

Em bedded
System



§ 2. Microcontrollers (μ C, MCU)

- * Structure
- * Digital I/O
- * Interrupts
- * Timer/ Counter
- * Analog I/O

1. * Stand-alone device for embedded applications

≈ low-end microprocessor + memory + I/O +
additional peripherals

* Not a general purpose device (ad-hoc device)

* cost-optimized control unit for particular application area

* More general than ASIPs & SoCs Systems-on-Chips
Application Specific Instruction Set Processors

2. How to access internal blocks

Memory: - All memory types share a common address range or
- different memory types are mapped into one address range.

Digital I/O, on-chip peripherals: by dedicated registers

3. Special registers for Digital I/O pins

- Data Direction Register (DDR)
- Port Register : (PORT)
- Port Input Register (PIN)

→ Examples < LED control
Reading Button

* Digital Output: the controller drives the pin according to PORT values of that pin.

* Digital Input: Sampling worst-case: ~1 clock cycle
↳ Impulses shorter than a clock cycle may be undetected.

- Schmitt Trigger



HW: Low Pass ; Built-in-noise cancellation
SW: Read signal twice or more

4. **Polling:** periodically check for event
- Waste of CPU Time if the event occurs infrequently
 - Polling sequence has to fit in the rest of the code
 - ↳ hard to modify or extend
- ⇒ → No precise timing is necessary
- * The state is important
 - * Impulses are long
 - * Signal is noisy (interrupts would be triggered very often)

Interrupts (IR)

: MCU polls the signal and interrupts the main program if a state change is detected. ⇒ **ISR** (interrupt service routine)

Interrupt Handling

- + Event occurs infrequently
- + Long intervals between events
- + Exact time of state change
- + Short impulse, polling might miss
- + Nothing else to do in main, could enter sleep mode

- * MCU monitors certain events (e.g. time overflow)
- * Event takes place → FLAG set (e.g. time overflow)
- * if all three bits are set ⇒ **ISR**
 1. Global Interrupt Enable Bit (I BIT)
 2. Individual Interrupt Enable Bit (e.g. time overflow enable bit)
 3. Interrupt Bit (time overflow FLAG)
- * Conflicts are resolved by priorities < static (ATmega) dynamic

ISR:

- Long delays
 - ↳ More to main routine
- More IRS
 - ↳ Order, complex timing
- Everywhere
 - ↳ disable it;

- ISR is triggered by event

* Save return add. (PC) to stack



* Clear I bit

* Clear interrupt flag bit (usually)



* Jump to corresponding interrupt vector table entry

SP → []

- Execute jump instruction at interrupt vector

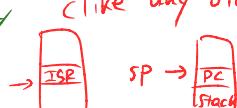


- Save additional context (anything not auto saved by HW)

SP → []

- Execute ISR body (like any other func)

- Restore context



- Leave ISR by assembly instruction RETI

* Return to PC popped from stack



* Set global interrupt enable bit (maybe delayed)

5. Timer, Counter : On-Chip Peripherals

↳ Counter

More than counting events and measuring time!

- * Input capture:
 - used to timestamp events (mostly external)
 - whenever the event occurs, timer auto copies its current count value to an input capture register
- * Output compare:
 - generate signals.
 - whenever a certain value is reached, the output compare event is triggered (can auto set or clear an output line)
- * Pulse Width Modulation
PWM
 - special case of output compare
 - Timer generates a periodic digital output signal with configurable high-time and period.

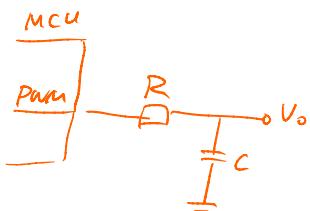
Watchdog

Special timer, monitor software execution.

- * enabled: counts down and resets the controller as soon as the count value zero is reached.
- * During SW execution the WD has to be reset to its initial value before it reaches zero. If fails the WD resets the controller.
- * Useful if program execution hangs and a restart solves the problem
- * Pathfinder problem

6. Analog I/O

* **RC LP:**



Simple & cheap, 1-pin, low quality, initially delayed, proportional to Pwm (high-time/period)

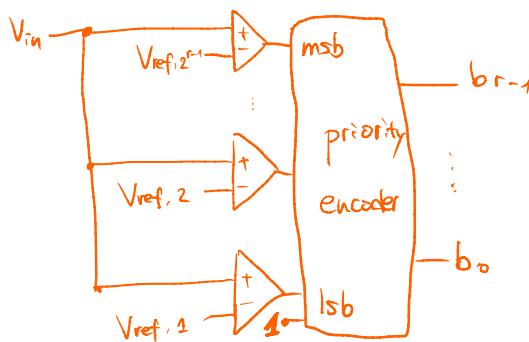
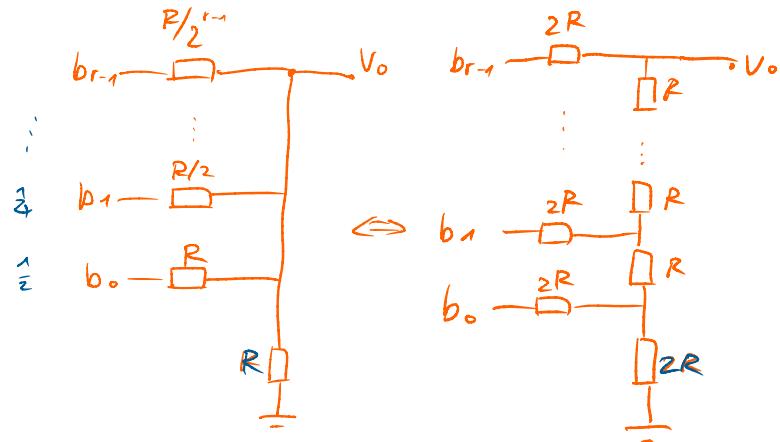
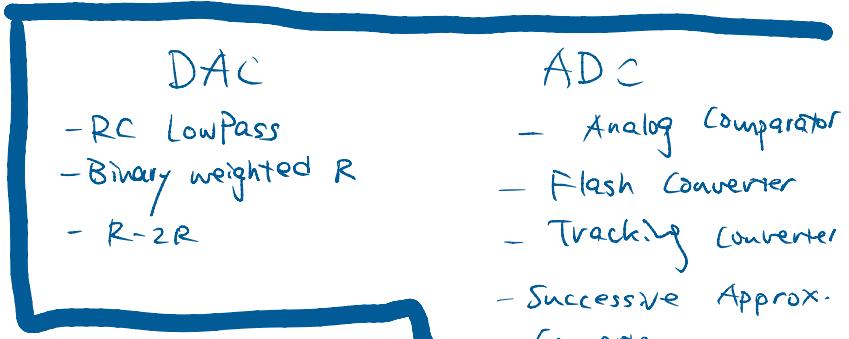
* **Binary Weighted Resistor Circuit**

↳ - high precision resistors!

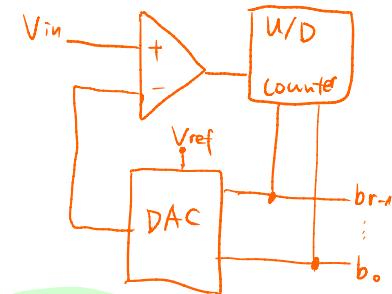
$$V_o = V_{ref} \cdot \sum_{i=1}^n \frac{1}{2^i} b_{i-1}$$

* **R-2R Ladder**

↳ precise, **Many Resistors**
Less Loss



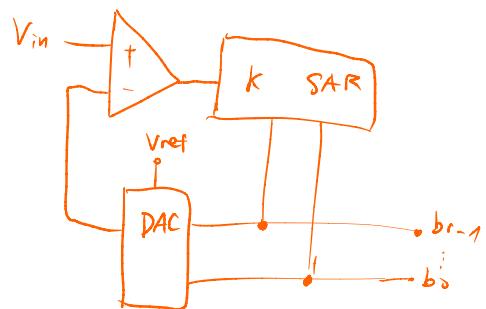
Flash Converter: fast!
simultaneous check
 2^{l-1} comparators needed → expensive



Tracking Converter (DAC to ADC)

Counter changes linearly according to outcome of comp. Counter holds digital estimate of value. Oscillation

$O(2^l)$



Successive Approximation Converter

SAR: exponential steps
start with MSB

$O(\log r)$

Ramp-Compare Converter:

- Saw signal compares measured signal
- ramp voltage reached → **comparator fires**
- Time diff → value

Ramp signal maybe reused for additional conversion.

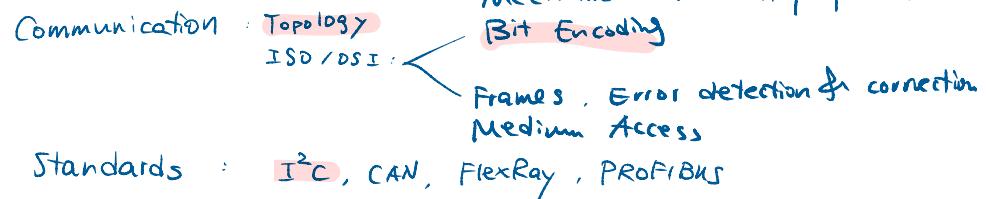
Offset Error: constant added, step size same → **Built-in offset correction**

Gain Error: step size constant added, gradient diverges → **Built-in gain adjustment**

Differential Non-Linearity: Non-constant step change → **worst case deviation**

Aliasing: Nyquist criterion: sampling freq > 2 signal freq. → **Anti aliasing filters: LP**

§3 Databus



Fourier: $f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\omega t) + \sum_{n=1}^{\infty} b_n \sin(n\omega t)$

Damping: higher frequencies are damped faster

Bus Topology

- * linear line (with terminators)
- * **passive** connection (no repeating)
- * Only one partner can send at a time
- * All partners can listen to all communication
- + cheap, simple
- Multiple access (Babbling idiot, security)
- single point of failure

Star Topology

- * Dedicated connection to central station
- * Buffering & repeating
- * Multiple partners can send at a time
- * Only sender & receiver can listen
- * + No multiple access.
+ only central station is single point failure
- Expensive central station
- More wiring

Ring Topology

- * Circular line
- * Active connection (repeating) changing
- * Multiple partners can send at a time
- * Some partners can listen (passive)
- + high quality of service
- + No multiple access
- Complex (expensive)
- Single point of failure

L1: Physical

- Defines mechanical properties: medium, connectors
- Defines electrical / optical properties: voltage, frequency, baudrate
- HW

Bit encoding

1. Return to Zero (RZ):  RZ after each pulse: 3 states. **Self synchronizing**

2. Non Return to Zero (NRZ):  2 states, **needs synchronizing**. Full data rate, capacitive problems

3. Differential NRZ: 0: level change

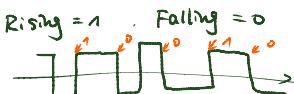


; 1: no change: No problems for long "0"

4. Bit Stuffing: Prevents long "1". Sender inserts a "0" after sequence of n "1" (e.g. USB: n=6)

Receiver checks and removes inserted 0's

5. Manchester Code: No neutral, self synchronizing.



No capacitive problems. half data rate

* Synchronization

- Clock never perfect

- A: 100µs, B: 90µs

- A: 9 ticks = B: 10 ticks

6. 4B/5B: Uses some form of NRZ

80% data rate.

Encoding table prevents long sequences

L2 : Data Link

Sublayers ↗ Logical Link Control
Media Access Control

- Encapsulate data (bits) into frames
- frame synchronization
- Logical link control :
- Media access control
- HW : Switch, bridge

Automatic Repeat Request ARQ
Forward Error Correction FEC
Flow Control

LLC :

Frames : Data → Frames :

SD	SA	RA	ID	Ack	L	DATA	CS	ED
Start Delimiter	Sender Add.	Receiver Add.	Acknowledgement	Length of frame/ data	Checksum	End Delimiter		

Error Detection : Parity bit : One bit error detection

Cyclic Redundancy Check (CRC) - Hash func. Detection of burst errors

Hamming Code : Set of parity bits ; single EC, double ED

ARQ : ACK : successful / NACK : error / - : lost

MAC : Regulates access to a shared medium

Static MAC :	Frequency	Time	Code	Hybrid
	FDMA	TDMA	CDMA	

Dynamic MAC :

CSMA/CD Carrier sense multiple access / collision detection.

- * Wait until medium is free.
- ↗ Start sending
- + If collision:
 - Scramble
 - back off

High data rate /
long range

CSMA/CR

Resolution

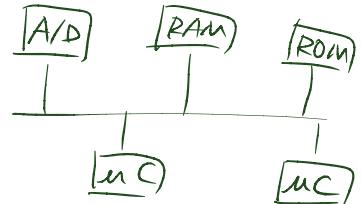
- * Wait until medium is free.
- + Start sending
- + Collision
 - Stop sending
 - Start receiving

Dominant : 1 wired or
0 wired and

Data Bus Standards : I²C CAN FlexRay PROFIBUS

1. Inter - Integrated Circuit Bus : (Two Wire Interface TWI)

- Connects multiple devices on the same board
- Modes with different Speed 100kbit ~ 5Mbit/s
- Noise-prone (used inside shielded casings)
- Simple & cheap & popular



Physical Layer

- Two lines connected to pull-up resistors:

SCL : serial clock line SDA : serial data line

- Devices are connected via open connectors
- Hi (1) : 3.3 - 5 V Lo (0) : -0.5 - 0.3 V
- Wired AND : dominant 0

Data Link Layer

- Each device has a unique 7 bit address (priority)
- Simple frame structure
- Master, Slave : Master polls / pushes data
- Multiple master : arbitration by CSMA/CR : First 0 wins
- No ED / EC
- Flow Control by ACK, Clock stretching



ACK: 10 NACK: hi in SDA

2. Controller Area Network (CAN)

5km : 10kbit/s

Automotive Bus, Also industrial (CANopen)

25m : 1 Mbit/s

Physical Layer

- Comfort Bus (low speed)
- NRZ with bit stuffing after 5 equal bits
- Wired-AND (dominant 0)

Data Link Layer

- Each message type has ID (priority)
- Device : no address
- 4 frame types
- Arbitration by CSMA/CR First 0 wins
- CRC (15bit) ED

Object Identifier

- Unique bit mask (11 bits or 29 bits)
- Each object ID should only be sent by one device
- Each device can have multiple object IDs
- Object ID is used for arbitration
- Assignment of object IDs is fixed in specs
 - CAN Matrix docu
 - Reserve object IDs for future extensions

Matrix & Frames ...

- CAN open:
1. Request: application requests service
 2. Indication: system notifies application of event
 3. Response: application replies to an indication
 4. Confirmation: system confirms service execution

Flex Ray: Automotive network communication protocol

Deterministic timing: Real-time capable

Hybrid MAC: TDMA + dynamic timing

- 2 channels: 10mbits each. for redundancy or single: higher datarate
- Distributed clock synchronization (no master)
- flexible topology

Physical Layer

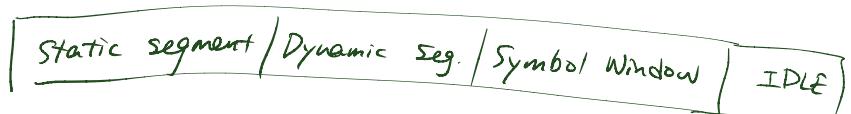
- * NRZ
- * shielded twisted pair

Data Link Layer

Repeating
communication
cycle

* Static: Realtime (CTDMA)

* Dynamic: other (FTDMA)



PROFIBUS : Process Field Bus : automation in industry

Physical

Shielded Twisted Pair , Bus , NRZ
Optical fiber , flexible topology

Data Link

7 bit add. 5 frame types. Master polls /pushes data

CRC (8bit) ED , Multi-Master by token passing

ED/EC Hamming Code.

$$(n, k) : n = 2^r - 1, k = 2^r - r - 1$$

n: block length

$$\text{Rate} : 1 - \frac{r}{2^r - 1}$$

k: message length.

r: parity

even-parity :

set parity is even

d_{\min} : Hamming distance. ($r \geq 1$)

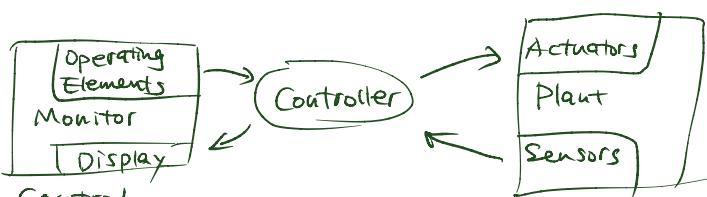
Detect : $d_{\min} - 1$

Correct : $\lfloor \frac{d_{\min} - 1}{2} \rfloor$

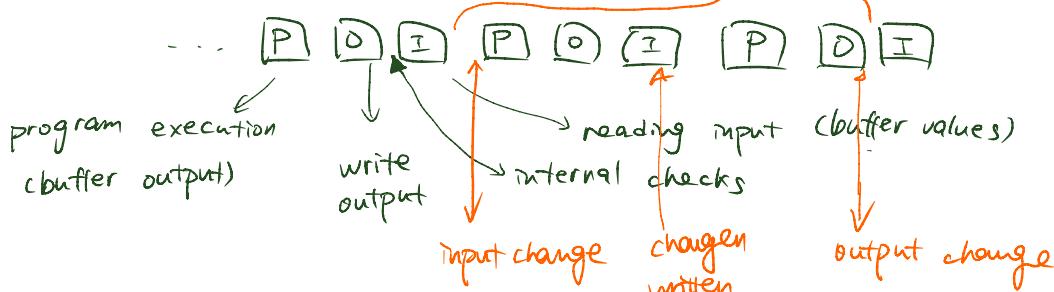
§ 4 Programmable Logic Controllers (PLC)

1. Intro & Tech

- Logic control vs continuous control



- PLC OS: Cyclic Scanning Mode: worst case: $2 \times$ cyclic time delay

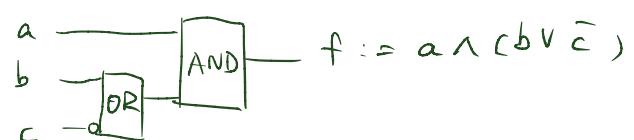


Supports estimation of Real-time behavior:

- Determine longest program execution path
- Measure or estimate cycle time for this execution path.
- Apply $2 \times$
- Reaction time too long? : Optimize program; more performant PLC; interrupt.

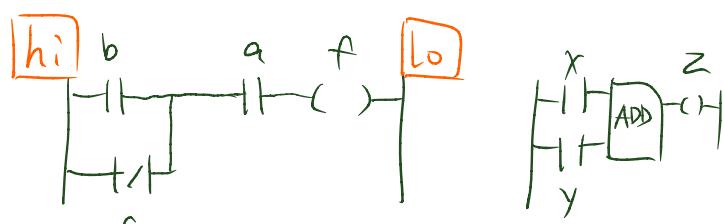
2. PLC Programming Language

• Function block (FB)



Not restricted to logic : NAND, ADD, MULT, TIMER...

• Ladder Diagram (LD)



contact: $\text{---} \text{---}$ True \rightarrow Closed

$\text{---} / \text{---}$ False \rightarrow Open

coil : $\text{---} () -$ Current \rightarrow True

$\text{---} (/) -$ Current \rightarrow False

{ Parallel: OR
Serial: AND }

Instruction List (IL) ~ Assembly

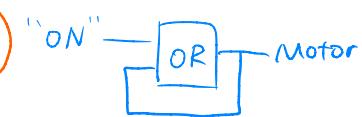
Accumulator ($CCR = \text{Current Result}$)

LD	b
ORN	c
AND	a
ST	f

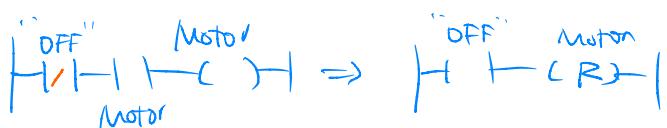
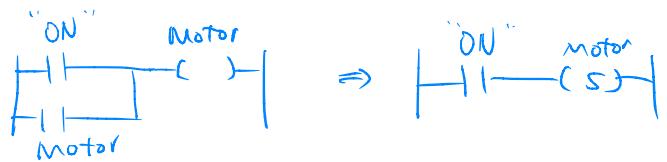
$$f = a \wedge (b \vee \neg c)$$

2.2 Standard Function Blocks

1 Latches :

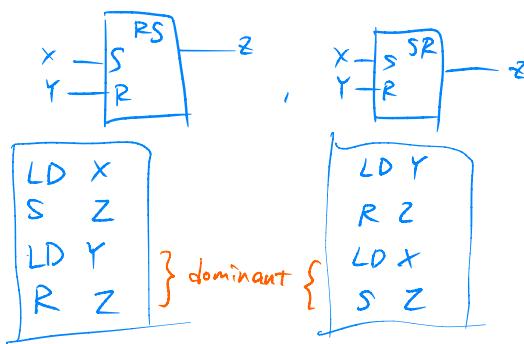
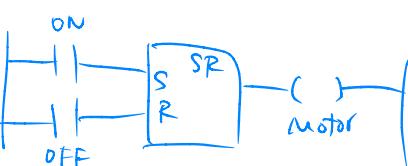
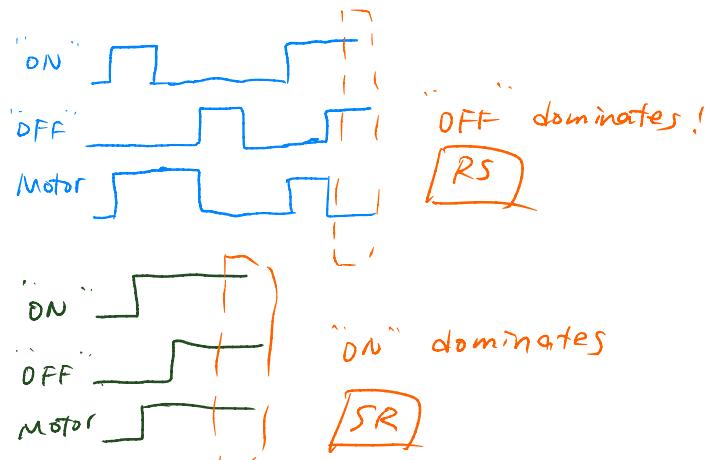
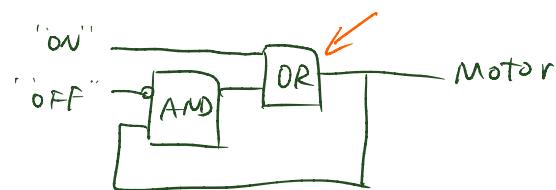
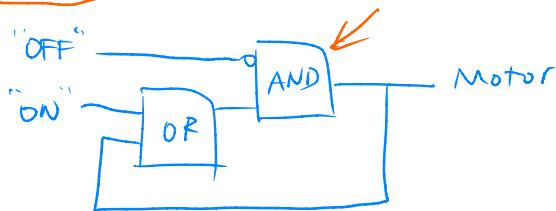


i)



2 Bistables :

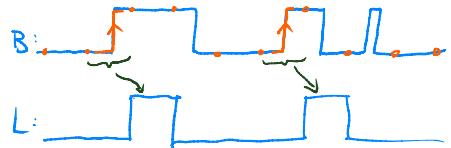
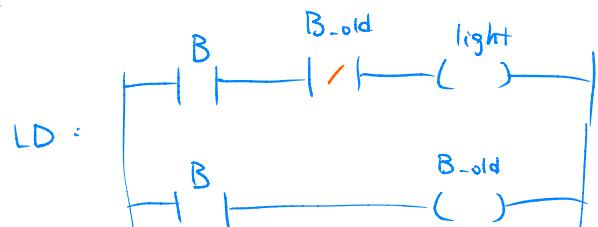
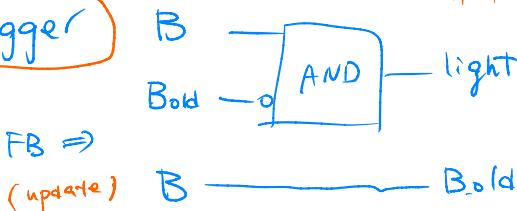
Combination of both latches ..



B	old	L
0	0	0
0	1	0
1	0	1
1	1	0

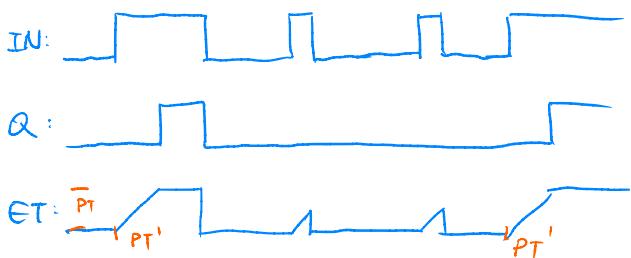
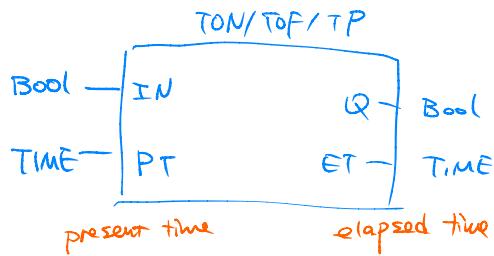
3. Edge Trigger

Task: Press the 'B', Light on for 1 cycle.

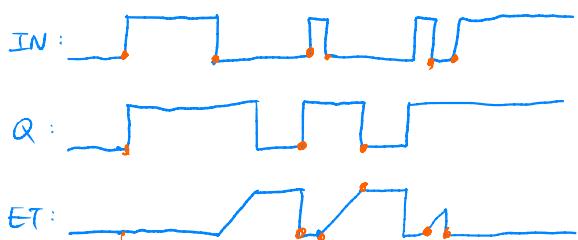


4. Timers

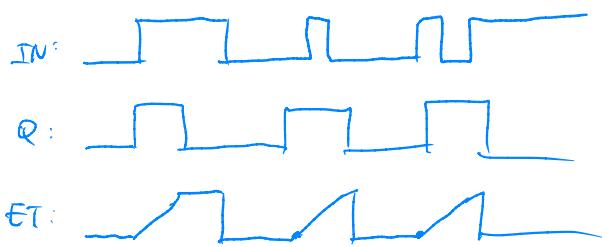
{ TON : Timer On Delay
 TOF : Timer Off Delay
 TP : Timer Pulse



TON: IN : hi \Rightarrow ET starts
 IN : lo \Rightarrow ET stops
 $ET = PT \Rightarrow Q : hi$



TOF: IN : lo \Rightarrow ET starts
 IN : hi \Rightarrow ET stops
 $ET = PT \Rightarrow Q : lo$



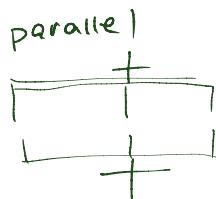
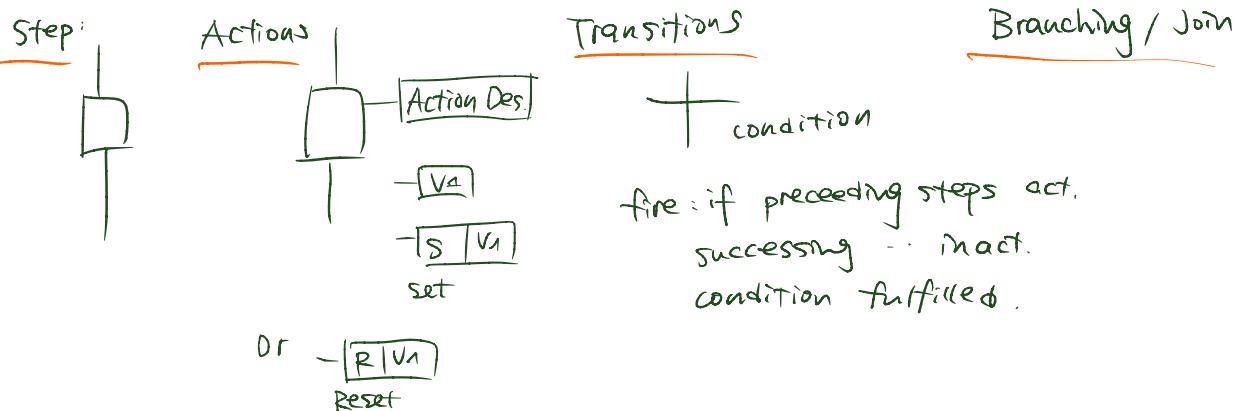
TP: IN : hi \Rightarrow ET starts
 wait 1 cycle
 IN : lo \Rightarrow ET stops
 $ET \nearrow : Q : hi$

3. Model-based development of Logic Control Programs

Automata:
 Moore : Output depends on State
 Mealy : Output depends on State & Input

Alarm Handling : danger —?— red
 receipt —?— yellow
 reset —?— yellow

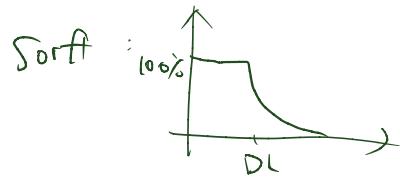
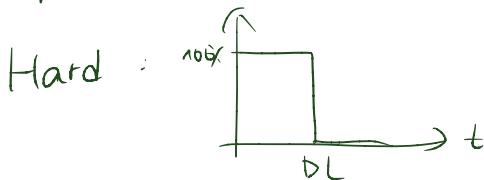
4. Sequential Function Charts (SFC)



§5 Real-Time

Computation is correct and finished in time.

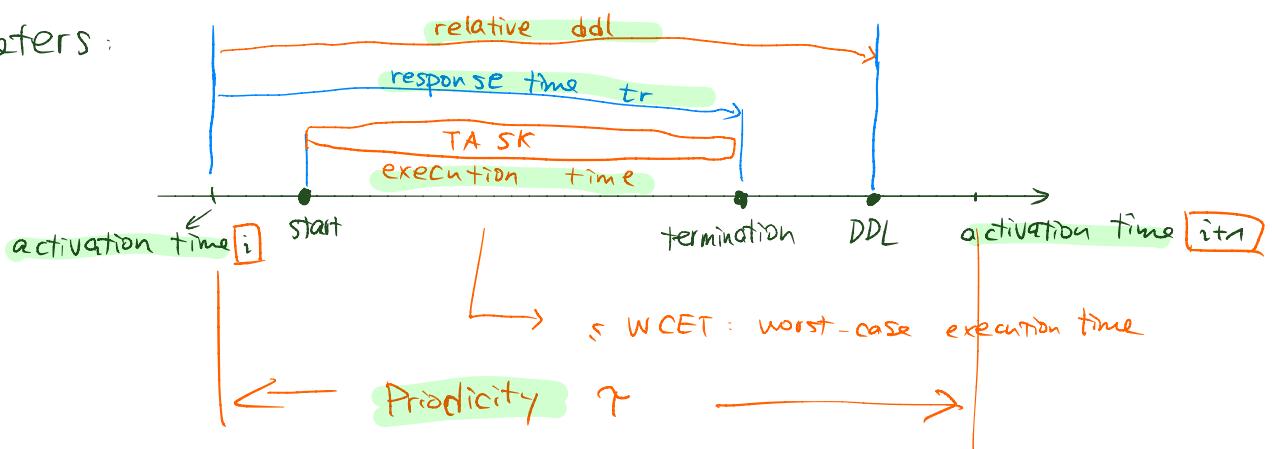
1. Requirements



stably a sufficient number of reactions in time.

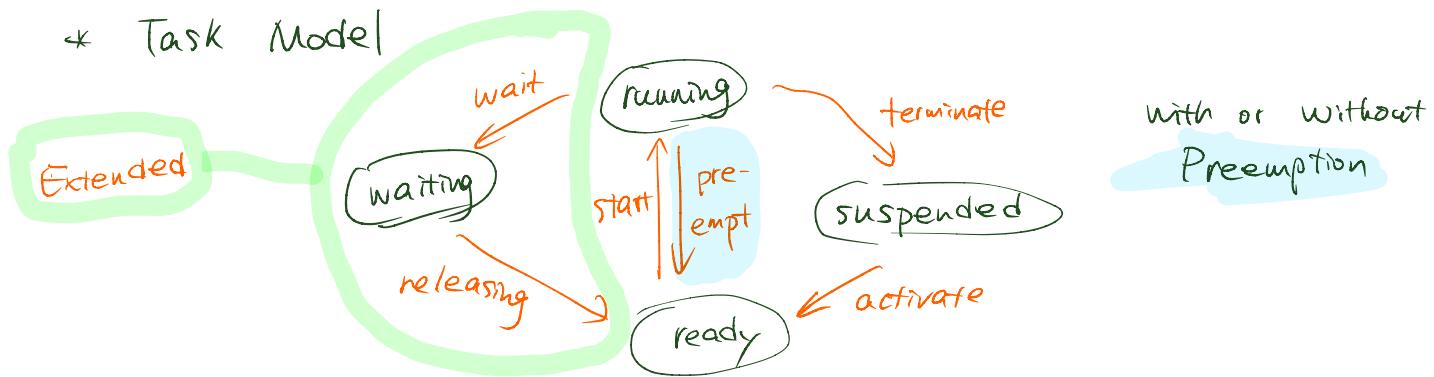
How to fulfill: Polling ; Main loop & Interrupts ; Realtime OS

Parameters:



2. DSEK: Offene Systeme und deren Schaltstellen für Elektronik in Kraftfahrzeugen

* Task Model



3. Scheduling, Priority

- * Preemptive:
 - + Tasks can be interrupted by tasks with higher priority and be finished afterwards (Starving)
 - Swapping the context (register content...) takes time!

Cooperative: Each task will be executed in its duration.
No Interrupts.

Schedule: Mapping of an execution sequence to a task sys.

Feasible: if no deadline is violated

Schedulable: Feasible Schedule exists

$$T = (\tau, T, D) \quad \begin{matrix} \text{period,} \\ \text{(repeating freq)} \end{matrix} \quad \begin{matrix} \text{task time, DDL} \end{matrix}$$

CPU utilization $U := \sum \frac{T}{\tau}$

$$T_1 = (3, 2, 3), T_2 = (4, 2, 4)$$

$$U = \frac{2}{3} + \frac{2}{4} > 1 : \text{NOT schedulable}$$

Earliest Deadline First (EDF) (early First)

Rate Monotonic Scheduling (RMS) Sort by $\frac{1}{\tau}$ (short First)

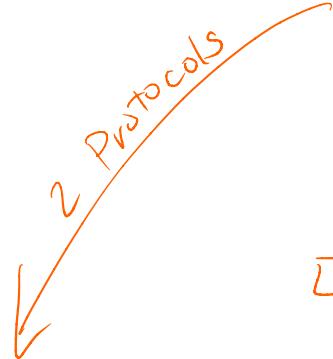
Deadlock:

A occupies R₁.

B occupies R₂.

B requires R₁

A requires R₂



Priority Inversion:

A > B > C.

C occupies R

A starts (preemptive)

- A requires R
B starts & terminates (< Problem)
C continues, terminates
A occupies R, terminate

Priority Inheritance Protocol:

* A (hi prio) needs Resource, B (lo prio) holds R.

* ASA A starts waiting for R:

Action: B inherits A's priority ①

Effect 1: B cannot be preempted by an intermediate priority

Effect 2: No priority inversion possible

* Complex: Current holder of R must be determined
Holder's prio must be changable while running.

Priority Ceiling Protocol

Task → ceiling (top) prio if it holds R

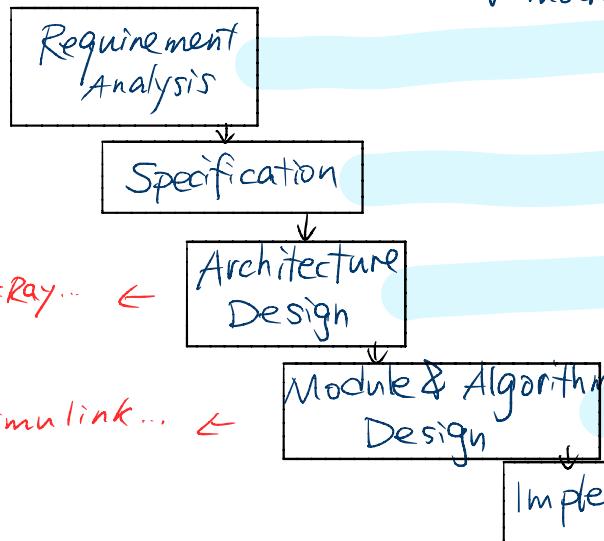
can be calculated during design time. Not optimal!

§ 6. Embedded Software Development & Design

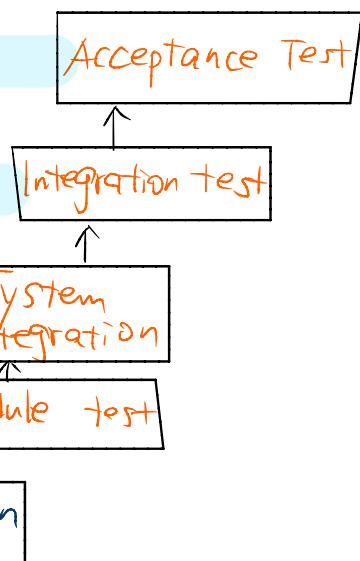
what
Functional / non-functional reqs
how good

Systematic: CAN/FlexRay...

Simulink...



V model:

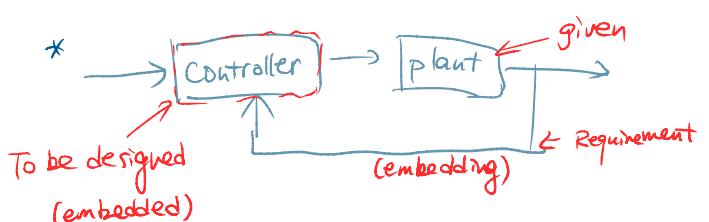


Requirement Engineering:

- * Elicitation (collect)
- * Analysis
- * Management

solution requirement + constraint

* checkable, understandable



Technically Oriented

functional Regs

Analysis of functional Regs

functional specs

= optimisation constraints

Requirement Elicitation

Requirement Analysis

Architecture Design

Business oriented

Non-functional Regs

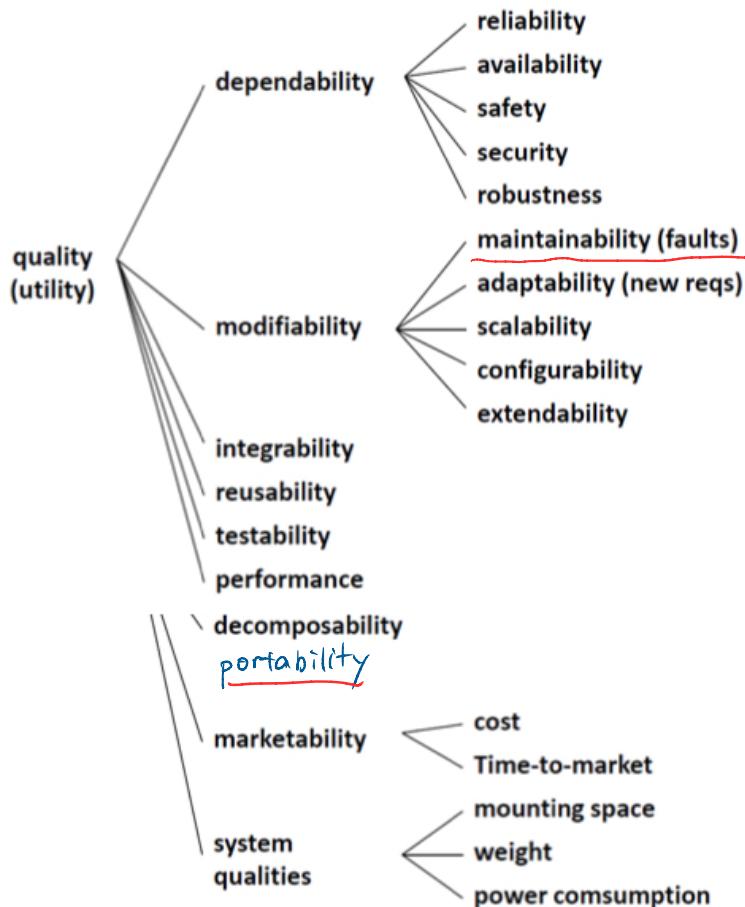
Analysis of Qualities

driving qualities

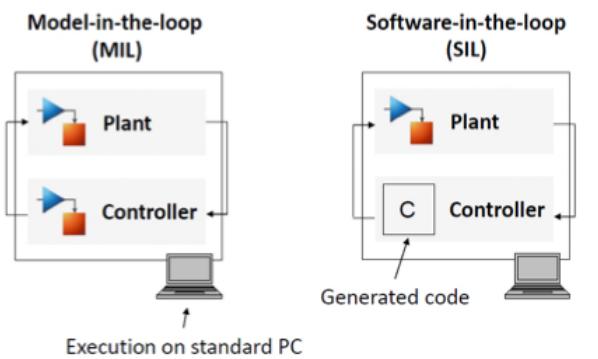
= optimisation criteria

Optimisation

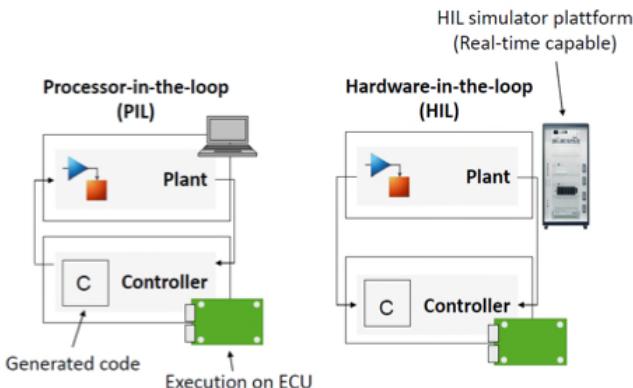
Quality Tree



MIL and SIL Tests



PIL and HIL Tests



Architecture

The architecture of a system is the structure or the structures of the system, which comprise elements, the externally visible properties of those elements, and the relationship among them."

- Defines constraints on implementation
- dictates organisational structure
- inhibits or enables a system's quality

Information hiding

Similar to the idea of object-oriented programming languages like C#.

1. Identify design decisions which are likely to change (e.g. use polar coordinates instead of Cartesian coordinates).
2. Assign each of these decisions to one module, if possible.
3. Encapsulation: For each module, hide the design decision behind an abstract interface such that the other modules do not have to know about it (e.g. implement method "GetCoordinates" that returns Cartesian coordinates) → "Secret of the module"

Information hiding supports:

- Modifiability
- Maintainability
- Reusability

Information hiding does not support:

- Performance (e.g. memory efficiency)