Communication

In designing a system you have the option of making the different components communicate either by using I2C or Bluetooth Low Energy. Discuss in what cases you would favor one communication type over the other and explain the reasons.

I2C (Inter-Integrated Circuit Bus) is ideal for short-range, on-board communication between components in a single device due to its simplicity, low cost, and support for multiple slave devices. It's best for applications requiring fast, bidirectional communication within a PCB, like sensor data aggregation.

Bluetooth Low Energy (BLE) is favored for wireless, low-power, longer-range communication between different devices. It's suitable for IoT applications, wearables, or situations needing connectivity with smartphones or tablets, where flexibility and interoperability are crucial, albeit at a higher complexity and cost compared to I2C.

Computation

Consider a system for the selection of tomatoes in a food packaging plant. Tomatoes travel on transportation belts and IoT nodes equipped with cameras are used to capture images of moving tomatoes. Depending on its quality, each tomato will be sent through two different paths. The path is chosen automatically by using the result of a deep learning model that classifies tomatoes into one of the two classes. Considering the speed of the belt, the nodes need to shoot a picture every 0.25s.

The nodes are connected by means of a wireless protocol to a server. Should the machine learning model be run on the server or on the nodes? Explain the pros and cons of each solution and the reasons why one of the two should be favored over the other.

Run the model on the server for robust processing and easier model updates. Server-side execution leverages powerful hardware, ensuring accurate, fast processing crucial for real-time classification. It simplifies maintenance, as updating the model on one server is easier than updating multiple nodes. However, this approach increases network reliance and can cause latency issues.

Running the model on the nodes reduces latency and network load, providing faster, localized decision-making. This is beneficial for real-time applications but burdens the nodes with computational demands and complicates updates, also depending on the models weight, it might not fit and an optimization is needed to fit into the node. Given the system's need for speed and accuracy, server-side processing is generally favored.

DAC

Describe how to convert a digital signal to an analog one

DAC works by taking **digital binary values and converting them into a corresponding analog signal**, usually in the form of voltage or current. This **process involves interpreting the binary numbers and generating an analog output that accurately represents these digital values**.

An **ideal Digital-to-Analog** Converter (DAC) converts **digital numbers into impulses**, **smoothed by interpolation**. In contrast, **practical DACs** create a **stepped waveform from rectangular functions**, less smooth but simpler.

Considering these aspects:

1. **Resolution**: A DAC's resolution, indicated in bits, determines its output levels. For example, an 8-bit DAC can reproduce 256 levels. <u>Higher resolution means finer detail</u> in the analog output.
2. **Maximum Sampling Rate**: This is the <u>fastest rate at which a DAC can operate and still produce accurate output</u>. It impacts how well the DAC can convert high-frequency digital signals to analog.
3. **Monotonicity**: This ensures that the <u>DAC's output consistently reflects the direction of the input change, without erroneous dips or spikes</u>.
4. **Total Harmonic Distortion and Noise** (THD+N): Measures <u>how much distortion and noise the DAC adds to the signal</u>. Lower THD+N indicates higher fidelity.
5. **Dynamic Range**: The <u>range between the largest and smallest signals a DAC can accurately reproduce</u>. It affects the DAC's ability to handle signals with varying amplitudes.
6. **Jitter**: <u>Variations in the timing of the DAC's sampling process</u>. Jitter can degrade sound quality in audio applications.

Scheduling

Suppose there are three processes with the following execution times and periods:

- P1 - exec. time 1 - period 4
- P2 - exec. time 2 - period 6
- P3 - exec. time 1 - period 2

All processes are ready at the same time (24).

What is the execution order of the three processes when an EDF scheduling algorithm is adopted? Consider a full hyperperiod. Will the order change if SJF was adopted instead?

EDF: Earliest Deadline First.
SJF: Shortest Job First,
Hyperperiod: the least common multiple (LCM) of the periods of multiple processes.

The periods are 4, 6, and 2 for processes P1, P2, and P3.
Calculate LCM of 4, 6, 2 = 12.
The utilization of the CPU is: 1/4 + 2/6 + 1/2 = 1.08, Scheduling is **NOT** feasible.

| : End of Period.
|X: Start of process
X|: End of process

For EDF (Earliest Deadline First) scheduling, processes are prioritized based on the earliest deadline. Since all processes are ready at the same time (24), their initial and the shorter the period, the higher the priority (the period is the deadline). We will consider a full hyperperiod and determine the execution order.

<table>
<tr><td>P3</td><td>_</td><td>|X|</td><td>_</td><td>_|</td><td>|_</td><td>|X|</td><td>_</td><td>_|</td><td>|_</td><td>|X|</td><td>_</td><td>_|</td></tr>
<tr><td>P2</td><td>_</td><td>_</td><td>_</td><td>|X</td><td>_</td><td>_|</td><td>|_</td><td>|X</td><td>_</td><td>_</td><td>_</td><td>X||</td></tr>
<tr><td>P1</td><td>|X|</td><td>_|</td><td>||X|</td><td>_|</td><td>||X|</td><td>_|</td><td>||X|</td><td>_|</td><td>||X|</td><td>_|</td><td>||X|</td><td>_|</td></tr>
<tr><td>T</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td></tr>
</table>

This sequence will repeat until the hyperperiod is completed.

For SJF (Shortest Job First), processes are prioritized based on their execution times. Since P1 and P3 have the same execution time (1) and are shorter than P2 (2), they will be prioritized, but their order will be determined by their deadline.

SJF:

<table>
<tr><td>P3</td><td>_</td><td>|X|</td><td>_</td><td>_|</td><td>|_</td><td>|X|</td><td>_</td><td>_|</td><td>|_</td><td>|X|</td><td>_</td><td>_|</td></tr>
<tr><td>P2</td><td>_</td><td>_</td><td>_</td><td>|X</td><td>_</td><td>_|</td><td>|_</td><td>|X</td><td>_</td><td>_</td><td>_</td><td>X||</td></tr>
<tr><td>P1</td><td>|X|</td><td>_|</td><td>||X|</td><td>_|</td><td>||X|</td><td>_|</td><td>||X|</td><td>_|</td><td>||X|</td><td>_|</td><td>||X|</td><td>_|</td></tr>
<tr><td>T</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td></tr>
</table>

In this case, EDF and SJF have the same outcome.

Watchdog

Consider the control system of a quadcopter drone that makes use of a watchdog. Explain how the watchdog can be configured in such a system and describe the possible advantages of adopting this solution.

Watchdog: an electronic timer that is used to detect and recover from computer malfunctions.

- – E.g., crashes of the operating system or hangs in the application.

Commonly found in embedded systems and other computer-controlled equipment where humans cannot easily access the equipment or would be unable to react to faults in a timely manner

During normal operation, the application regularly resets the watchdog timer to prevent it from timing out, In computers that are running operating systems, watchdog resets are usually invoked through a device driver, If the timer expires, it will generate a timeout signal, and trigger some measures to handle the issue.

In the case of a quadcopter (commercial drone), a watchdog can be used to detect if the system or a component freezes or failures. The system will reset the watchdog's timer regularly, and when the watchdogs detects no more resets it will initiate predefined actions to mitigate the problem, for example an auxiliary landing or hovering mode can be triggered on the quadcopter, while it also triggers a system restart.

Analog and digital numbers

An analog value is translated into a digital number by means of:

- Connecting the sensor to a digital pin
- Sampling
- Quantization
- An interrupt

CANBus

CANBus is:

- A wired network protocol suitable in environments where electromagnetic noise is present
- unsuitable for communications that exceed a 2m distance
- a point-to-point wired protocol
- an obsolete wireless network protocol

In the CANBus protocol:

- quality of service is provided based on the address of the sender
- quality of service is provided based on the address of the receiver
- quality of service is not available
- quality of service is set as requested by each device when it is connected

Continuous dynamics systems

Continuous dynamics systems:

- Are physical systems that can be described by means of a mathematical model
- Are physical systems that cannot be described by a mathematical model
- Cannot be controlled by using a microprocessor
- They require analog sensors

Cross compile

A cross compiler:

- Optimizes machine learning models
- Produces optimized machine code for ARM architectures
- Is a compiler that runs on a machine with an architecture, but it produces machine code for a different target architecture
- Runs on an embedded architecture and produces machine code for PCs

Discrete dynamics systems

Discrete dynamics systems:

- Require a closed-loop control
- Are the only type of systems that can be controlled by microprocessors
- Can be described by a state machine
- State machines can be used if the system is open loop, they cannot be used otherwise

Interrupts

Interrupts:

- Is another name of *polling*
- Provide the ability to handle asynchronous events
- Can deal only with synchronous events
- Keep the CPU busy all the time

mBed OS

ARM mBed OS is:

- A general-purpose operating system
- A set of libraries for embedded devices
- An operating system for embedded devices
- A set of libraries providing support for multithreading for embedded devices

Open loop control

Open loop control:

- Control is adjusted based on the feedback
- Control is adjusted based only on the input
- Can be implemented by using a PID controller
- Can be used only for discrete dynamics systems

PID

A PID controller provides a control that has three components

- proportional to the error, integral of the error, and derivative of the error
- proportional to the input, integral of the input, and derivative of the input
- proportional to the output, integral of the output, and derivative of the output
- proportional to input, output, and error

Porting machine learning models in edge nodes

When porting machine learning models to edge nodes, the pruning and quantization techniques can be used. What is the purpose of those techniques?

- Improve model accuracy
- Minimize memory footprint and required computational resources
- Increase the memory footprint
- Include real-time capabilities

Precision of a sensor

Precision of a sensor:

- corresponds to the higher threshold on measured values
- describes repeatability of measures
- is the smallest absolute difference between two values of a physical quantity whose sensor readings are distinguishable
- is not related with the dynamic range

Priority inversion

Priority inversion is:

- A problem that cause high-priority processes to wait for lower-priority processes to obtain a shared resource.
- A problem that is never present in real-time operating systems
- A technique adopted in the BLE communication protocol for handling quality of service
- A technique for dealing with low-priority processes that are not assigned the CPU for a very long time

Real-time

Real-time is obtained when:

- Response from the system is as quick as possible
- Response is within predetermined deadlines or the system can sustain a specific throughput, depending on the requirements
- Interrupts are used
- interrupts are not used

Sampling rate

A node is used to monitor a lamp that needs to maintain a specific luminosity level with a specific level of accuracy. The lamp can vary its luminosity according to a sinusoid with a period of 0.1s. The value of the luminosity is averaged over a 0.5s period and the standard deviation is computed in the same period of time.

What is the best sampling rate to be adopted for the luminosity sensor?

The lamp can vary its luminosity as fast as 0.1s

10 samples per second → T = 0.1s → fs = 10Hz

Nyquist rate: twice the bandwidth of a band-limited function or a band-limited channel, i.e., twice the highest frequency (of interest) of the signal.

A faithful reproduction of the original signal is only possible if the sampling rate is higher than the **Nyquist rate.**

10Hz * 2 = 20Hz

- 1Hz
- 2Hz
- at least 10Hz
- at least 20Hz

Security

Supporting security requires suitable resources (computational resources, energy, memory). Which of the following options is best? This one can't be more obvious.

- Disregard security during the design of the device and add it when the customers start complaining.
- Do not use any security mechanism, as they cannot be integrated into embedded devices
- Design systems without security, but overprovision resources for later integration of security mechanisms
- Design system security from the beginning, planning system resources with security included

Software on bare metal

When running software "on bare metal":

- The operating system schedules threads
- Tasks with different periodicity are handles automatically
- The operating system is set in real-time mode
- The developer needs to handle task scheduling and coordination

SPI

In SPI:

Serial Peripheral Interface: among others, it works on **Full Duplex** Protocol.

- Communication between master and the slave devices is parallel and it happens in one direction at a time
- Communication between master and the slave devices is parallel and it happens in both directions at the same time
- Communication between master and the slave devices is serial and it happens in one direction at a time
- Communication between master and the slave devices is **serial** and it happens in **both directions** at the same time

Tasks

When a system needs to deal with multiple tasks that have different timing requirements, which is the best option?

- Use multiple threads, one for each task
- Use a single thread that includes all tasks with interrupts
- Use a single thread that includes all tasks
- Use a timer

TMR

In TMR:

TMR: Triple Modular Redundancy

- There are ten copies of each critical component and their output is evaluated by means of a voter
- There are three copies of each critical component and their output is evaluated by means of a voter
- A voter is used to decide if a component is faulty
- The computation is repeated three times and the correct results is selected by the voter