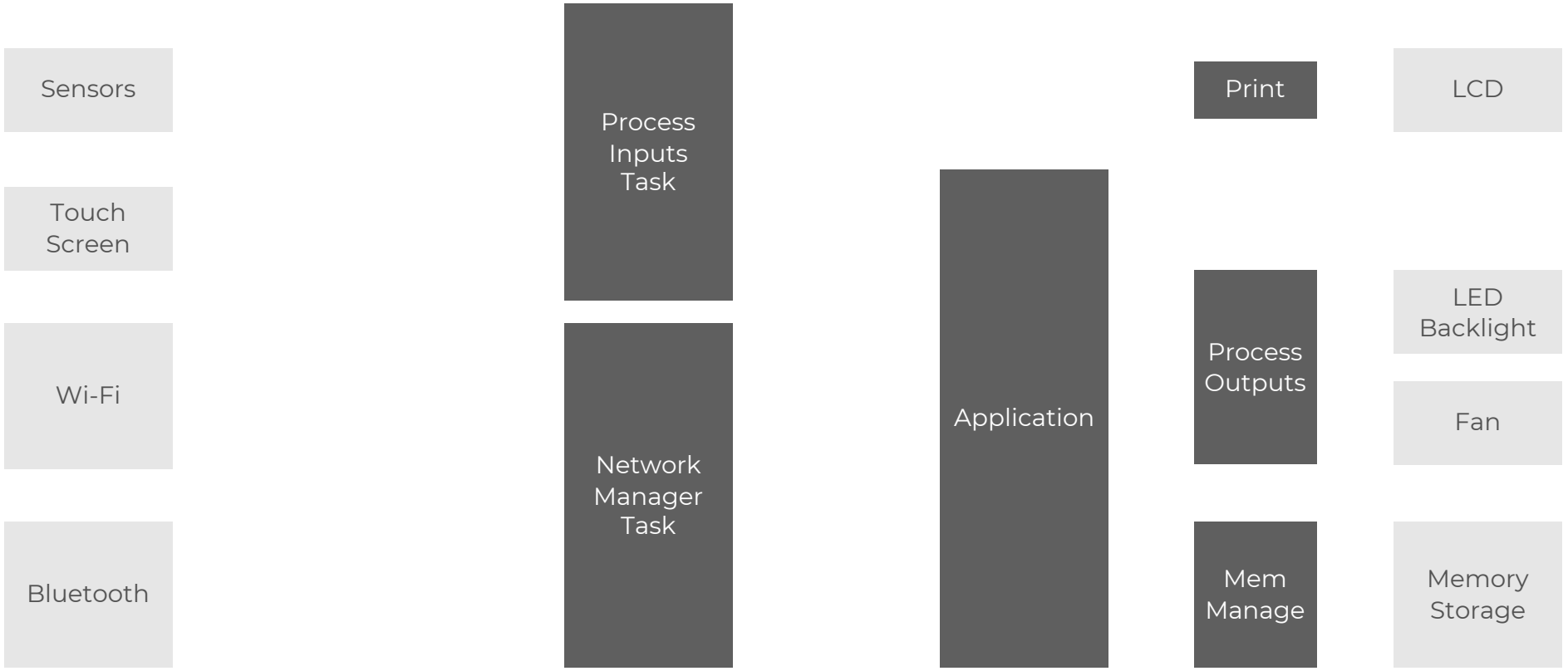
How do you design your RTOS application architectures?
- I don't use an RTOS
- Use the outside-in approach
- Just guess at the tasks needed
- Other

2

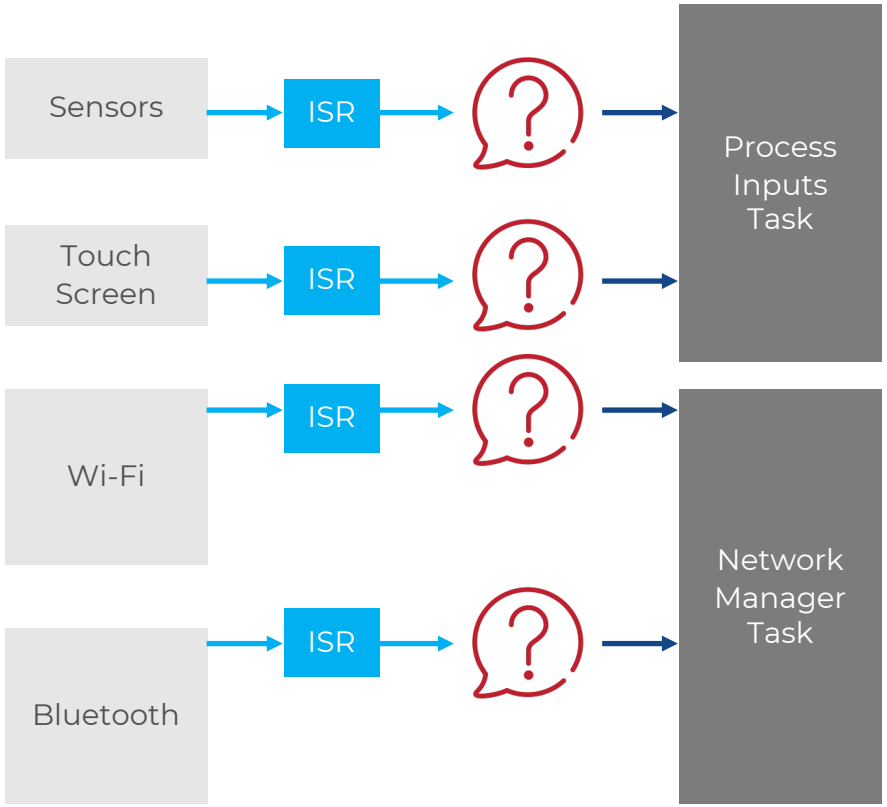
Application Data Flow

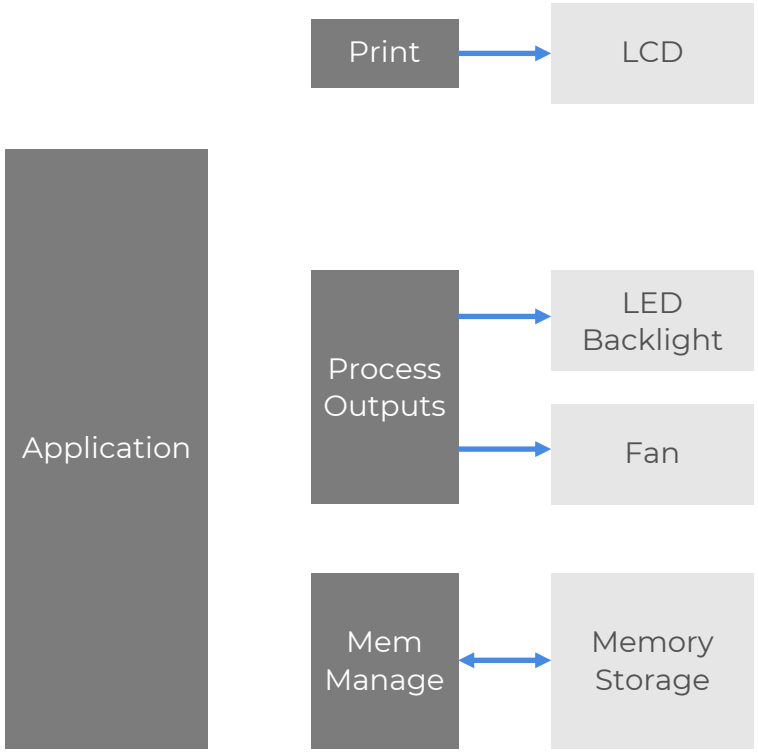
Embedded systems are all about receiving, transferring, processing and outputting data.

Application Data Flow

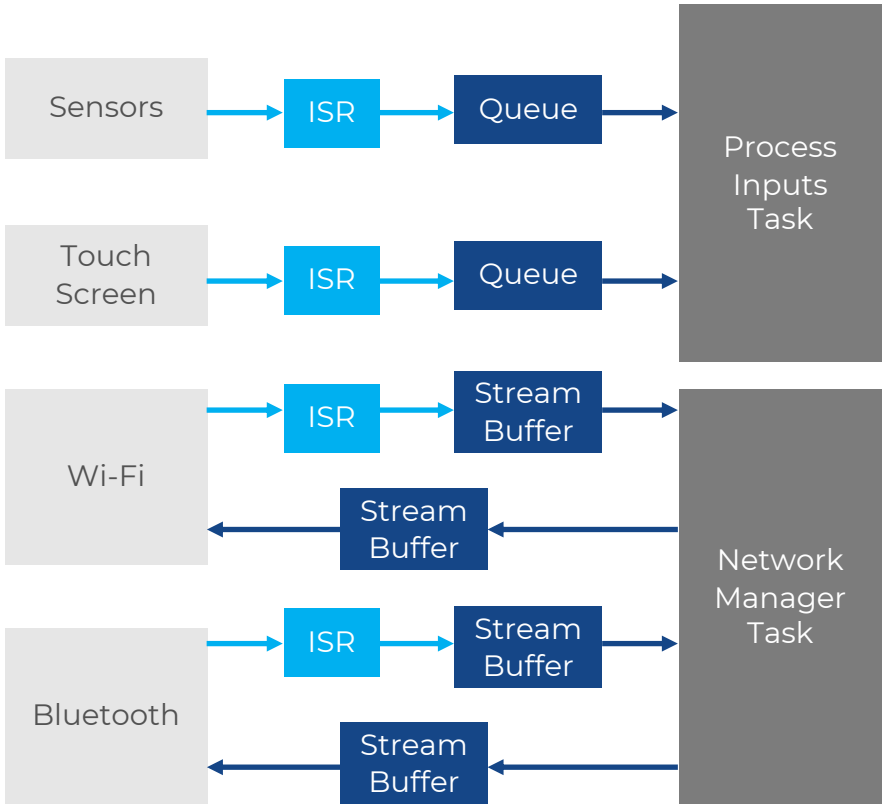


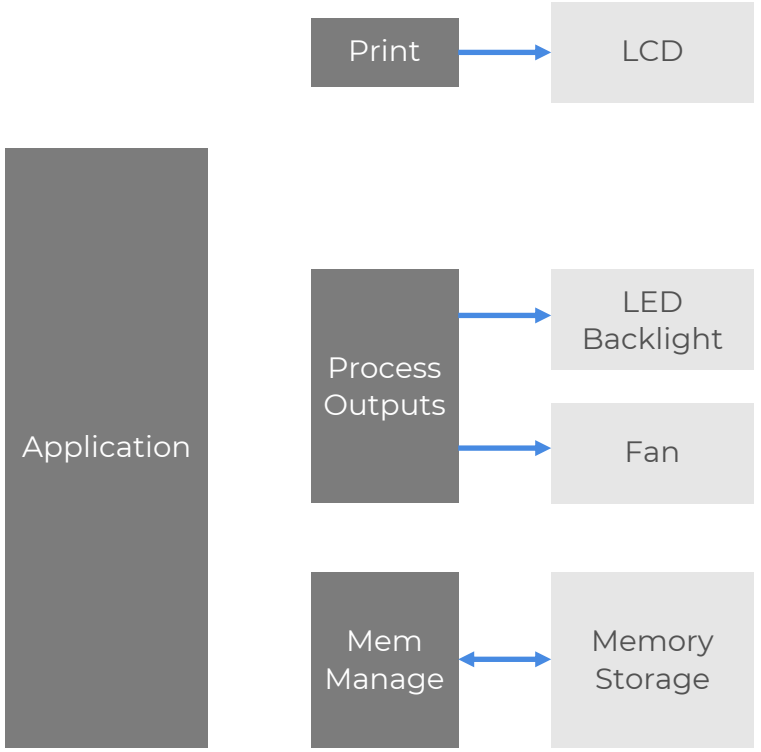
Application Data Flow



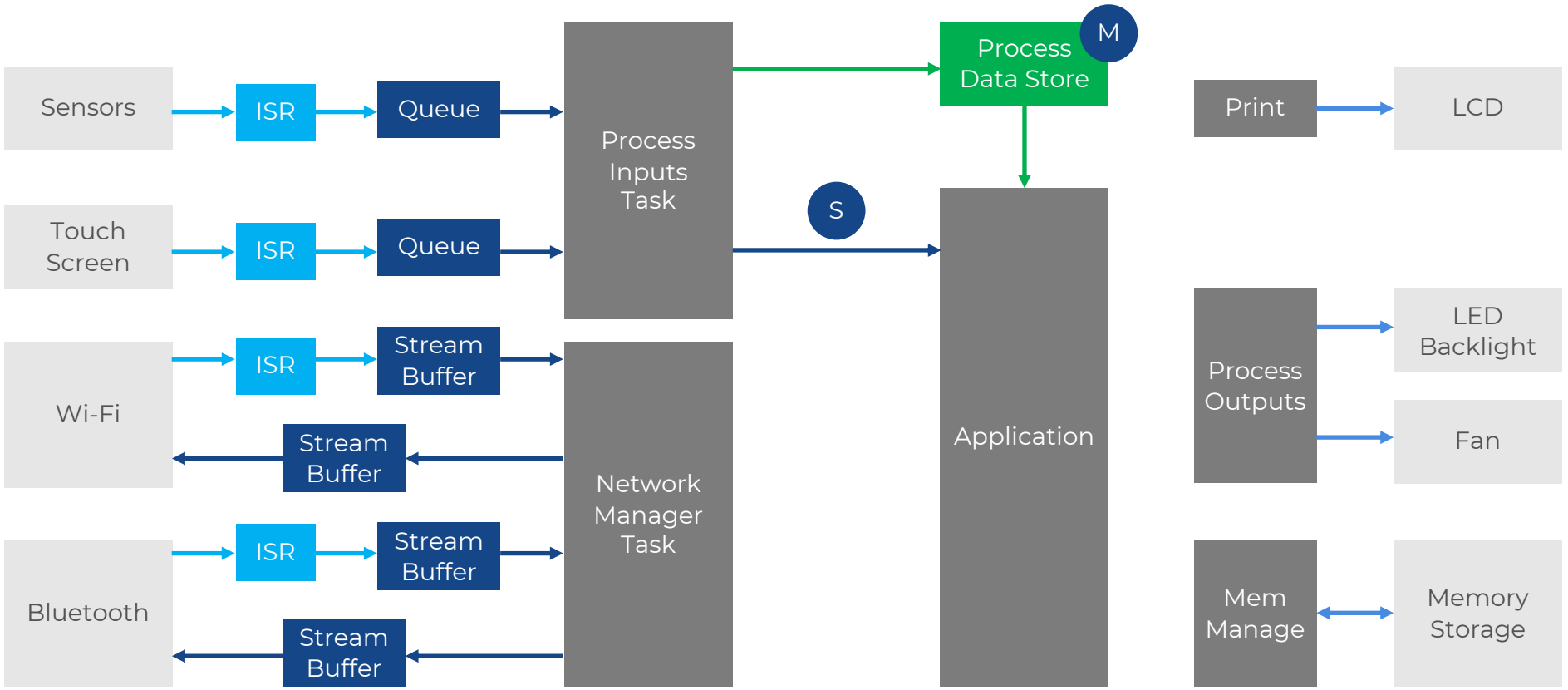


Application Data Flow

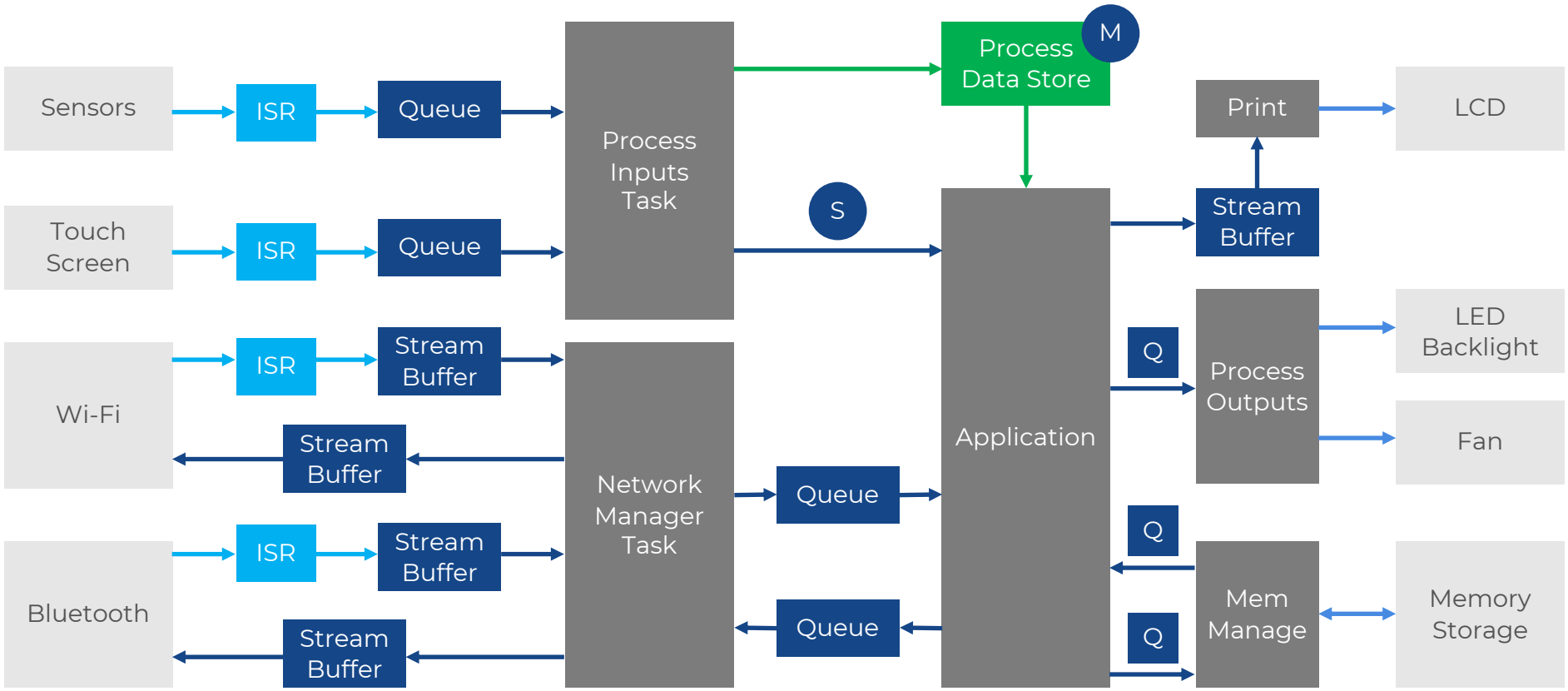




Application Data Flow



Application Data Flow



Do you identify your data assets and track how they flow through your application?
- Yes
- No

3 Rate Monotonic Analysis (RMA)

Rate Monotonic Analysis (RMA)

Rate Monotonic Scheduling (RMS) is an analysis technique to determine if all tasks can be scheduled to run and meet their deadlines.

Assumptions:

- Tasks are periodic
- Tasks are independent
- Preemptive scheduling is used
- Each task has a constant execution time (can use worst case)
- Aperiodic tasks are limited to start-up and failure recovery
- All tasks are equally critical
- Worst case execution time is constant

Rate Monotonic Analysis (RMA)

Basic RMS Schedulable Test

CPU Utilization(U)

$$\sum_{k=1}^{n} \frac{E_k}{T_k} \leq n\left(2^{\frac{1}{n}} - 1\right)$$

U(1) = 1.0
U(2) = 0.828
U(3) = 0.779
U(4) = 0.756
U(5) = 0.743
U(6) = 0.734
U(∞) = 0.693

	Execution	Period	Utilization
Task 1	15	100	0.15
Task 2	30	150	0.20
Task 3	60	300	0.20

Total Utilization = 0.15 + 0.20 + 0.20 = 0.52

0.52 <= U(3)
0.52 <= 0.779

Rate Monotonic Analysis (RMA)

Task	Execution (ms)	Period (ms)	Utilization	Priority	Comments
Network Manager	10	100	0.10	1	Estimating 10% load
Application	25	100	0.25	1	Processing and managing all inputs and coordinating output behavior
Print	25	250	0.10	5	Assuming 250 period for fast refresh to human. Serial printing is fast.
Process Outputs	5	1000	0.005	10	Minor calculation and updating PWM register
Memory Manager	200	1000	0.20	10	Slow memory to write. Storing all sensor data and logs.
Process Inputs	20	1000	0.02	10	1 Hz execution rate

How do you set your task priorities?
- Using RMA
- By how important I think a task is
- Guess
- Other

4 Going Further

Thank you for attending

Please consider the resources below:

- www.beningo.com
  - Blog, White Papers, Courses
  - Embedded Bytes Newsletter
    - http://bit.ly/1BAHYXm
  - Embedded Software Design
    - https://bit.ly/3PZCtNO

From www.beningo.com under

- Blog > CEC – Embedded Software Design Techniques

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key ELECTRONICS

BENINGO EMBEDDED GROUP

informa markets