Bus Systems

Communication Principles

Prof. Dr. Reinhard Gotzhein, Dr. Thomas Kuhn

Principles of Communication Systems

Learning goals of this section

- Understand principles and description of communication systems on *system level*
  - Bus system topologies
    - Physical and logical bus topologies
    - Advantages and disadvantages of specific bus topologies
  - Grey communication channels
    - Possible communication failures
    - Relationship between Failures and failure modes
  - Bus System and Protocol behavior
    - Finite state machines describe bus system behavior on layers of ISO/OSI model

Perception vs. Reality

What we perceive and how we interpret it depend on the frame through which we view the world around us.

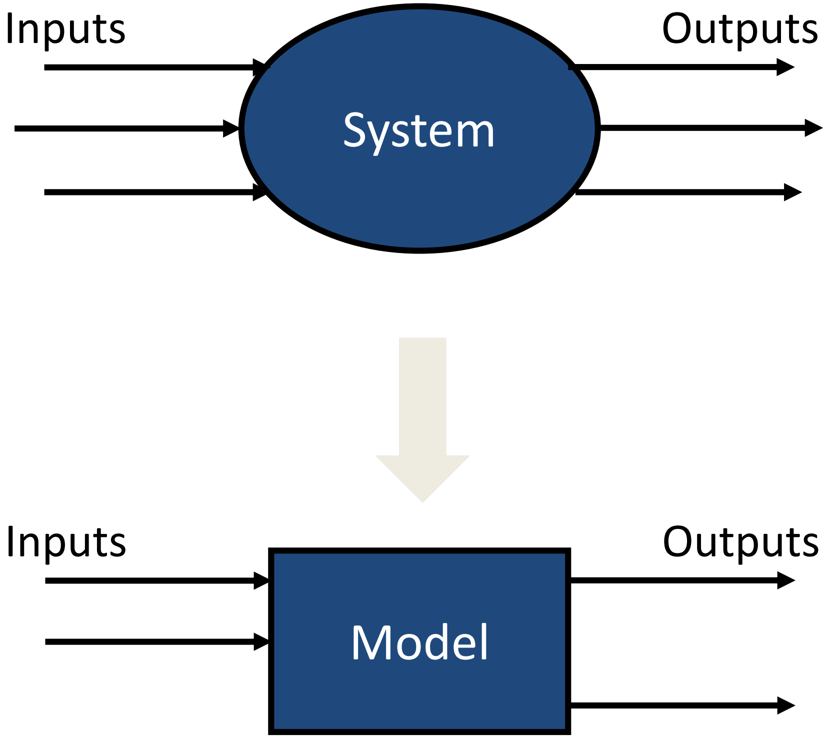
System Level

- A **system** is a combination of components that act together to perform a function not possible with any of the individual parts

  Architecture describes how the system has to be implemented

- A **model** is an abstracted (formal) description of the system, which covers selected information.

  Describes how the system works

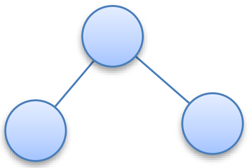


Communication principles

Bus system topologies – Overview

- A model of the main communication principles of a bus system

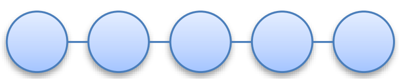
- Bus systems differ with respect to topology
  - Topologies define communication paradigms
  - Example: One-to-one communication vs. One-to-many communication



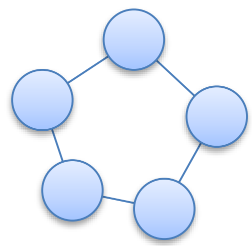
- We distinguish between physical and logical topologies
  - Physical topologies define physical communication links within a network
    - Defined by hardware
  - Logical topologies define logical communication links
    - Defined by protocol

Communication principles

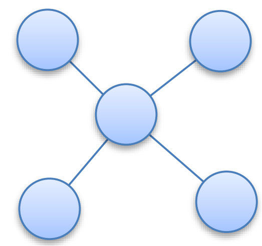
Bus system topologies – Overview



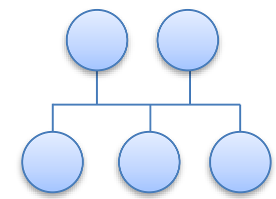
Line Topology



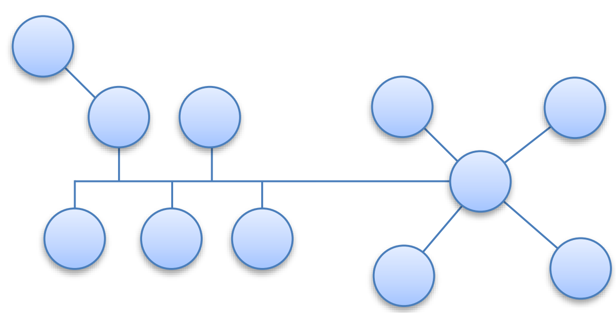
Ring Topology



Star Topology



Bus Topology

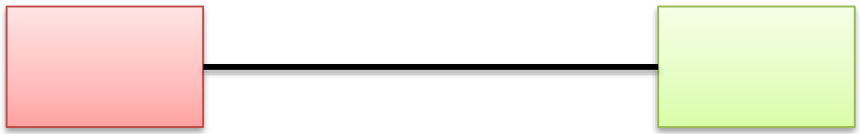


Mesh Network

Communication principles

Bus system topologies

- Point to point connections (Line Topology)

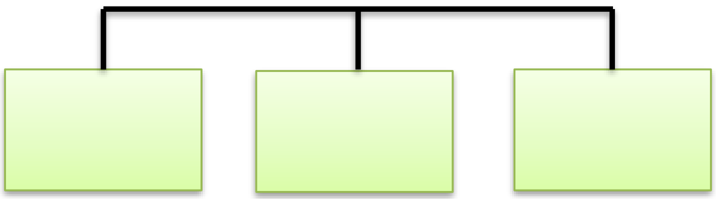


- Physical topology: Direct connection (wire) between two nodes
  - Used to connect simple sensors or actuators to smarter processing devices
    - No need for medium arbitration → predictability
  - Both nodes must agree on communication protocol
    - This includes physics (e.g. power levels), logical aspects (framing), error detection…
- Logical topology: Direct communication link between two nodes
  - One-to-one communication

Communication principles

Bus system topologies

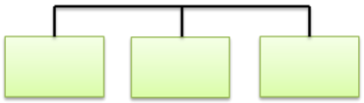
- Bus topology



- Physical: Shared line between a number of nodes
  - All nodes must agree on common protocol for using the bus system
  - Concurrent access requests must be handled, e.g. through medium arbitration
  - One failing node can render the bus useless for all other nodes (babbling idiot)
- Logical: Information sent by one node is transmitted to all other nodes
  - One-to-many communication

Communication principles

Bus system topologies



- Manages access to shared communication medium
  - Relevant if only one node may use medium at a time
  - Contention based and contention free bus access schemes

- Contention based access
  - Nodes use contention scheme to decide about transmission rights
    - Example: ALOHA – Every participant attempts to transmit at any time and detects failed transmission based on missing post-transmit confirmation from receiver
  - Contention schemes affect predictability
    - Priority based (predictable), Random (not predictable), Mixed (stochastic)
    - When is a contention scheme predictable?
  - Contention schemes affect fairness
    - When is an access scheme fair / not fair?

Communication principles

Bus system topologies

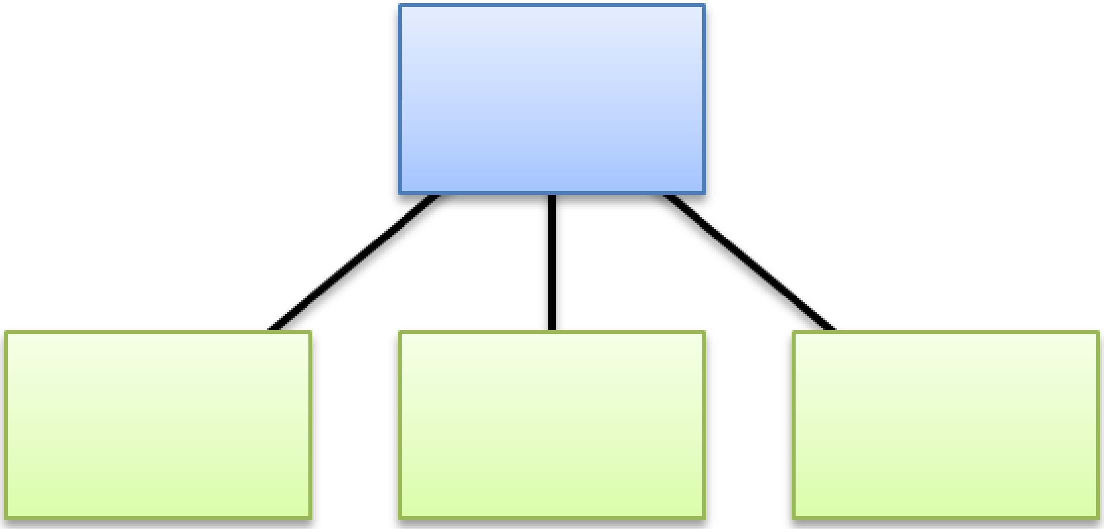
- Contention free access
  - Medium access is resolved without contention

  - Examples
    - Time division plan is replicated among all nodes (static access scheme)
      - This requires time synchronization between nodes
    - Modulation effects are used to divide a medium: Frequency division, code division
    - Central master node assigns transmission rights (Repeatedly polls nodes)
    - Token is passed between nodes

Communication principles

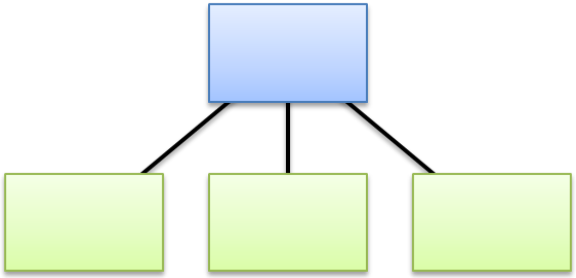
Bus system topologies

- Star topology



Communication principles

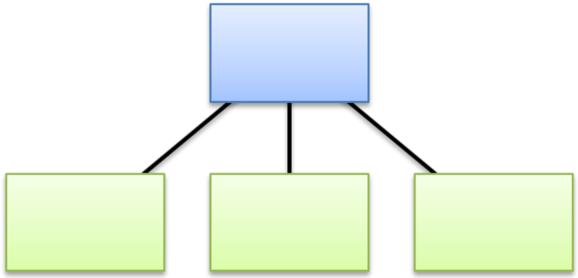
Bus system topologies



- Physical star
  - All communication is between star and leaf nodes
    - Supports concurrent communication between nodes
  - Robust against link failures
    - Link failures only affect the connected node or subnet
  - (Mostly) robust against node failures
    - Failure of regular nodes only affect that node
    - Failure of central node breaks whole network
  - May be robust against “babbling idiots”
    - Master node can perform link isolation

Communication principles

Bus system topologies



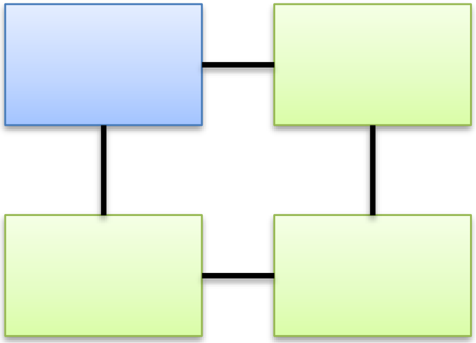
- What is logic star?

- Master based communication
  - Performance → all communication via master
  - Predictability → contention based access ensures predictability
  - Reliability → communication depends on master node
  - Scalability → one master must know all transmission needs

Communication principles

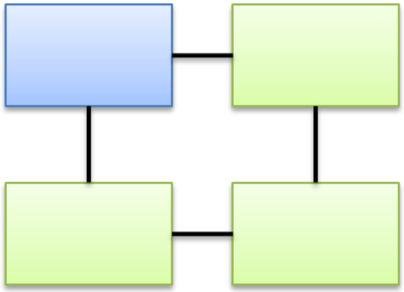
Bus system topologies

- Ring topology
  - Communication token(s) is/are passed between directly connected nodes
  - Exist with and without master nodes



Communication principles

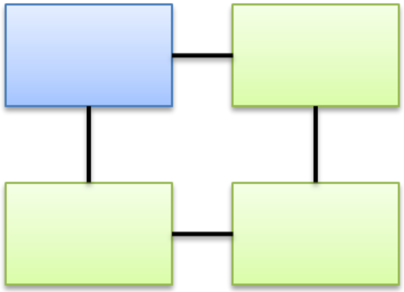
Bus system topologies



- Physical Ring Topology
  - Network topology consists of pairwise linked nodes
    - Contention free access and short cables enables fast transmission rates
  - Ring networks are predictable
    - Quality of Service guarantees (QoS) are supported
    - E.g. maximum delay/jitter bounds may be given based on maximum frame size and node count

Communication principles

Bus system topologies



- Logical ring topology
  - Transmission token is passed between nodes
    - Manages transmission rights
    - Cooperative approach
    - Decentral contention free network access scheme

Communication principles

Bus system topologies

- Define the basic communication principles of a bus system
- Phyiscal topology is defined by hardware
  - Affects performance, cost, range, predictability
  - Constrained by medium, hardware cost and complexity, available technology
- Logic topology is defined by protocol
  - Regulated medium access
  - Integrates additional non-functional properties, e.g. energy saving mechanisms

Communication principles

Grey communication channels

- Bus systems realize communication channels
  - Communication channels transmit information between communicating participants

- Grey communication channels
  - Untrusted communication channels between applications
  - In fact, all real-world communication channels must be considered “grey”
  - Absence of one or more communication failures cannot be guaranteed
  - Communication systems must address communication failures

- Failures
  - Which failures could occur in communication channels in general?
  - Which high-level failures occur because protocols try to mitigate failures?

Communication principles

Grey communication channels

- Defined possible failures in E2E communication (according to ISO 26262)
  - Repetition of information
  - Loss of information
  - Delay of information
  - Insertion of information
  - Masquerading of information
  - Incorrect addressing of information
  - Incorrect sequence of information
  - Corruption of information
  - Asymmetric information sent from sender to multiple receivers
  - Information from a sender only received by subset of receivers
  - Blocking access to communication channel

Communication principles

Grey communication channels

- Repetition of information
  - Information is received more than once
- Loss of information
  - Information or parts of information are removed from a communication stream
- Delay of information
  - Information is received later than expected
- Insertion of information
  - Additional information is inserted into communication stream

Communication principles

Grey communication channels

- Masquerading of information
  - Incorrect information is accepted as correct by receiver
- Incorrect adressing of information
  - Information is accepted from incorrect sender or by incorrect receiver
- Incorrect sequence of information
  - Sequence of information in stream of transmitted information is altered
- Corruption of information
  - Communication fault that changes information

Communication principles

Grey communication channels

- Asymmetric information sent from sender to multiple receivers
  - Receivers do receive different information from same sender

- Information from a sender only received by subset of receivers
  - Communication fault that causes some receivers to not to receive information

- Blocking access to communication channel
  - Access to a communication channel is blocked

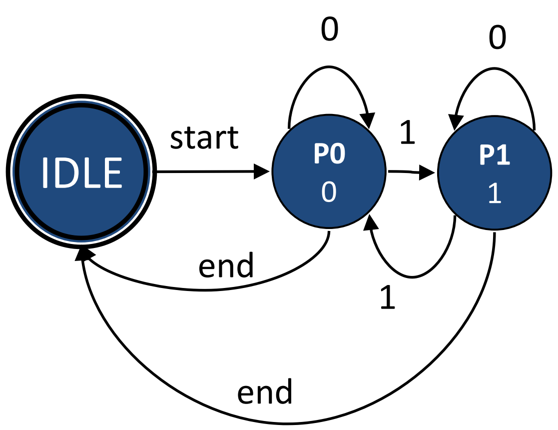
Finite State Machines

Finite state machines define behaviour as a system of countable states

- System behavior is characterized by states and state transitions
  - Every state machine is in one state at every time

- State transitions are triggered based on events, temporal logic, or spontaneous

- States generate outputs (Moore automatons)
- States generate outputs based on inputs (Mealy automatons)

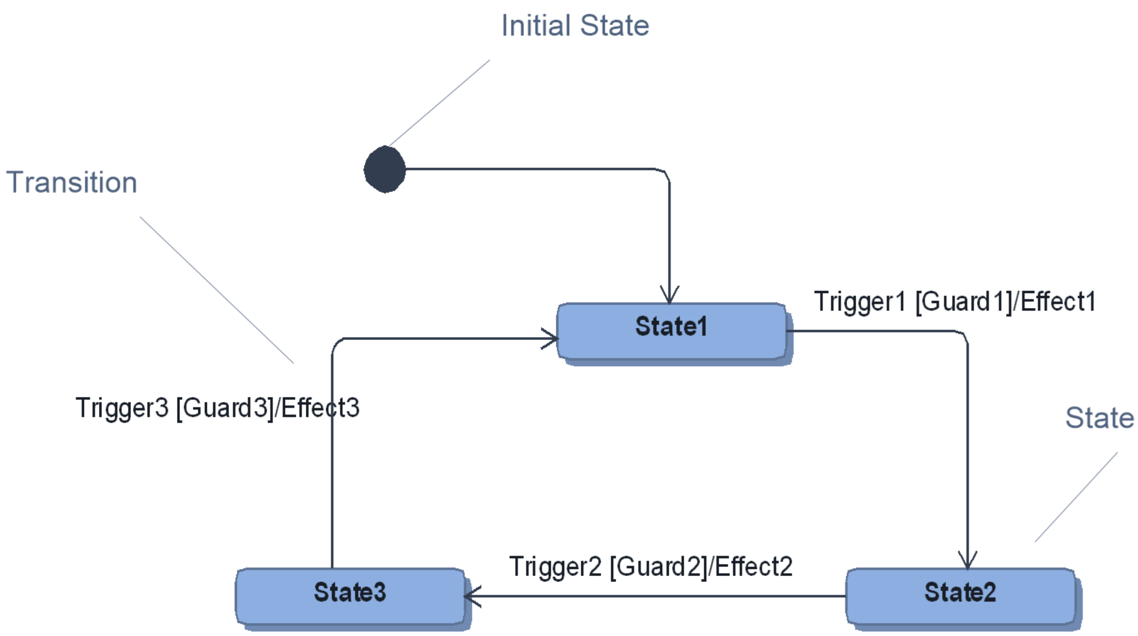


FSM calculating even parity
(Moore automaton)

Finite State Machines

UML state machines

- UML state machines have characteristics of both Moore and Mealy state machines



Finite State Machines

Different actions can be assigned to states

- **Entry actions**:
  Entry actions are executed for a state when the state is entered (becomes active).

- **Exit actions**:
  Exit actions for a state are executed when the state is active and a transition out of the state is taken.

- **Do actions:**
  Do actions for a state are executed as long as the state is active

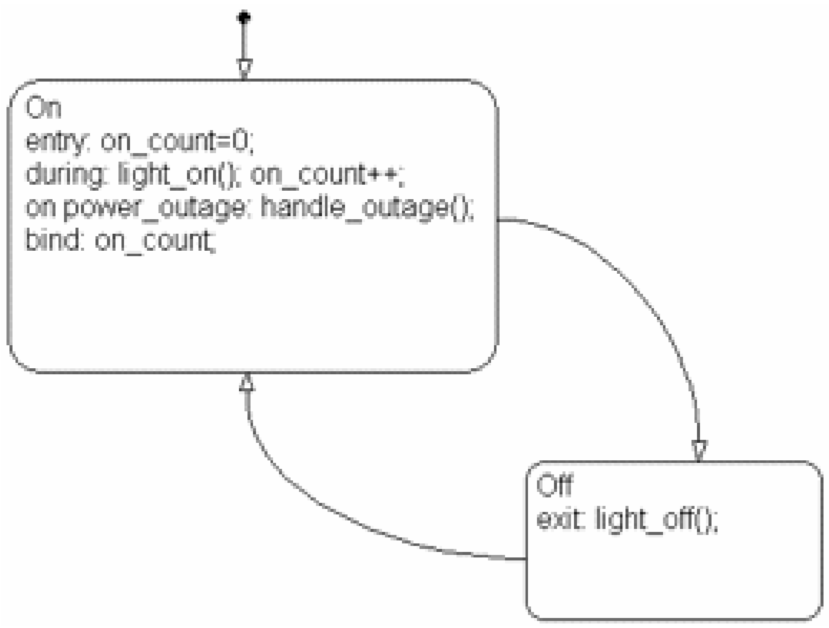
Temporal Logic Operators

Temporal Logic Operators may be used as event triggers

- Support the realization of temporal dependencies in finite state machines

- **after (n, E)**
  is true after event E has been executed at least n times since activation of source state
- **before(n, E)**
  is true as long as event E has been executed less than n times since activation of source state
- **at(n, E)**
  is true at the n^{th} occurance of event E since activation of source state
- **every(n, E)**
  is true at every n^{th} occurrence of the event E since activation of source state

State actions – Example



Transitions

- Transitions have the following syntax:

  `event[guard]/transition_action`

- **Event Trigger**
  specifies an event that causes the transition to be taken, provided the condition, if specified, is true. Specifying an event is optional. The absence of an event indicates that the transition is taken upon the occurrence of any event. Multiple events are specified using the OR logical operator (|).

- **Guard**
  specifies a Boolean expression that, when false, prevents firing of the transistion.

- **Transition Action**
  The transition action is executed after the transition destination has been determined to be valid provided the condition, if specified, is true. If the transition consists of multiple segments, the transition action is only executed when the entire transition path to the final destination is determined to be valid. Precede the transition action with a backslash.

Transitions

- Transition with event and guard fires…

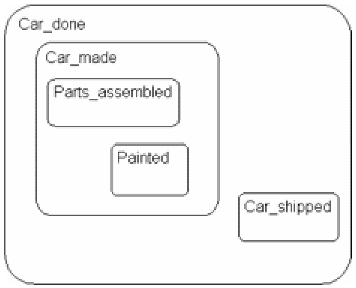
- Transition with event only fires…

- Transition with guard only fires…

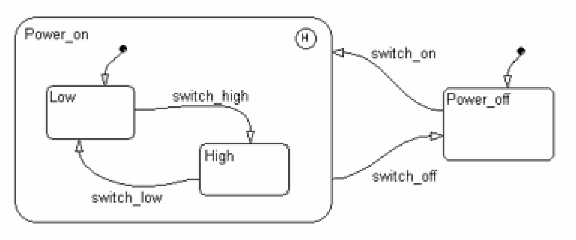
- Transition without guard and event fires…

State-Hierarchy

- **State Hierarchy**
  - States can be hierarchically structured
  - States can have an arbitrary number of sub states
- **History States**
  - History states are supported by **History Junctions**
  - If a hierarchical state containing a history junction is entered …
    - … the first time: the initial sub state is activated (*Low*)
    - … any other time: the last active sub state is activated (i.e. the sub state that was active when the superordinate was deactivated)



State Hierarchies



History Junctions

Concurrent States

Concurrent state machines run in parallel

- Supports limited concurrency

- Example: