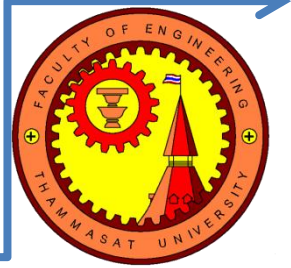
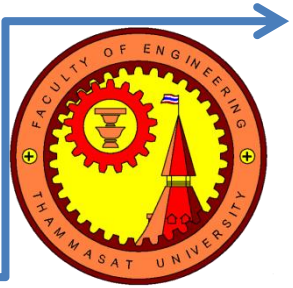
MBD workshop

Asst.Prof.Dr.Supachai Vorapojpisut
Faculty of Engineering
Thammasat University



AUTOMOTIVE EMBEDDED SYSTEMS

Automotive technologies

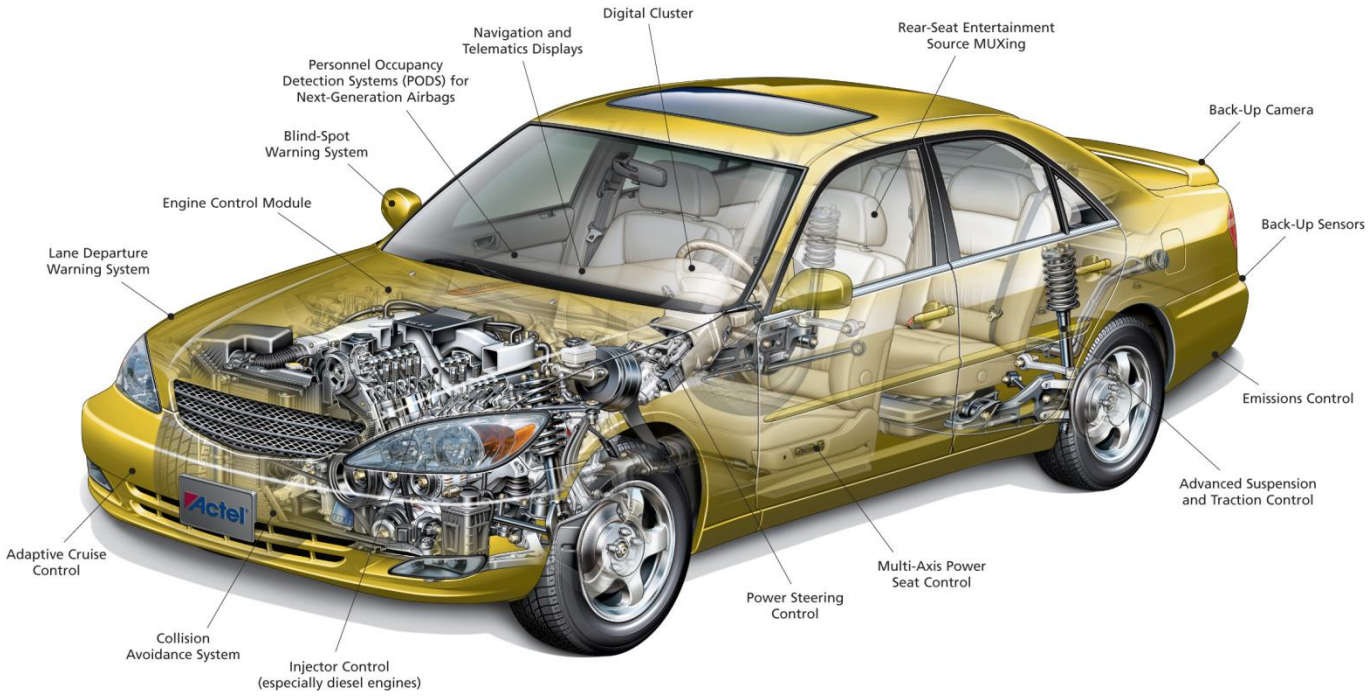


	Toyota Corona	Toyota Altis	Toyota Prius	Tesla Model S
				
energy	carburetor	EFI		motor
energy	distributor	VVT-i		
environment		hybrid engine		
environment		catalytic converter		
safety		ABS, safety airbag, …		
safety			automotive radar	
safety				self-driving

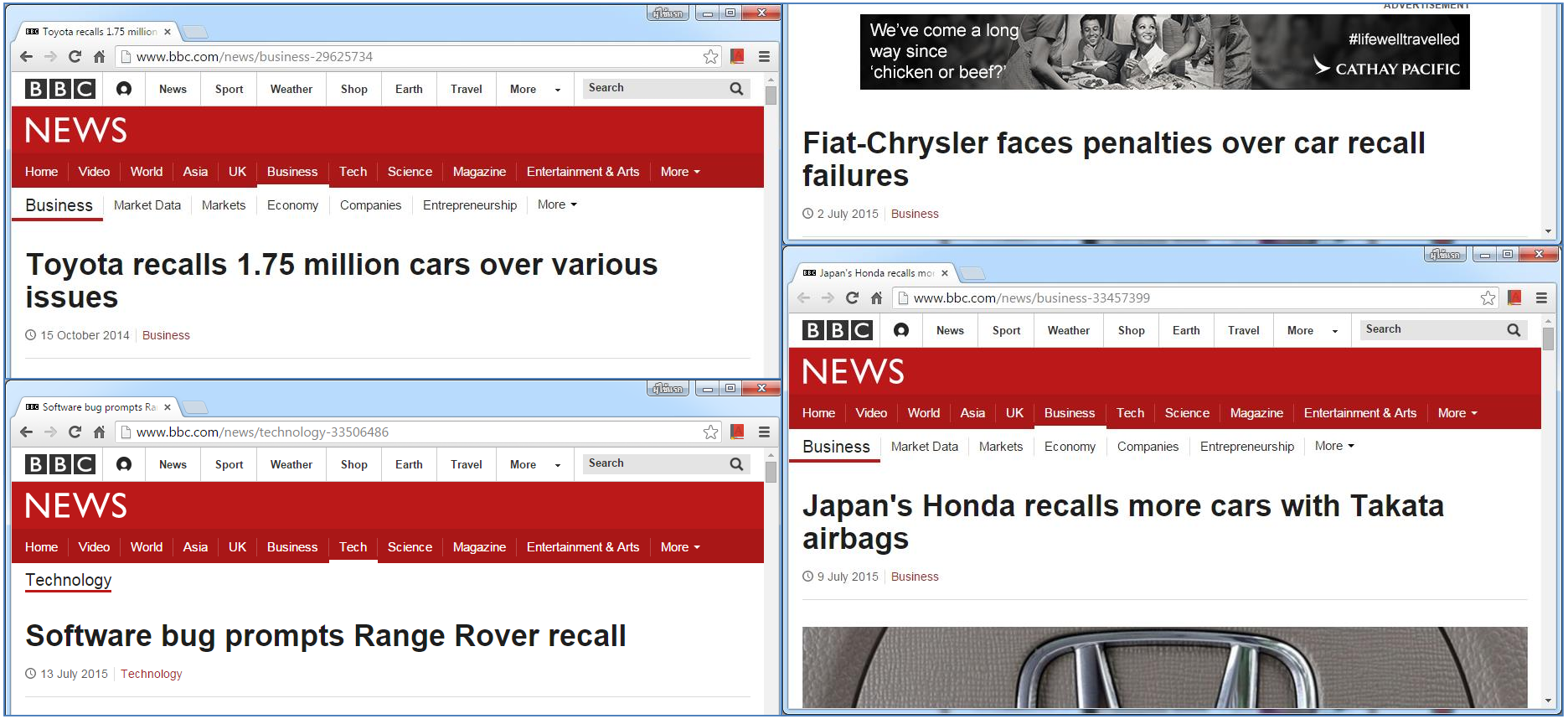
Automotive software

IEEE Spectrum

Recent premium-class automobile probably contains close to 100 million lines of software code.



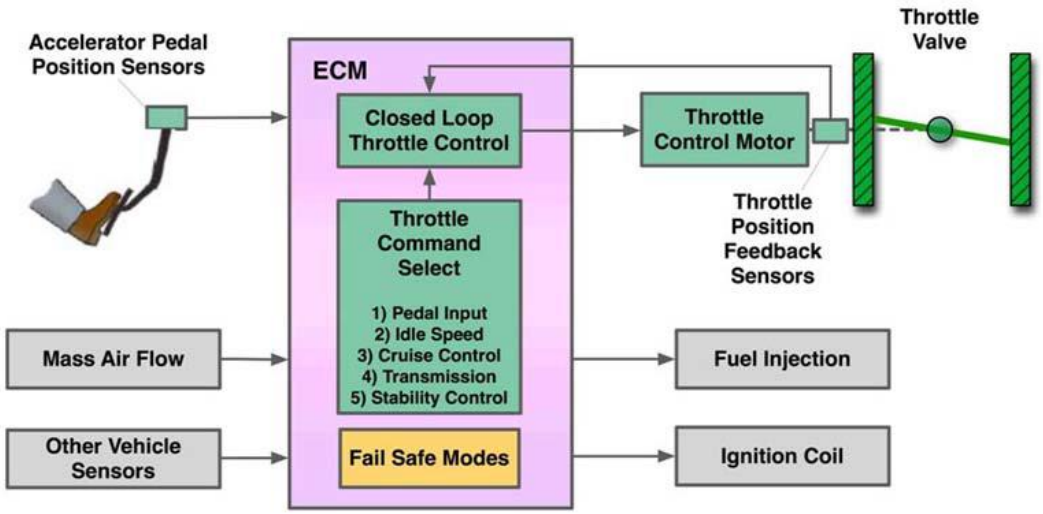
Automotive software challenges

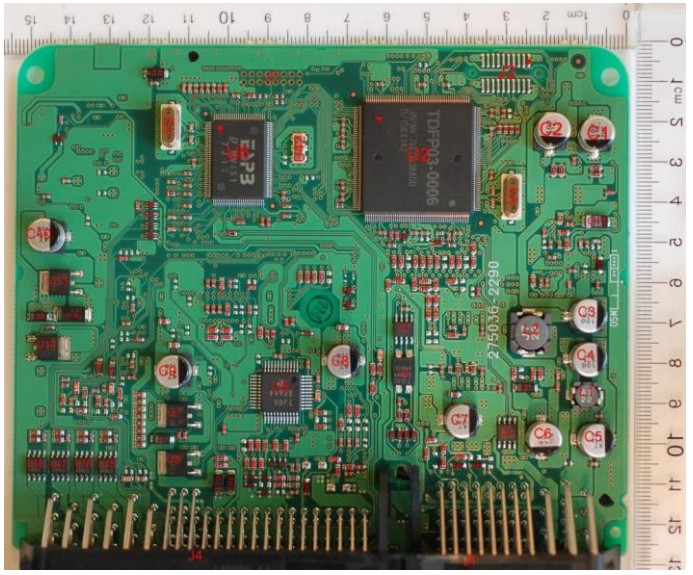


Embedded systems

Wikipedia

A computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints.





Automotive embedded systems

Engine Control Unit

crank angle sensor, air flow sensor, O2 sensor, …
throttle position sensor, pedal position sensor, …

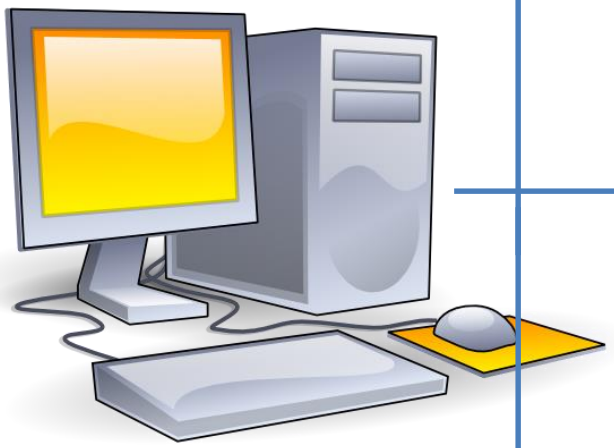
motor driver signal, fuel injector driver signal, ignition signal, solenoid signal, …

OBD-II, LIN, Flexray, …

Nature	Timing	Constraint	Reliability

Embedded software development

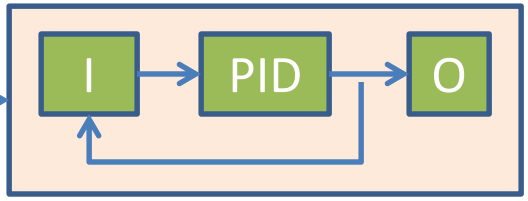
Host computer



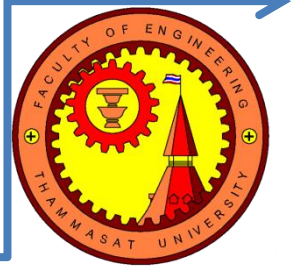
```
#include <stm32.h>
int main() {
  platform_init();
  while(1) { ... }
}
```

```
#include <stdio,h>
int main() {
  printf("hello");
}
```

Linux OS



MATLAB/Simulink

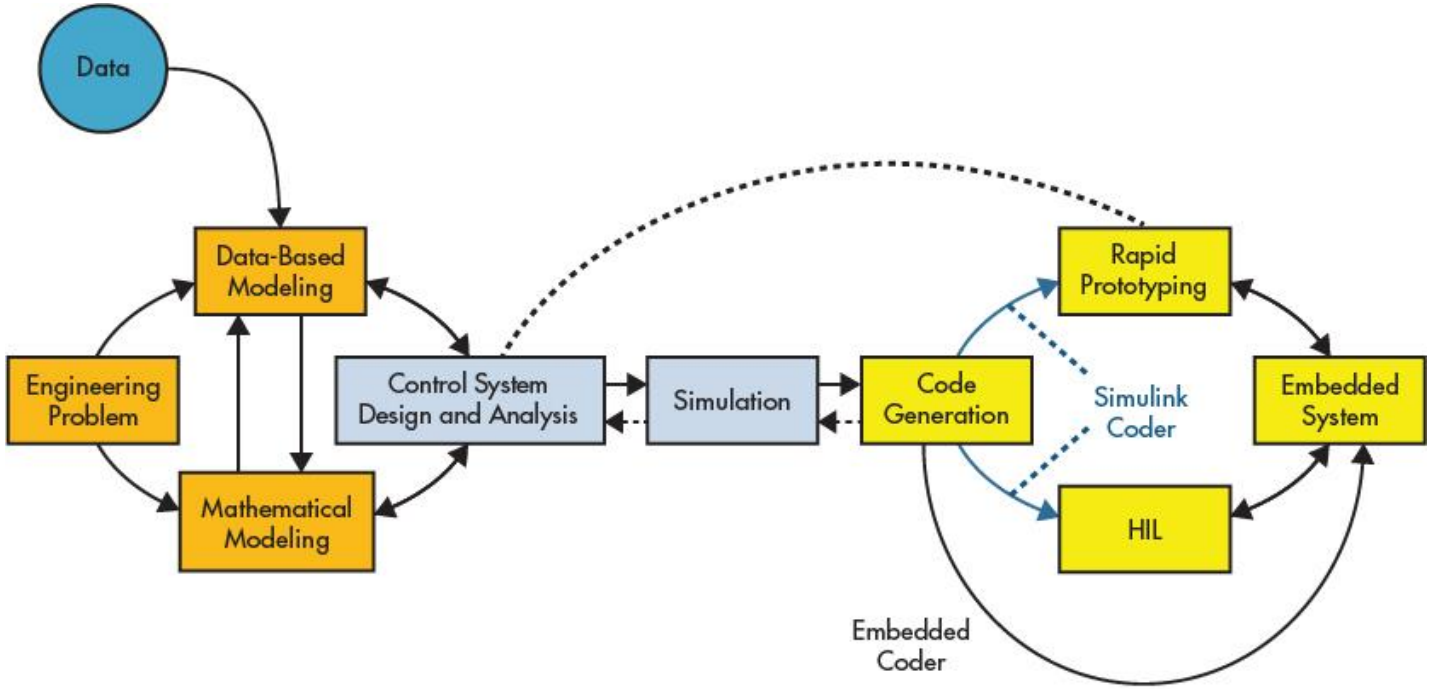


MBD CONCEPTS

Model-based development

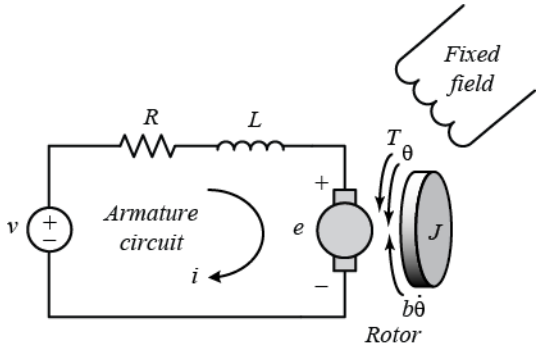
Mathworks

A system model is at the center of the development process, from requirements development, through design, implementation, and testing.

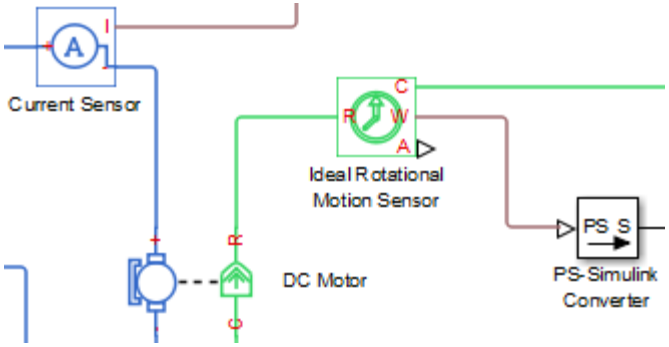


Modeling

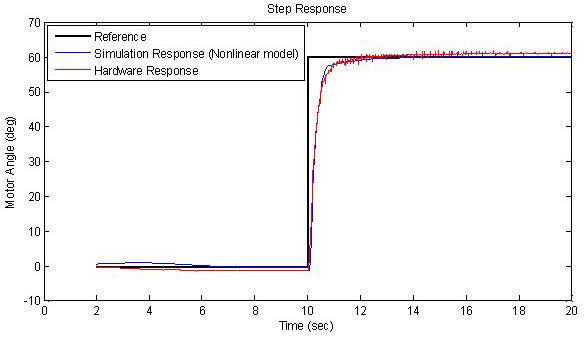
First principles modeling



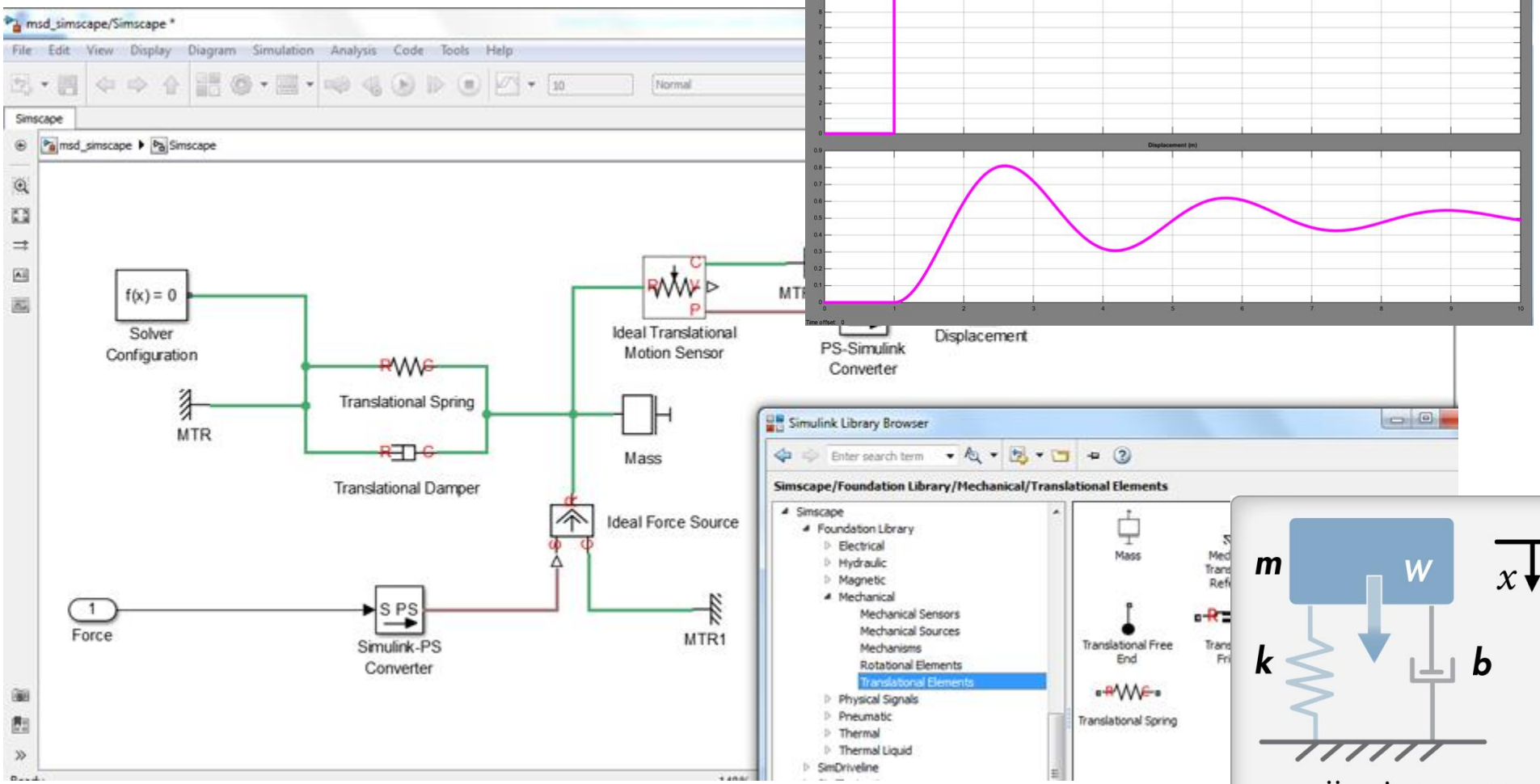
Physical modeling



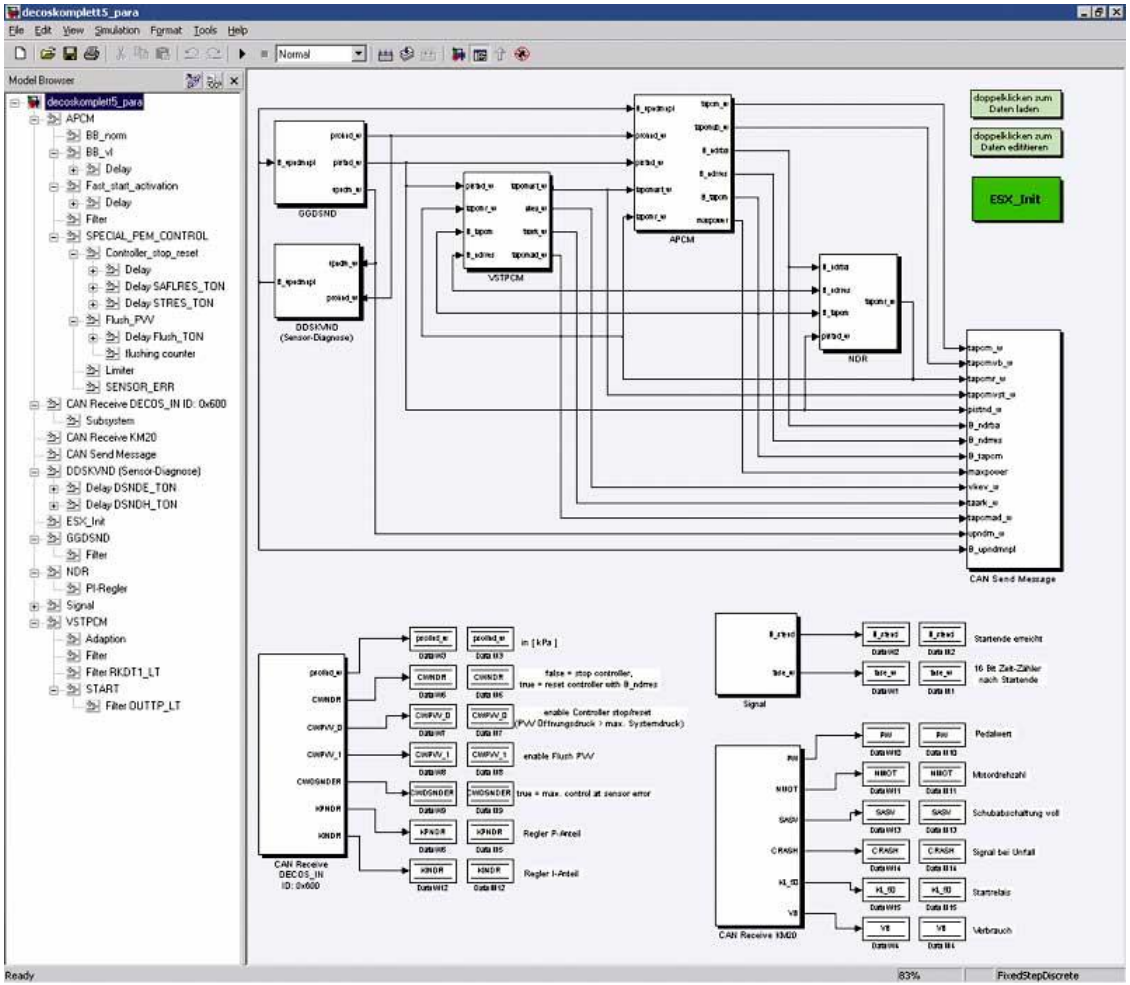
Data-driven modeling

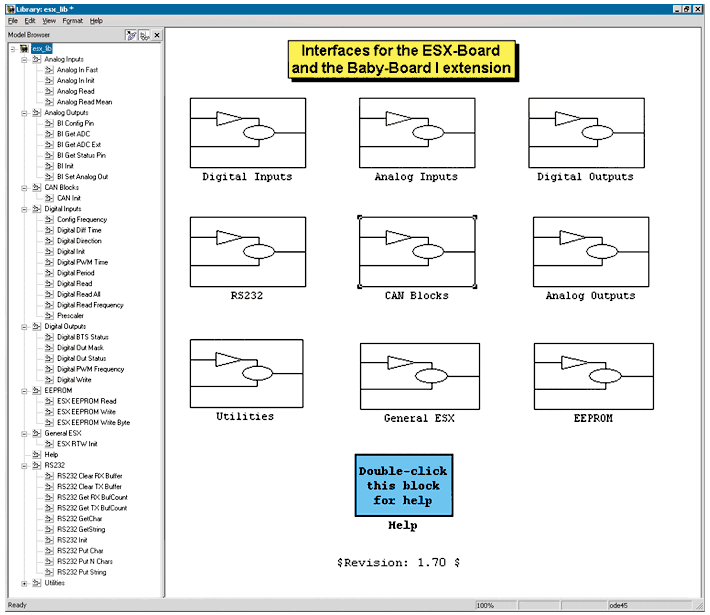


Simulation

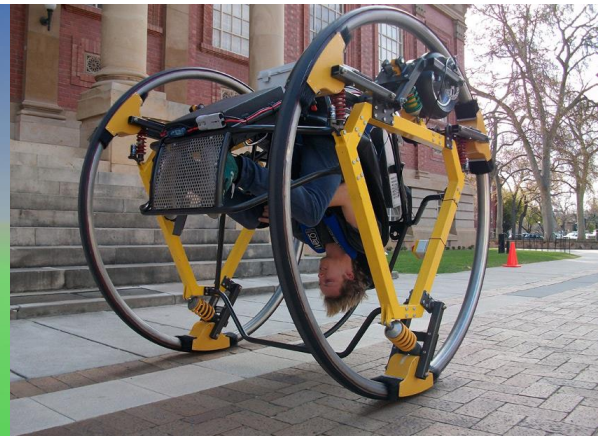

msd_simscape/Simscape *
File Edit View Display Diagram Simulation Analysis Code Tools Help
Simscape
msd_simscape ▸ Simscape
f(x) = 0
Solver Configuration
MTR
Translational Spring
Translational Damper
Ideal Translational Motion Sensor
Mass
Ideal Force Source
1
Force
S PS
Simulink-PS Converter
MTR1
PS-Simulink Converter
Displacement
Force(N)
Displacement (m)
Simulink Library Browser
Simscape/Foundation Library/Mechanical/Translational Elements
Simscape
Foundation Library
Electrical
Hydraulic
Magnetic
Mechanical
Mechanical Sensors
Mechanical Sources
Mechanisms
Rotational Elements
Translational Elements
Physical Signals
Pneumatic
Thermal
Thermal Liquid
SimDriveline
Mass
Translational Free End
Translational Spring
m
w
x
k
b
$m\ddot{x}+b\dot{x}+kx=W$

Code generation

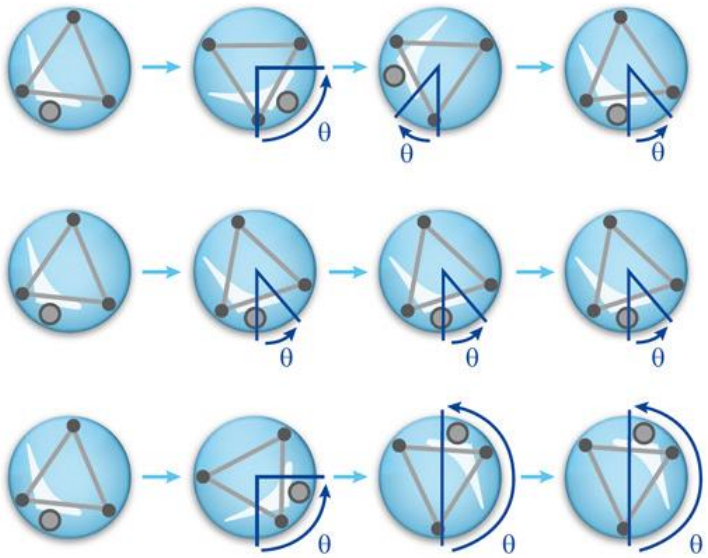




Case study: diwheel



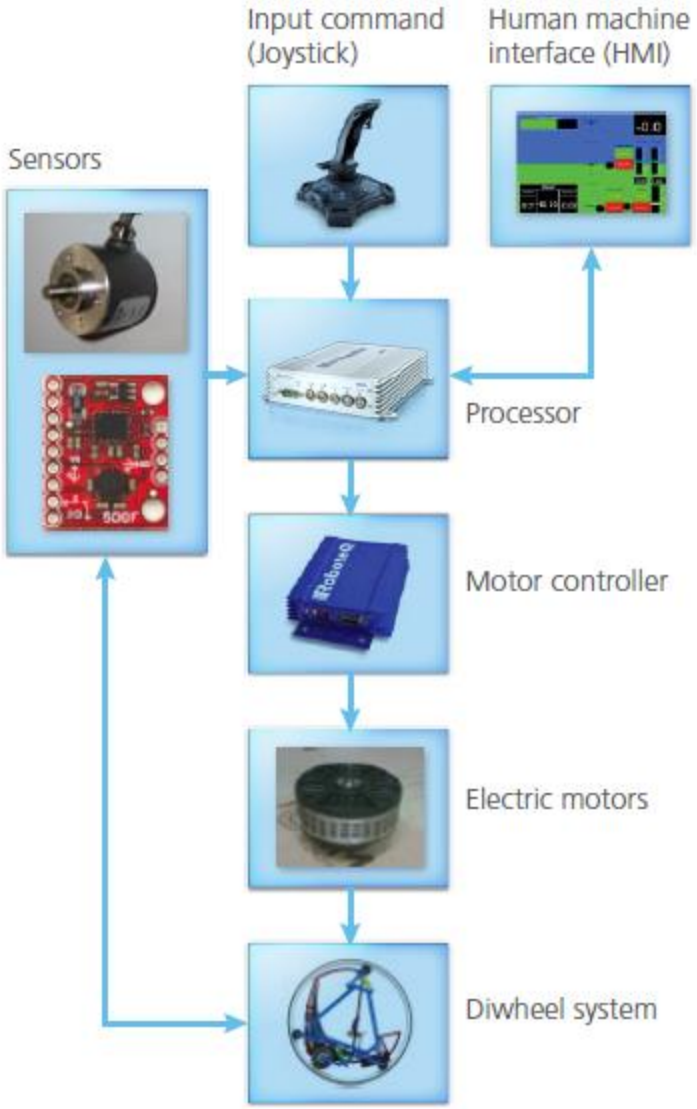
Control Modes

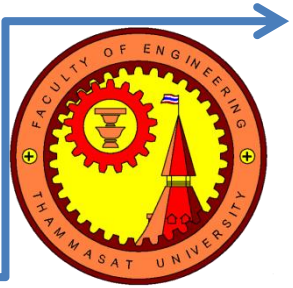


Open Loop
- No feedback from sensors
- Experiences 'slosh' and is hard to control

Slosh Control
- Uses feedback from the IMU sensor measuring the inner frame angle
- Keeps the inner frame in a stable position using a proportional-derivative controller

Inversion Control
- Uses feedback from the IMU and encoder sensors
- Swings the inner frame to the inverted position
- Balances the inner frame upside down using a full-state feedback regulator





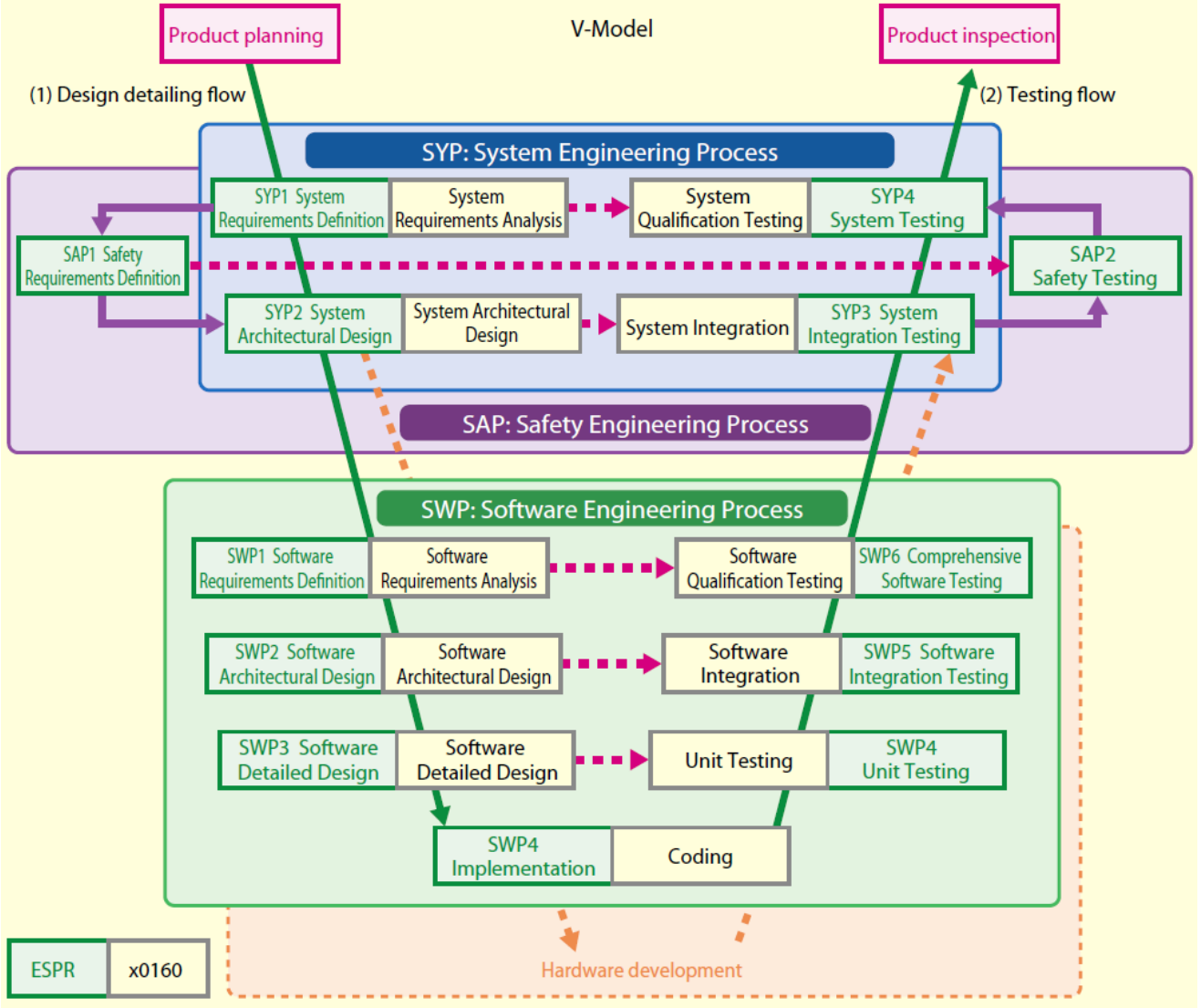
MBD PROCESS AND TOOLS

V-model

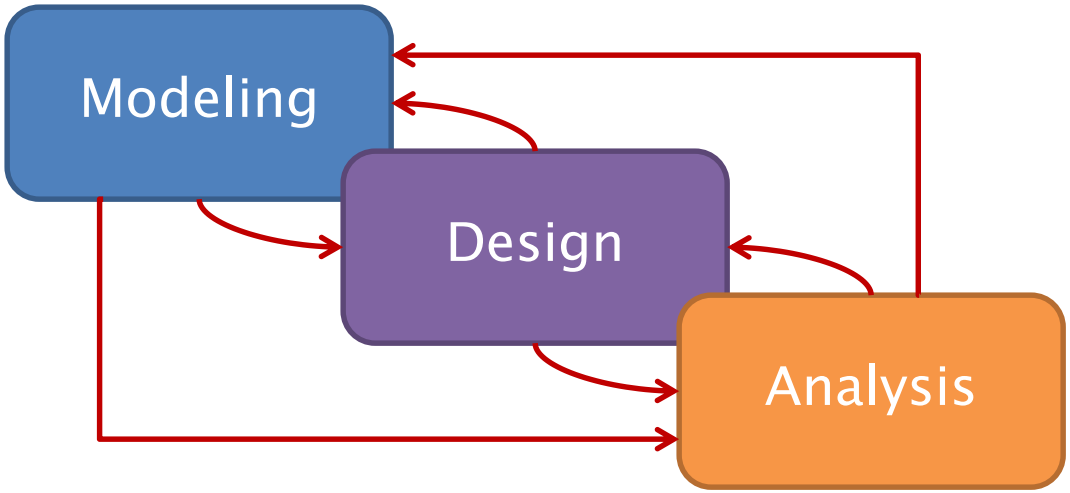
System Engineering

Safety Engineering

Software Engineering



MBD tools



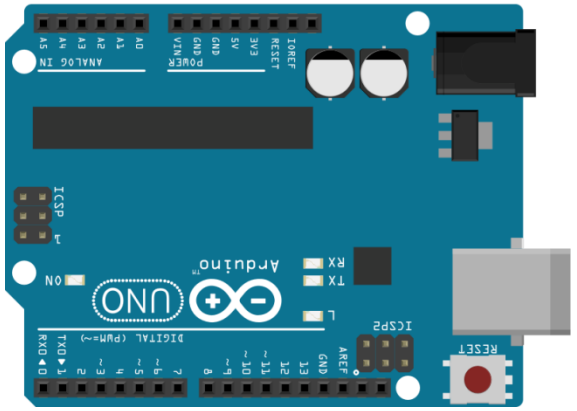
Embedded Coder

Stateflow | SimScape

Simulink

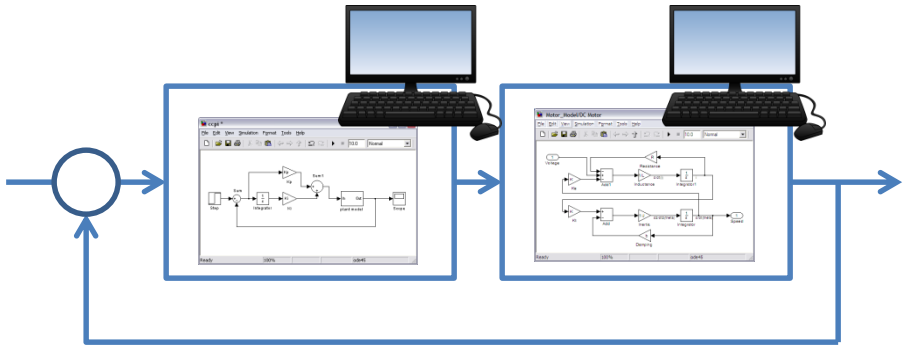
MATLAB

C code

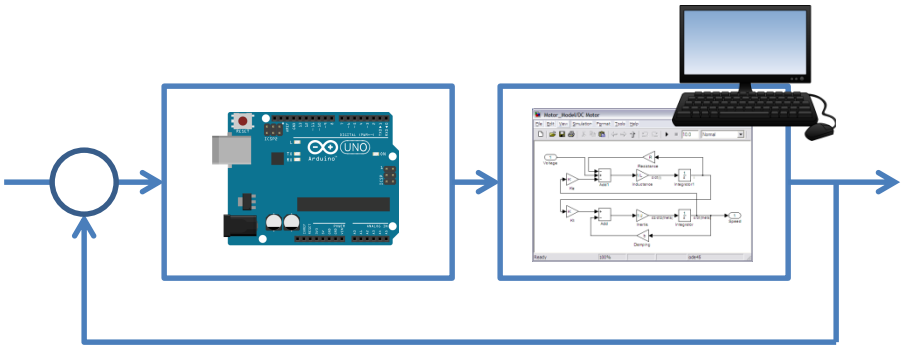


Cross toolchain

Simulation techniques

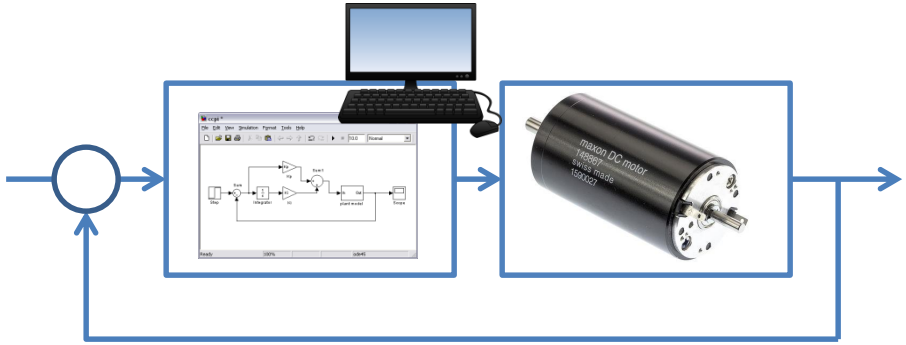


Model/Software-In-Loop Simulation

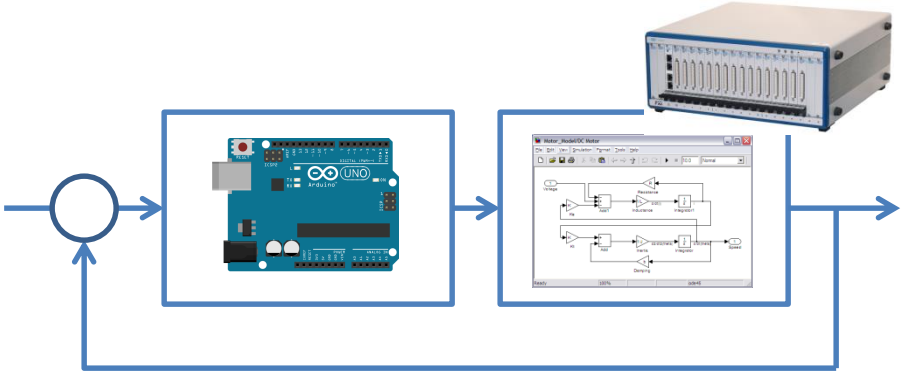


Processor-In-Loop Simulation

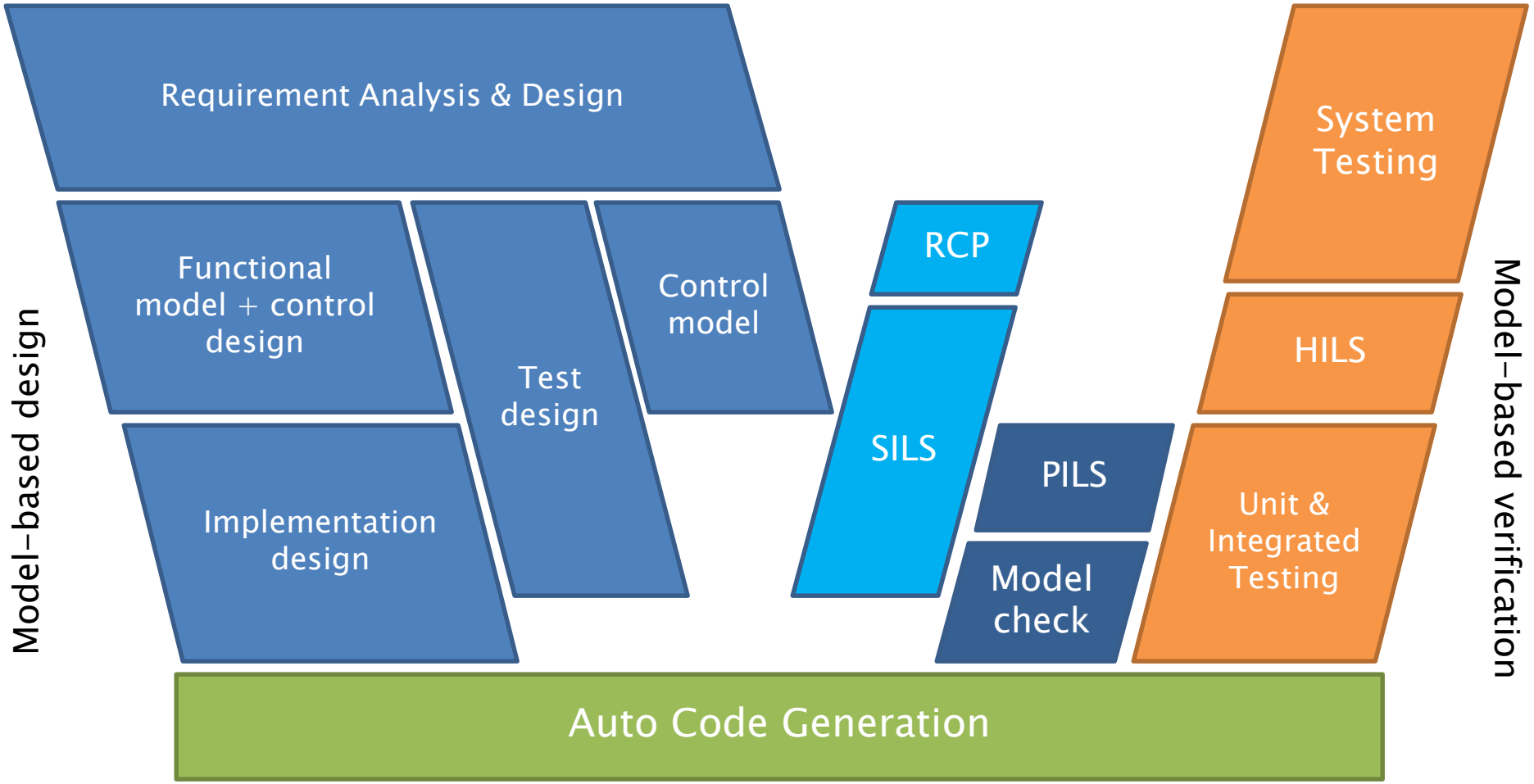
Rapid prototyping

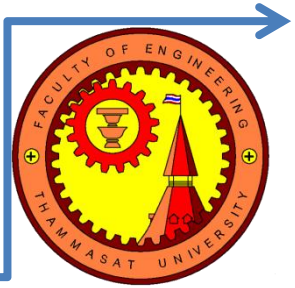


Hardware-In-Loop Simulation



V-model for MBD



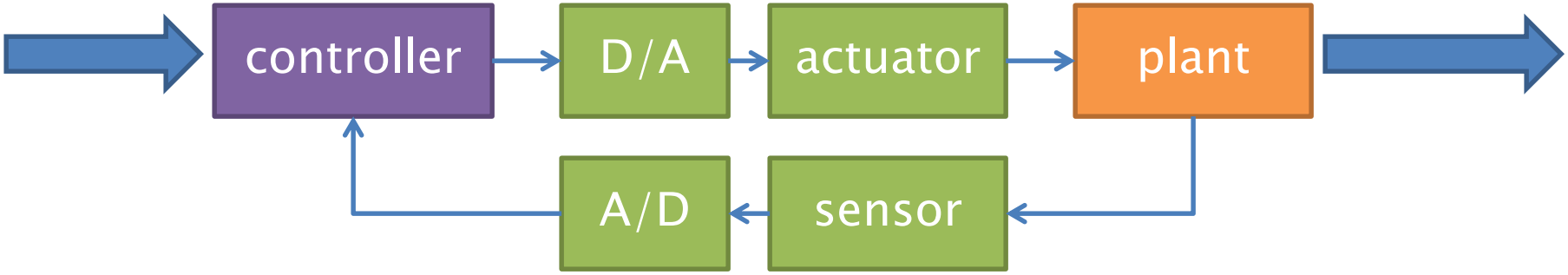


MODELING OF MECHATRONICS

Mechatronic systems

controller model

plant model



MBD for feedback control

Plant modeling

Simulink/SimScape

Controller design & synthesis

Simulink/Stateflow

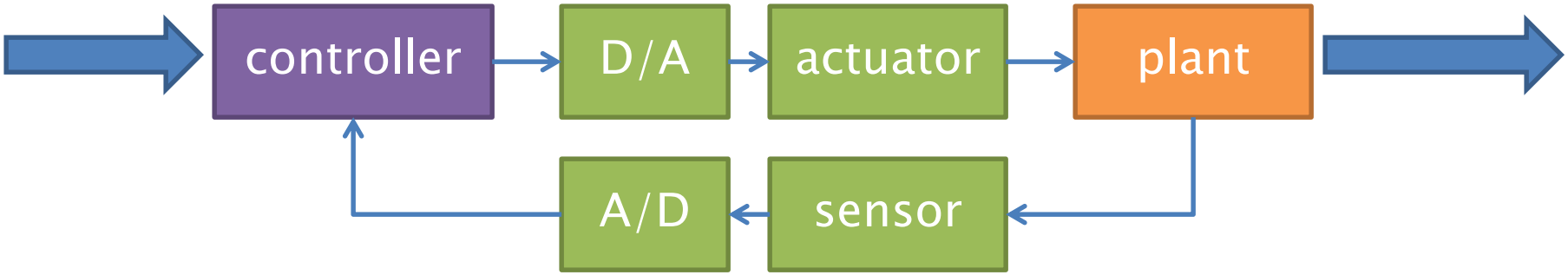
Simulation: offline & real-time

Simulink

Hardware deployment

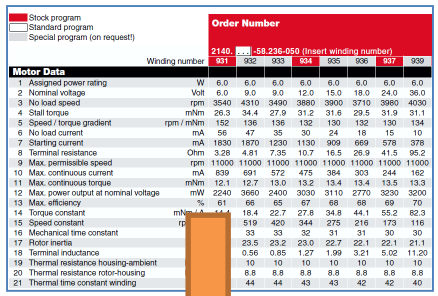
Embedded Coder

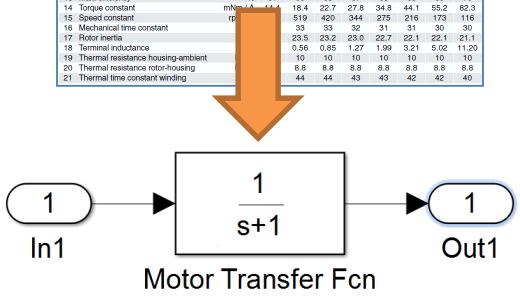
System modeling



Bottom-up approach

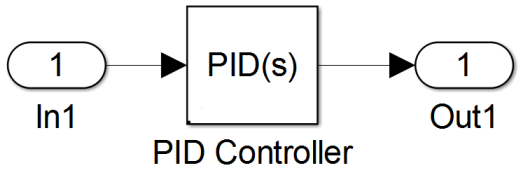
Top-down approach





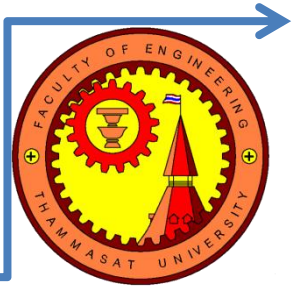
Requirements:

1. Control system shall not have any steady-state error.
2. ...



...

Signal and systems



Wikibooks

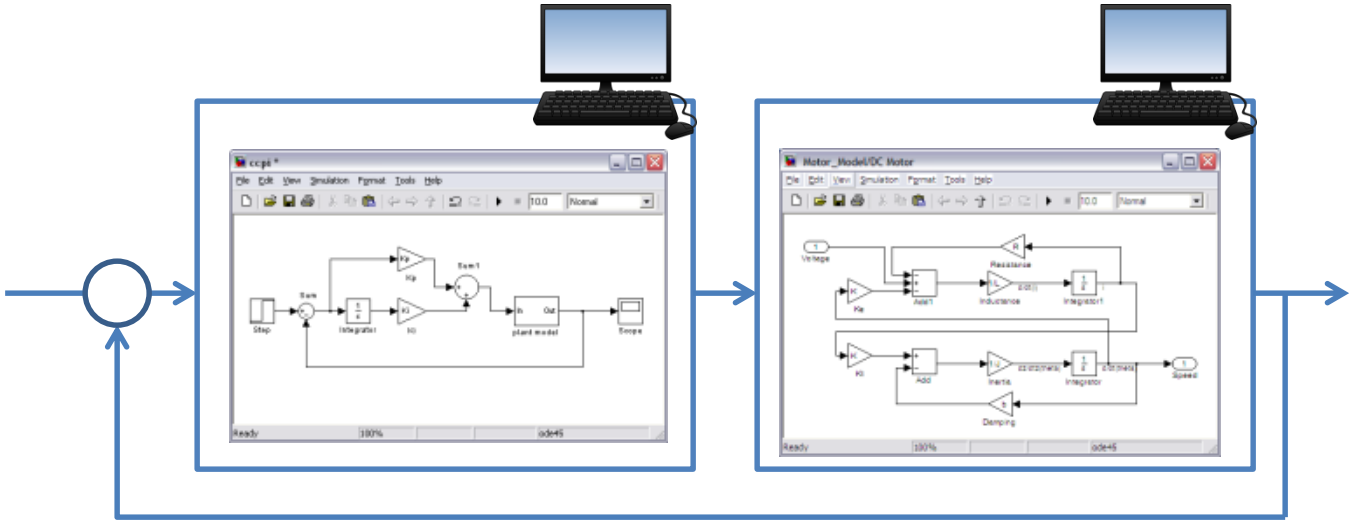
Signal is function of independent variables that carry some information.
System is any physical set of components that takes a signal, and produces a signal.

signal → system →

analog	digital
deterministic	nondeterministic
period	aperiodic

analog	logic
continuous	discrete-time
linear	nonlinear
time-invariance	time-varying

Simulation in feedback control



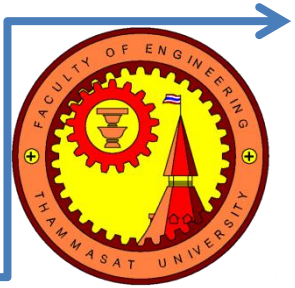
Model/Software-In-Loop Simulation

Proof of concept/method/algorithm

Processor-In-Loop Simulation

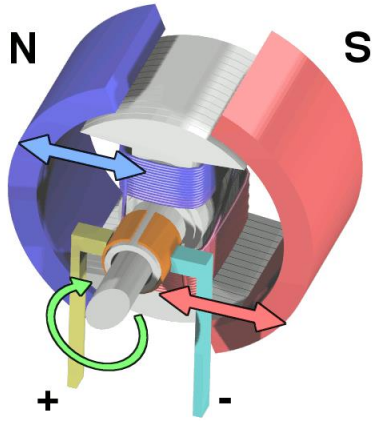
Rapid prototyping

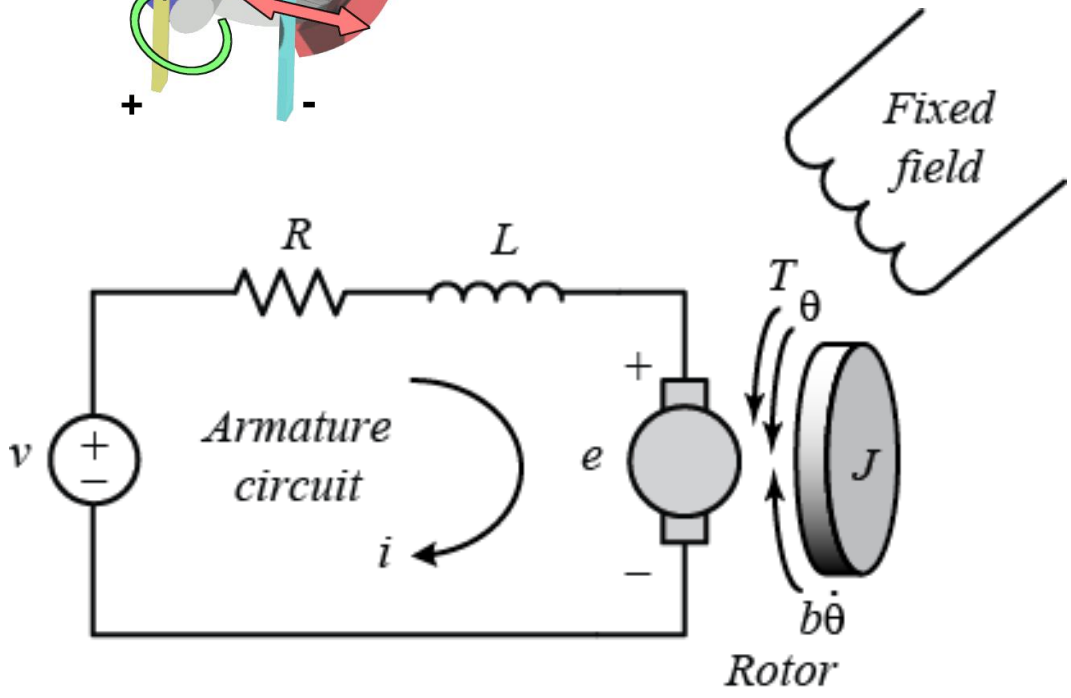
Hardware-In-Loop Simulation



DC MOTOR MODELING

DC motor modeling





Electrical equations

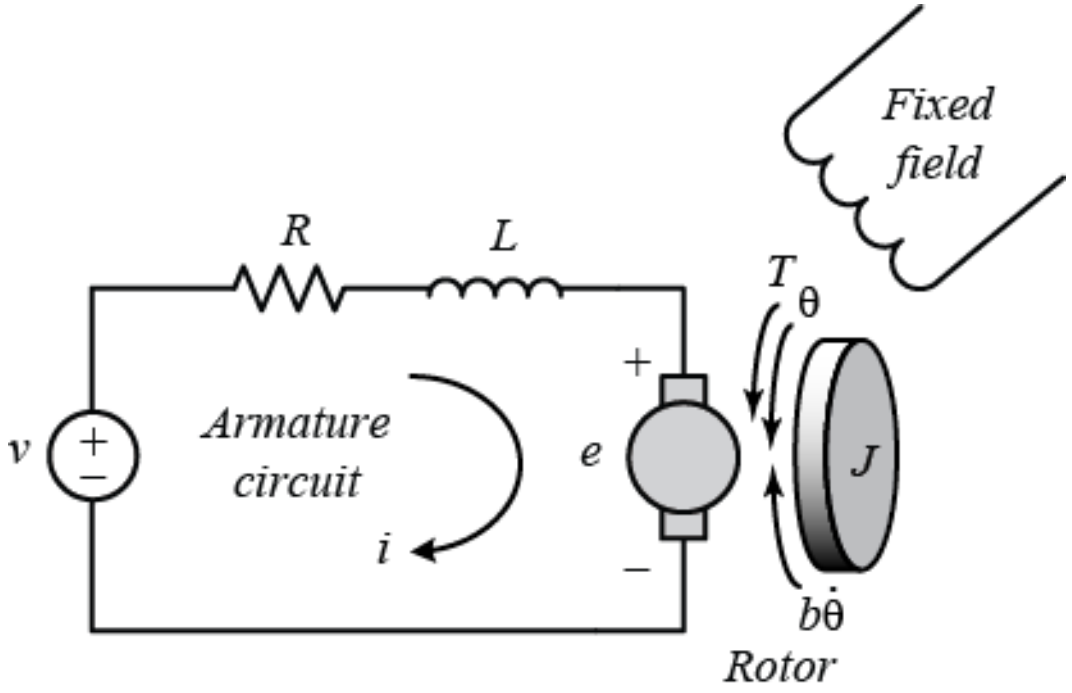
Kirchhoff's circuit laws

Energy conversion

Mechanical equations

Newton's laws of motion

First-principles modeling



Torque equation

$$T = K_i i(t)$$

Back EMF. equation

$$e = K_e \omega(t)$$

Electrical equation

$$L\frac{di(t)}{dt} + Ri(t) + e(t) = v(t)$$

Motion equation

$$J\frac{d\omega(t)}{dt} + b\omega(t) = T(t)$$

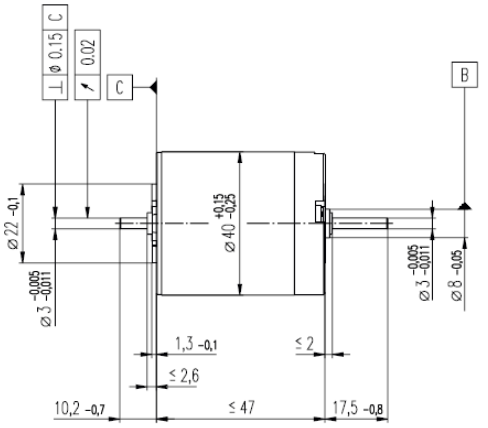
Maxon motor: 2140 937

Stock program
Standard program
Special program (on request!)

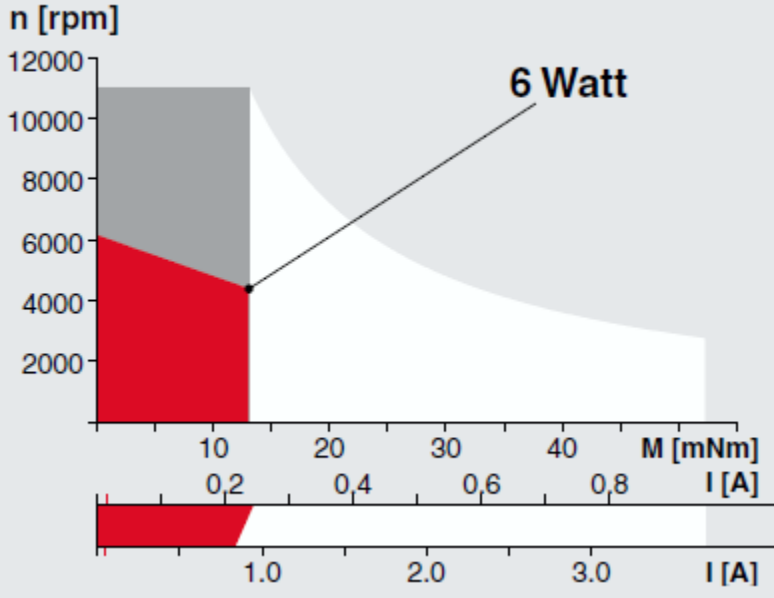
Order Number

2140. [...] -58.236-050 (Insert winding number)

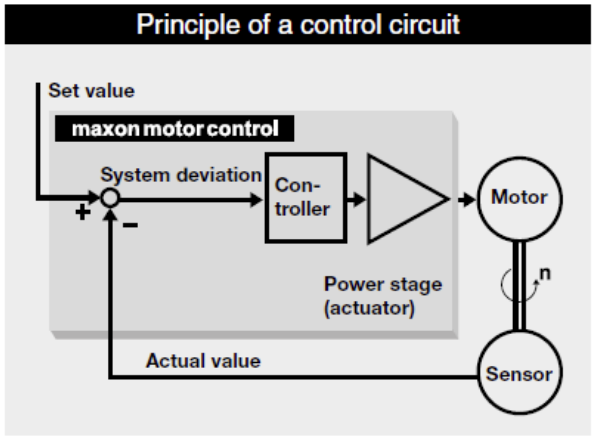
			Winding number	931	932	933	934	935	936	937	939
Motor Data											
1	Assigned power rating	W		6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0
2	Nominal voltage	Volt		6.0	9.0	9.0	12.0	15.0	18.0	24.0	36.0
3	No load speed	rpm		3540	4310	3490	3880	3900	3710	3980	4030
4	Stall torque	mNm		26.3	34.4	27.9	31.2	31.6	29.5	31.9	31.1
5	Speed / torque gradient	rpm / mNm		152	136	136	132	130	132	130	134
6	No load current	mA		56	47	35	30	24	18	15	10
7	Starting current	mA		1830	1870	1230	1130	909	669	578	378
8	Terminal resistance	Ohm		3.28	4.81	7.35	10.7	16.5	26.9	41.5	95.2
9	Max. permissible speed	rpm		11000	11000	11000	11000	11000	11000	11000	
10	Max. continuous current	mA		839	691	572	475	384	303	244	
11	Max. continuous torque	mNm		12.1	12.7	13.0	13.2	13.4	13.4	13.5	
12	Max. power output at nominal voltage	mW		2240	3660	2400	3030	3110	2770	3230	
13	Max. efficiency	%		61	66	65	67	68	68	69	
14	Torque constant	mNm / A		14.4	18.4	22.7	27.8	34.8	44.1	55.2	
15	Speed constant	rpm / V		664	519	420	344	275	216	173	
16	Mechanical time constant	ms		36	33	33	32	31	31	30	
17	Rotor inertia	gcm²		22.9	23.5	23.2	23.0	22.7	22.1	22.1	
18	Terminal inductance	mH		0.34	0.56	0.85	1.27	1.99	3.21	5.02	
19	Thermal resistance housing-ambient	K / W		10	10	10	10	10	10	10	
20	Thermal resistance rotor-housing	K / W		8.8	8.8	8.8	8.8	8.8	8.8	8.8	
21	Thermal time constant winding	s		43	44	44	43	43	42	42	

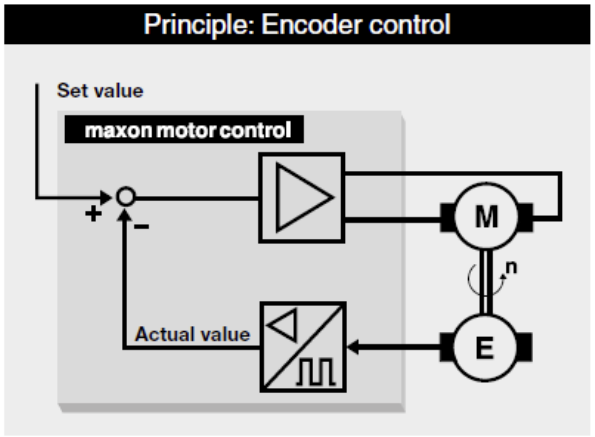


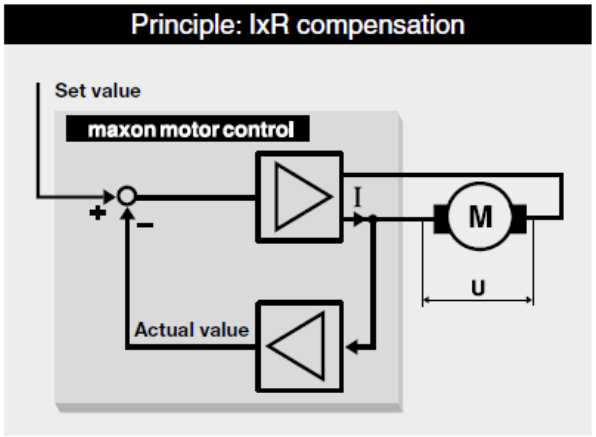
Operating Range

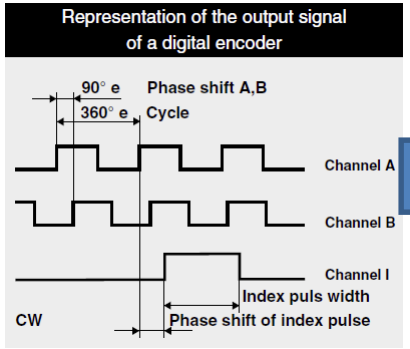


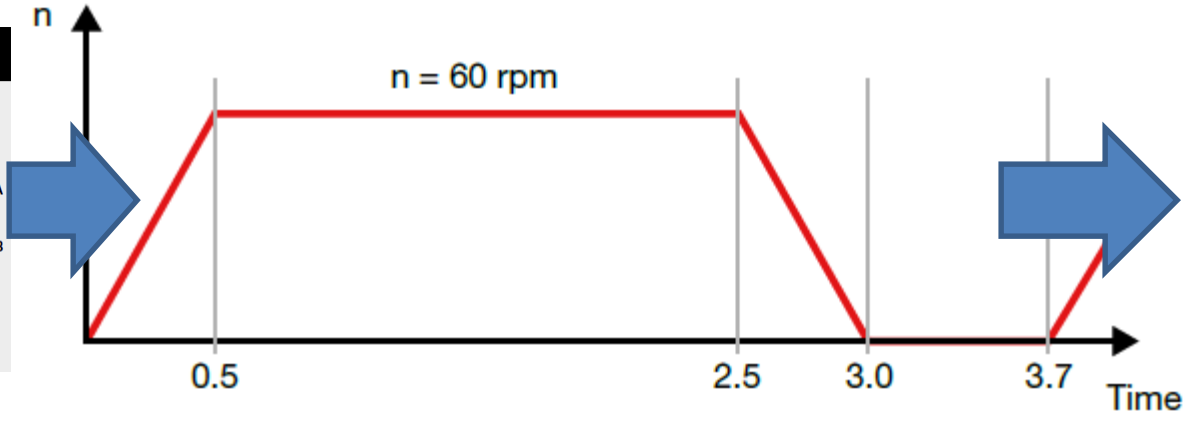
DC motor control

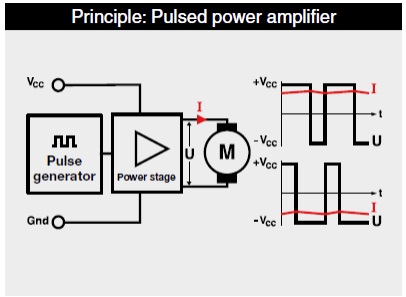






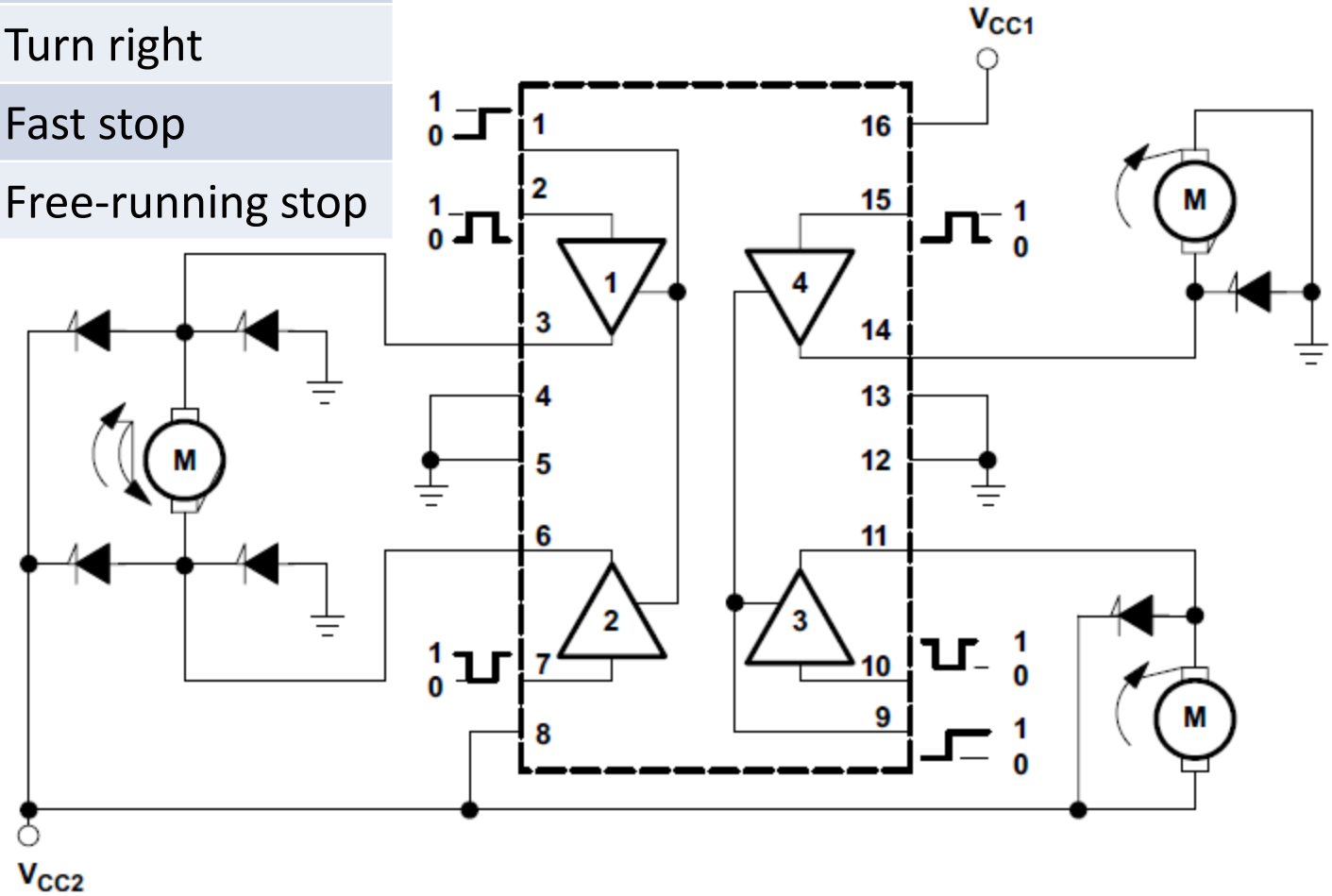




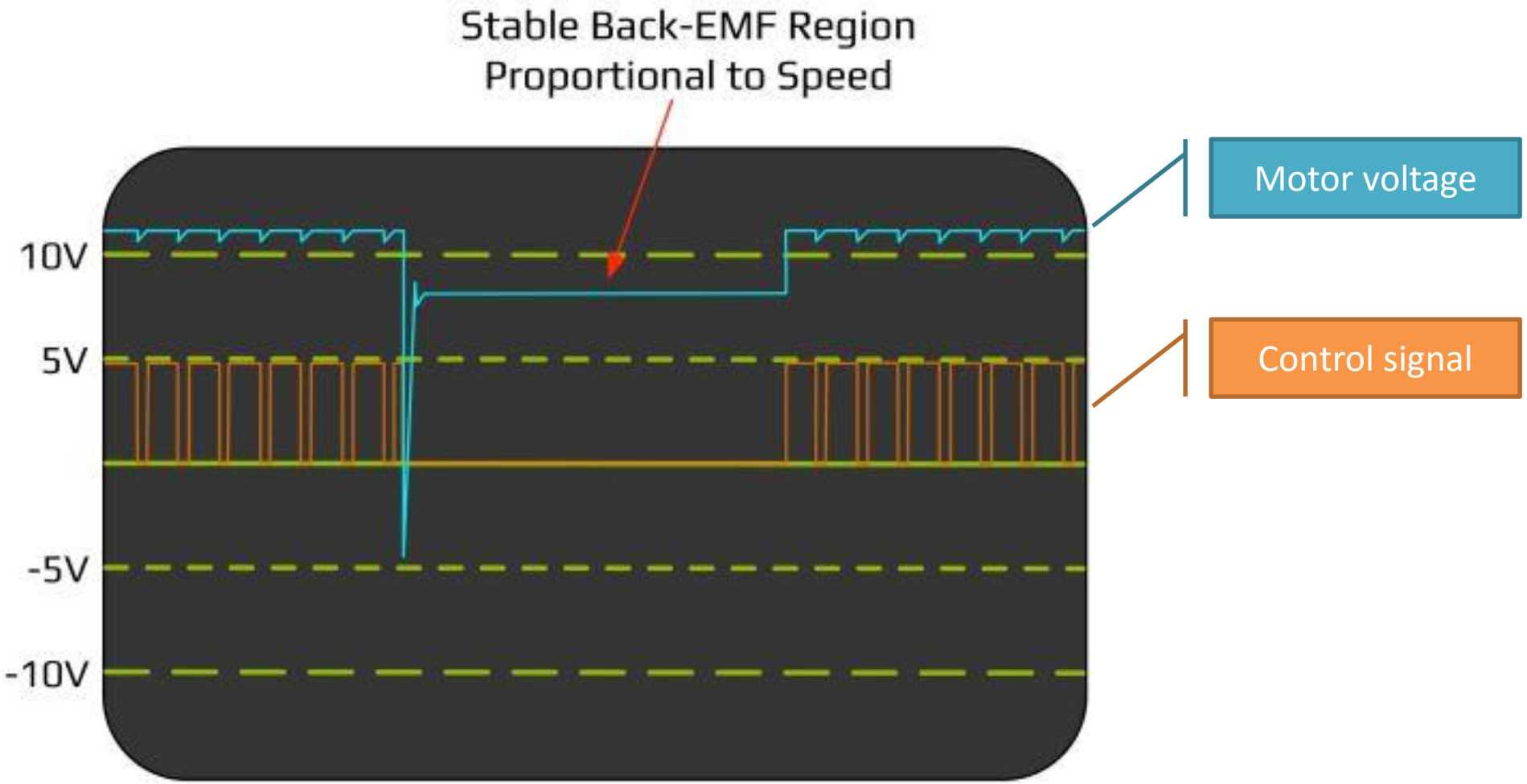


Actuator: H-bridge + PWM

EN	1A	2A	Function
H	L	H	Turn left
H	H	L	Turn right
H	L/H	L/H	Fast stop
L	x	x	Free-running stop



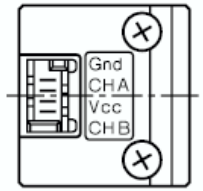
Actuator: PWM



Maxon encoder: 103935

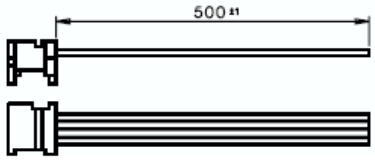
Technical Data	
Supply voltage	5 V ± 10 %
Output signal	TTL compatible
Number of channels	2
Counts per turn	100
Phase shift Φ (nominal)	90°e
Logic state width s	min. 45°e
Signal rise time (typical at C_L = 25 pF, R_L = 11 kΩ, 25°C)	200 ns
Signal fall time (typical at C_L = 25 pF, R_L = 11 kΩ, 25°C)	50 ns
Operating temperature range	-20 ... +85°C
Moment of inertia of code wheel	≤ 0,05 gcm²
Output current per channel	min. -1 mA, max. 5 mA
Max. operating frequency	min. 20 kHz

Pin Allocation

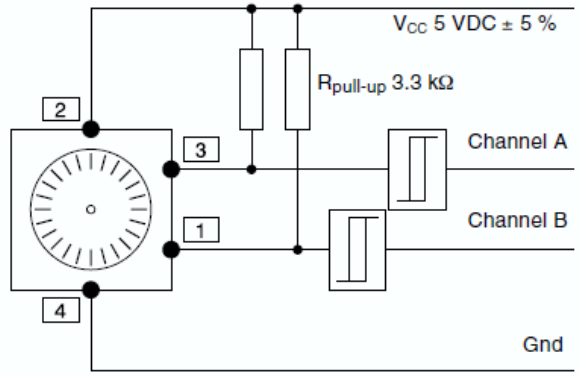


Micromodule contact strip
Type Lumberg MICS 4
Pin 4 Gnd
Pin 3 Channel A
Pin 2 V_{CC}, Pin 1 Channel B
Recommended connectors:
Micromodule connector
Type Lumberg MICA 4

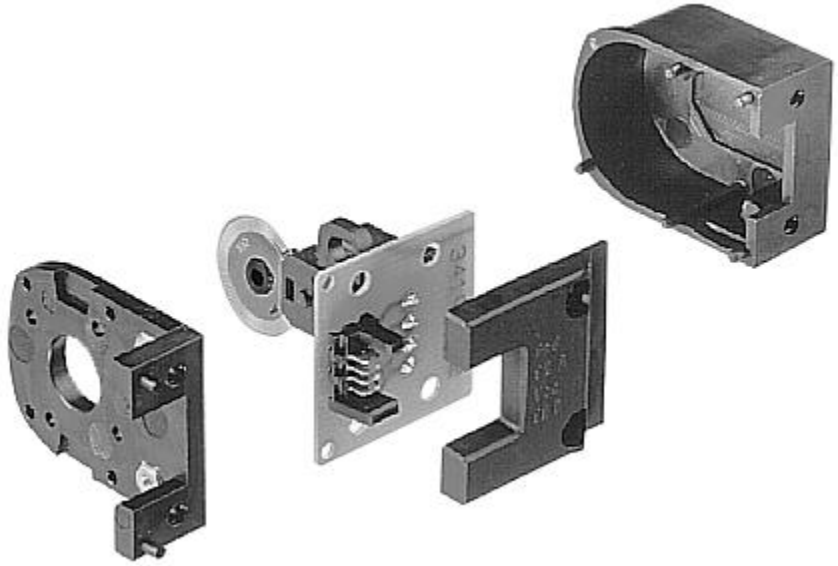
Order number for connector: 3419.506

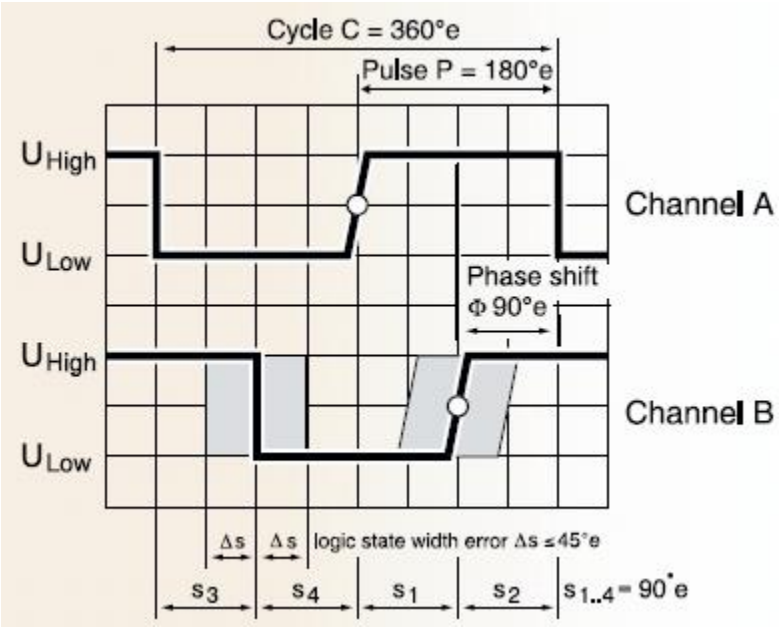


Connection Example

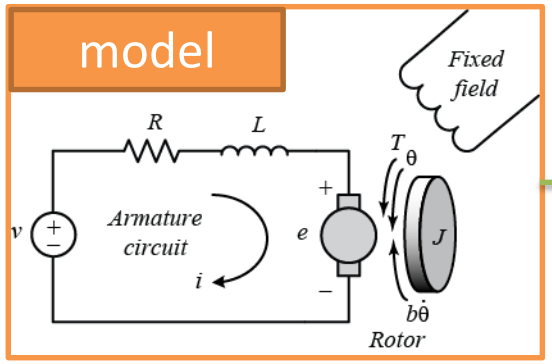


Ambient temperature δ_U = 22 - 25°C





Data-driven modeling



Parametrization

Model

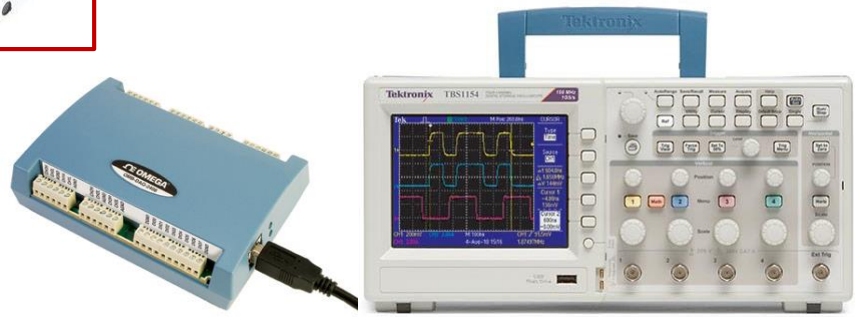
System identification

Experiment

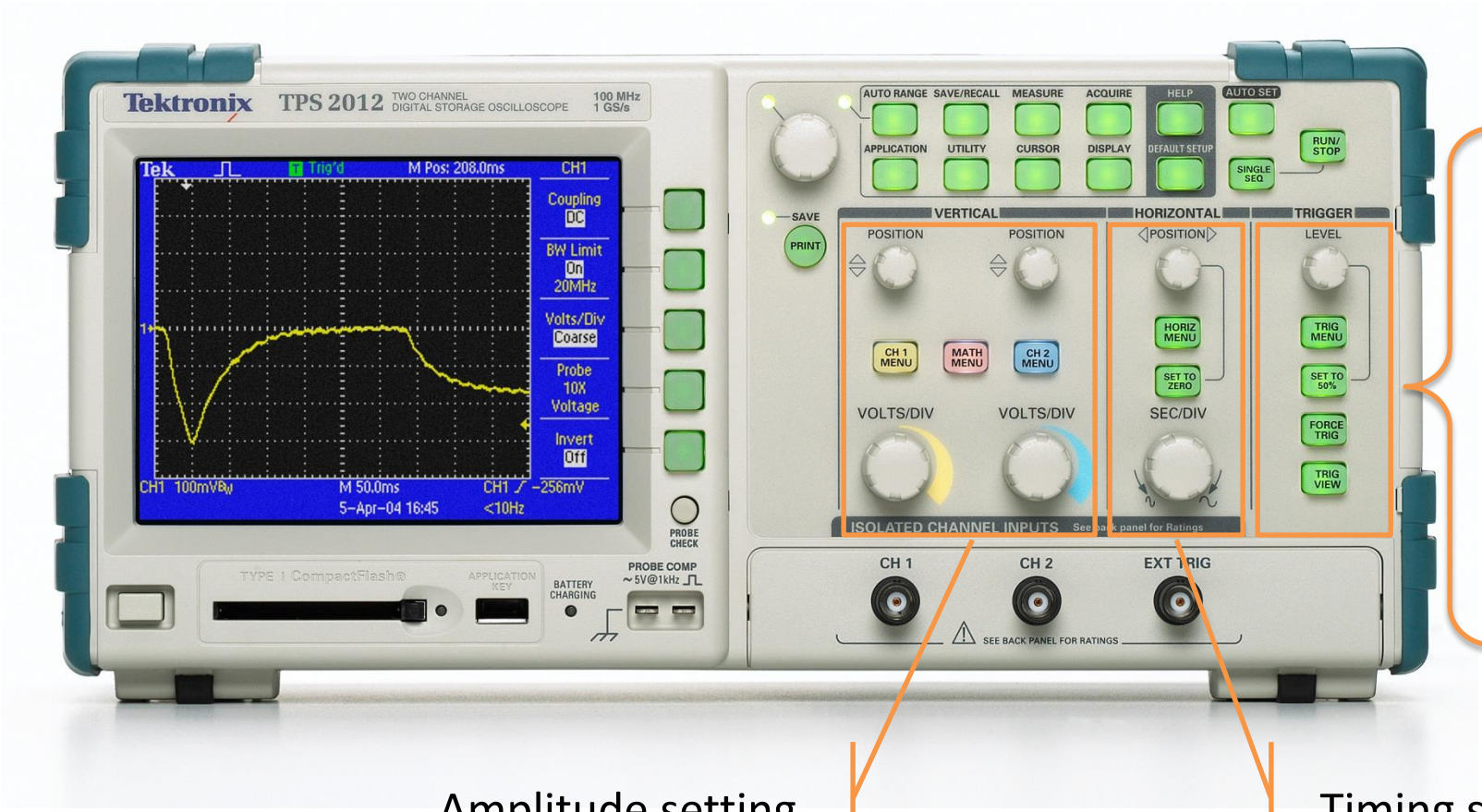
Perturbation signal

Excited output

- Step signal
- Swept sine signal
- Pseudo-Random Binary Sequence



Using oscilloscope



Mode:
NORMAL

Level:
2~4 volt

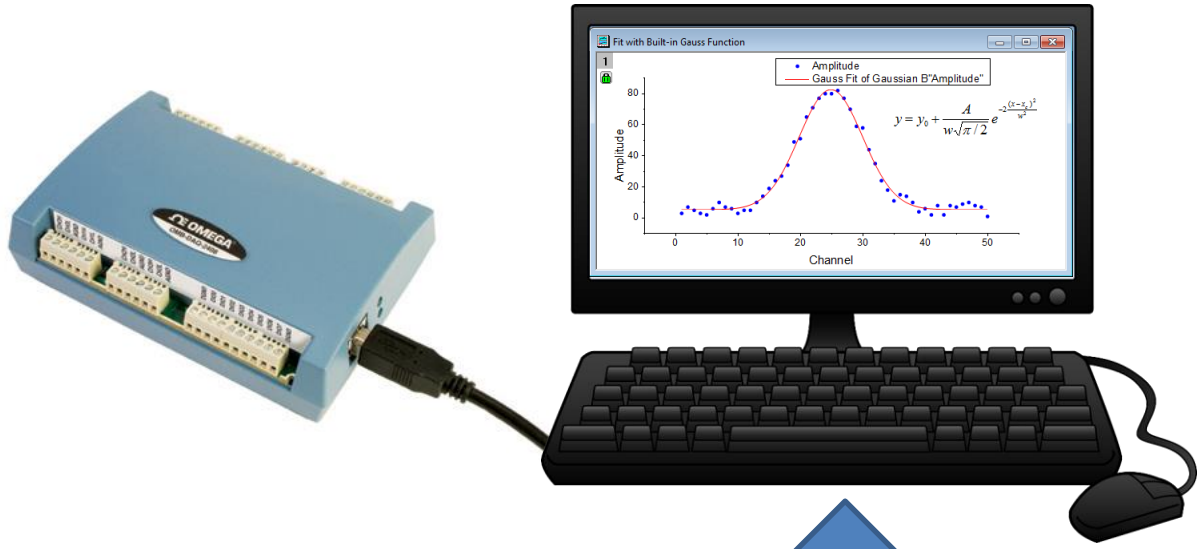
Slope:
Rising or Falling

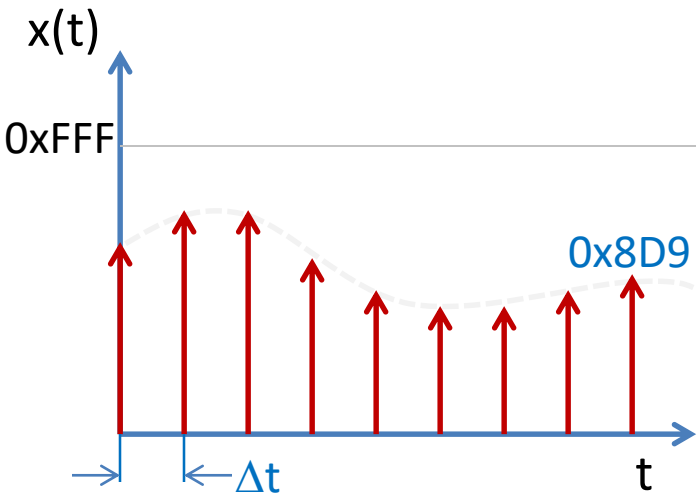
Amplitude setting

Timing setting

Using DAQ

System



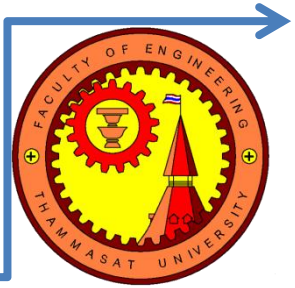


polling

cont. sampling

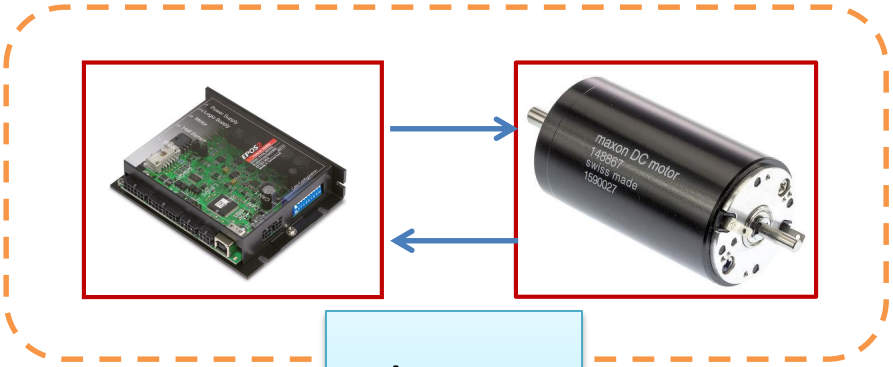
Analysis

storage



CONTROLLER DESIGN

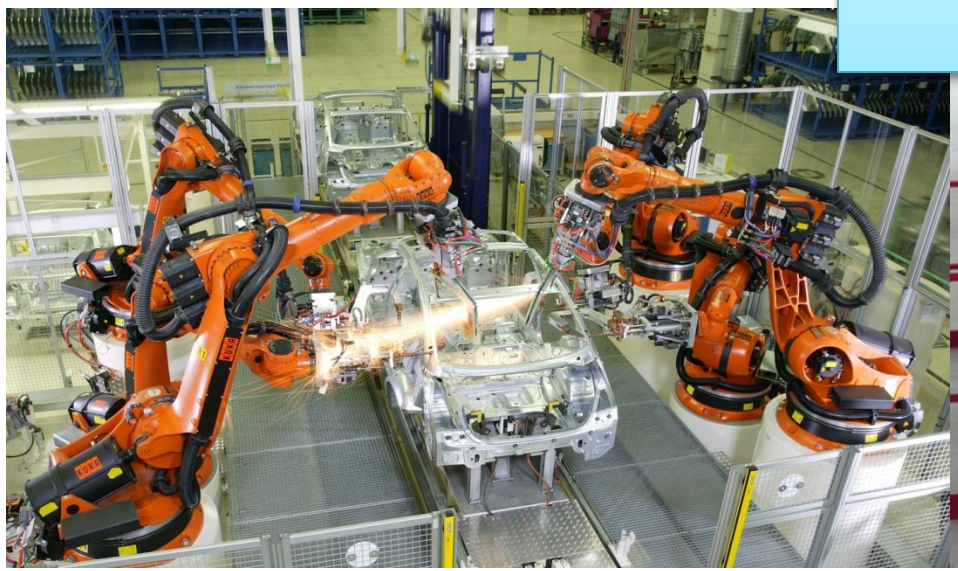
Control system design



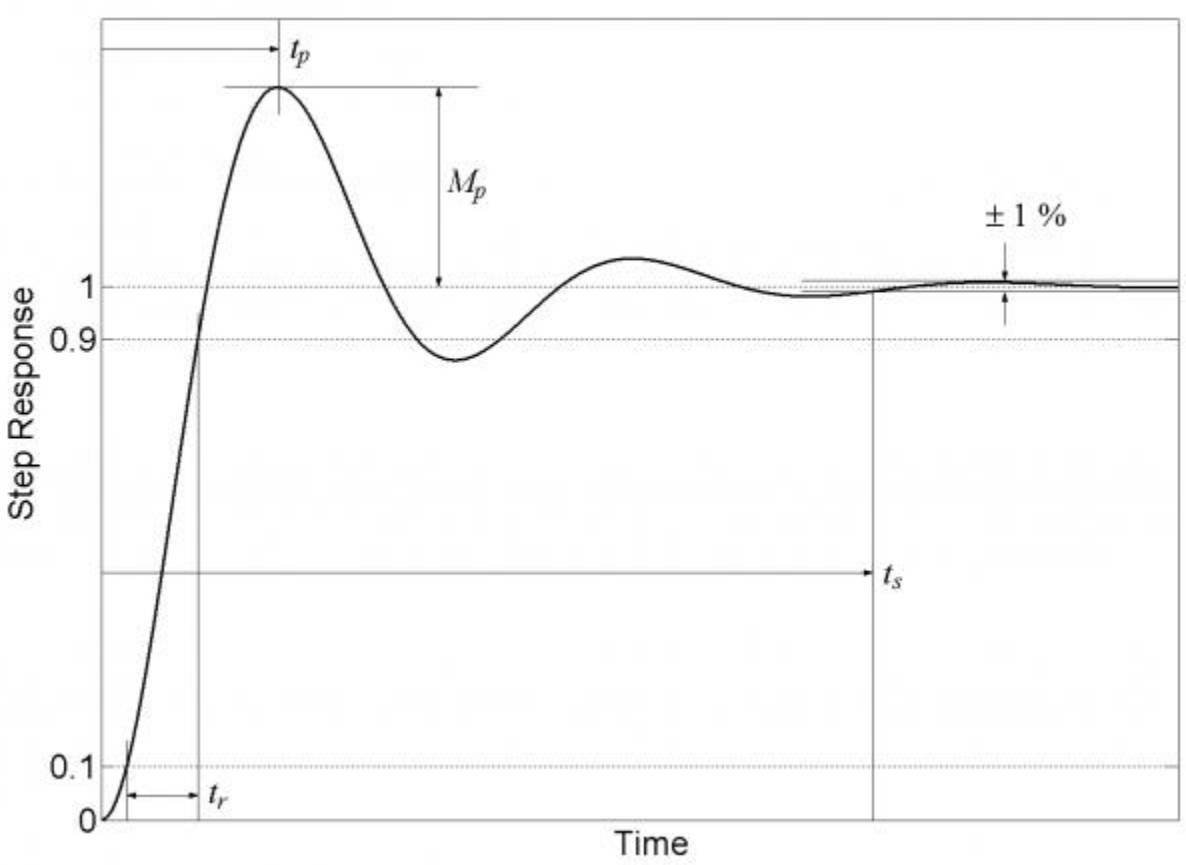
Position control

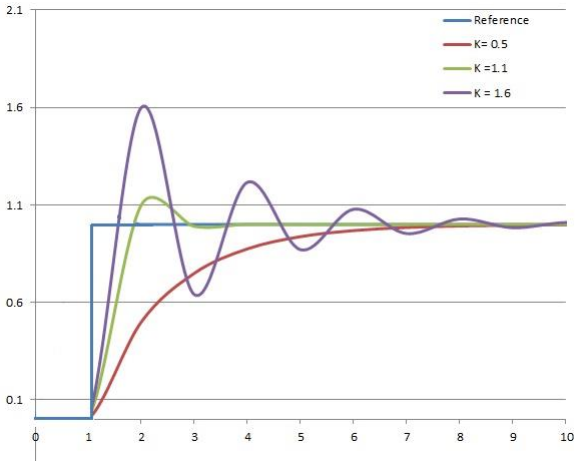
Speed control

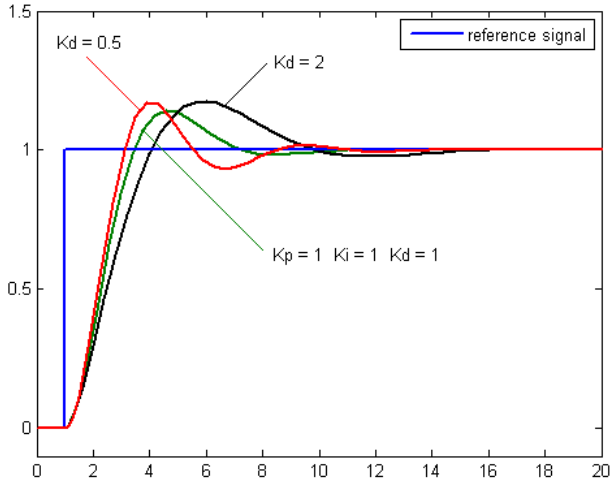
design spec.



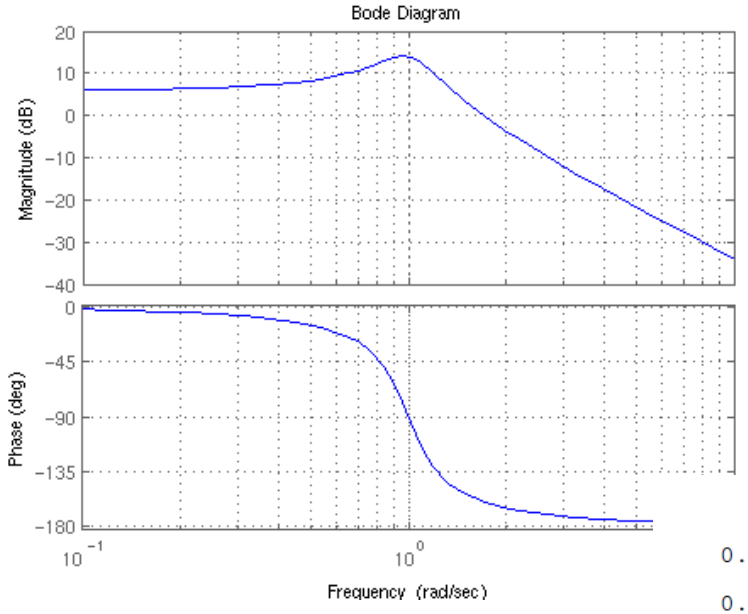
Time-domain performance







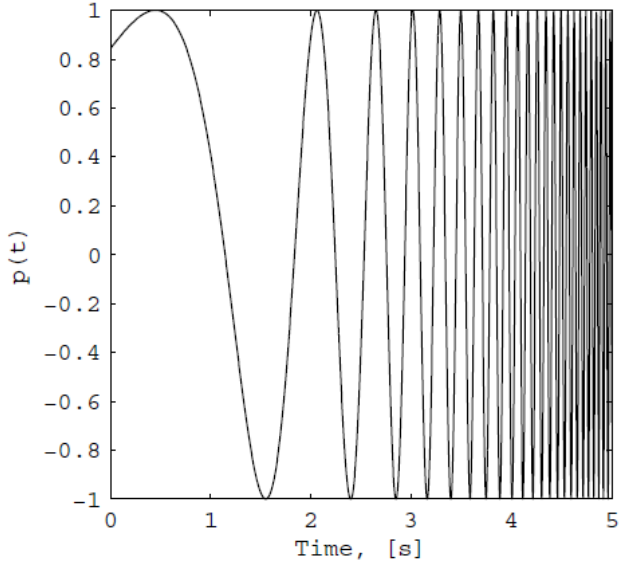
Frequency-domain performance

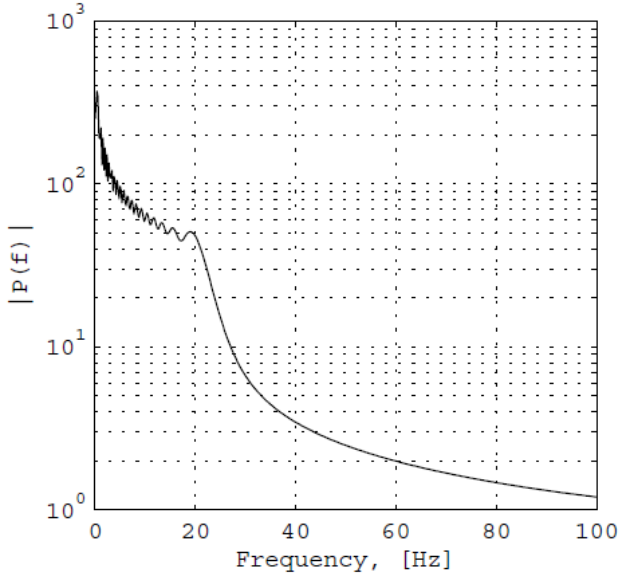


Frequency Response Function

$$G(j\omega) = \mathbf{Re}(G(j\omega)) + j\mathbf{Im}(G(j\omega))$$

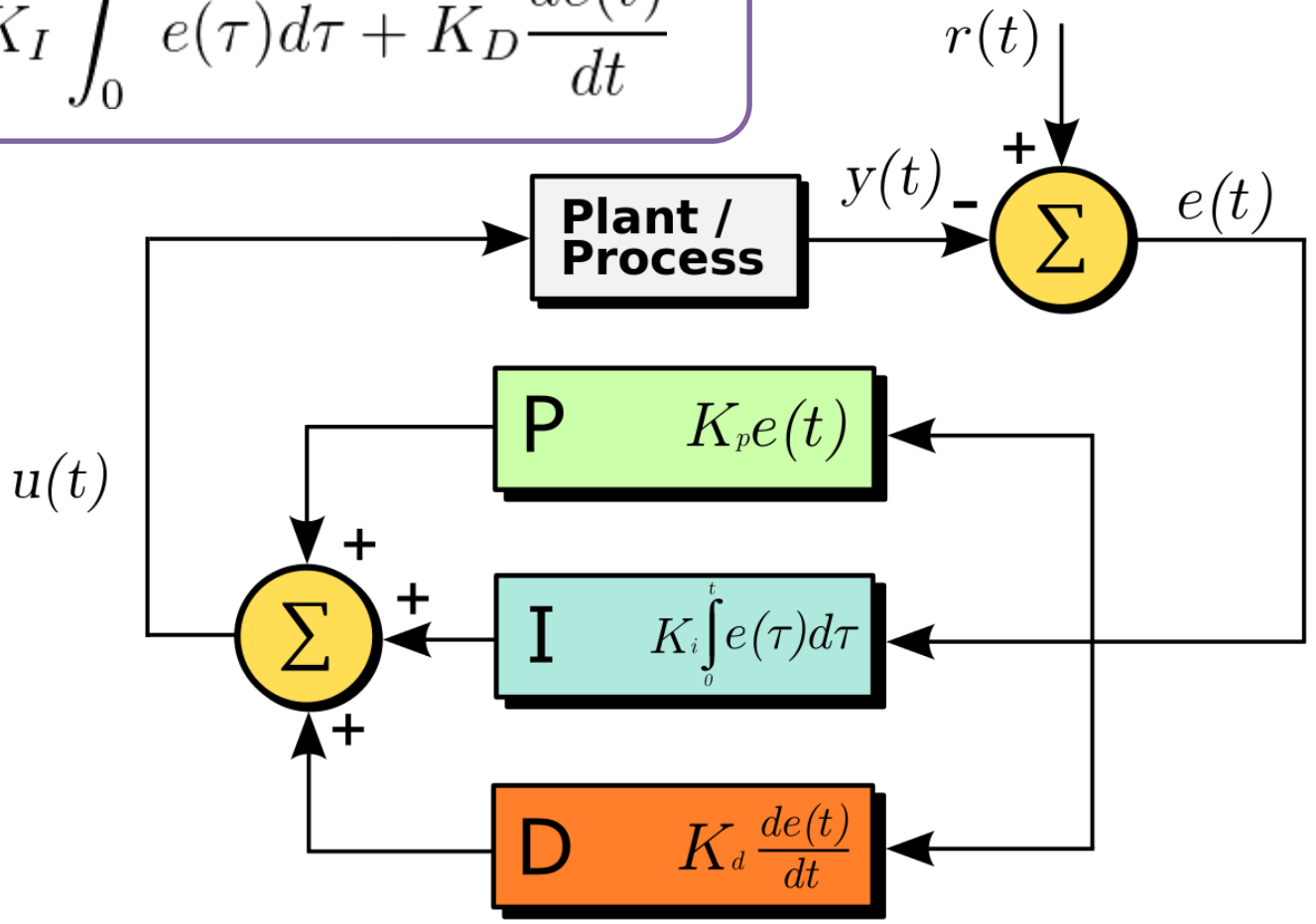
$$G(j\omega) = M(\omega)\angle\theta(\omega)$$



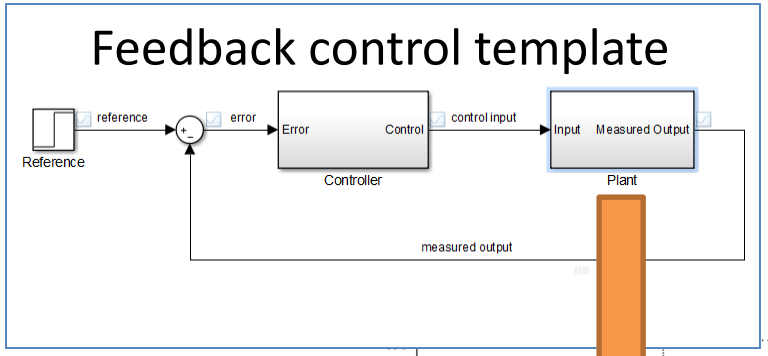


PID controller

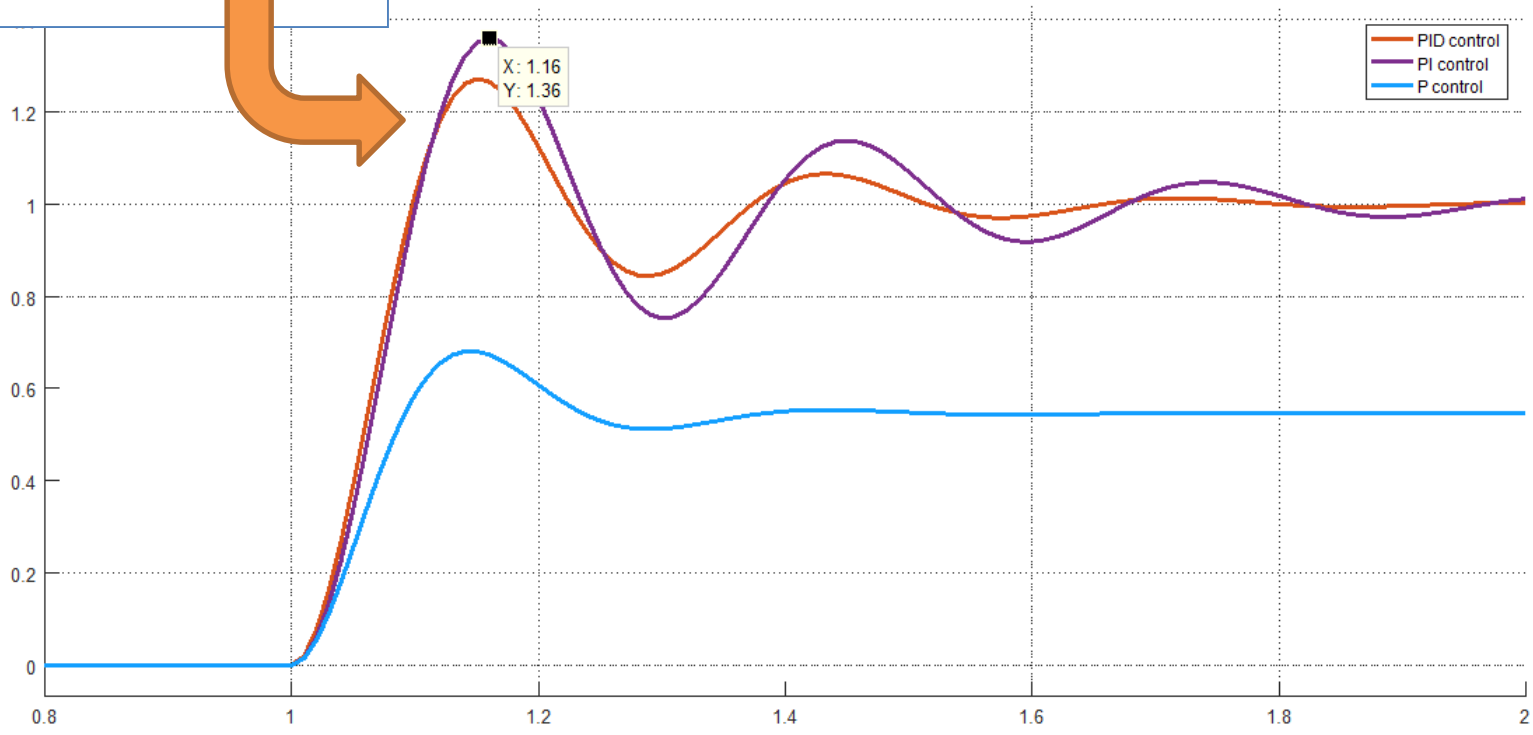
$$u(t) = K_p e(t) + K_I \int_0^t e(\tau)d\tau + K_D \frac{de(t)}{dt}$$



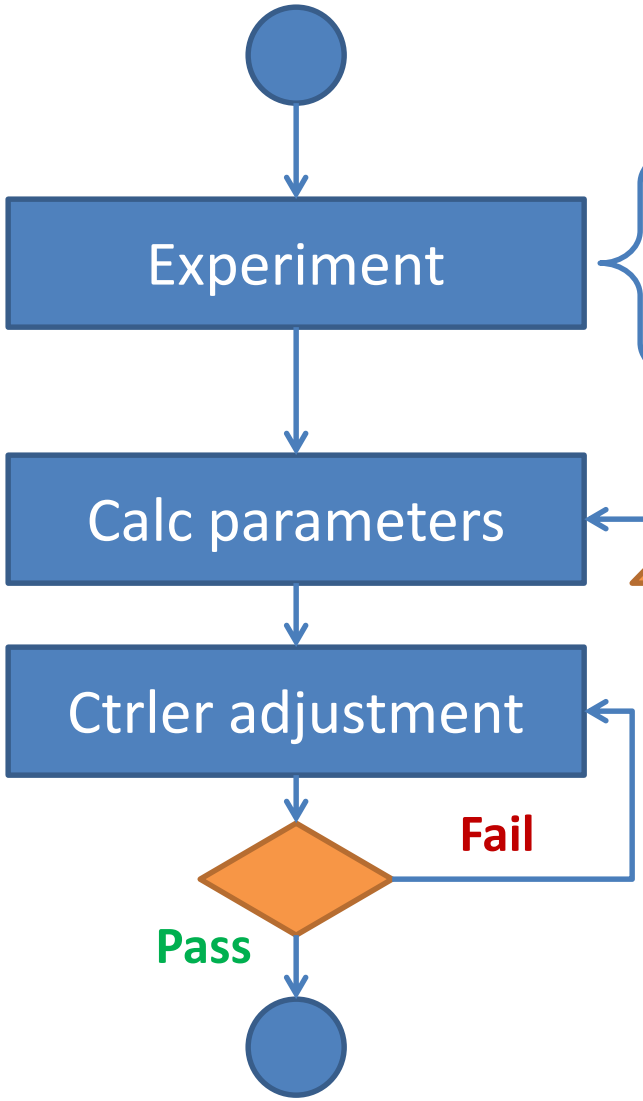
PID control → performance



- Proportional gain = faster, more oscillation
- Integral gain = solve ess, more oscillation
- Derivative gain = faster, less oscillation



Controller tuning



Open-loop tuning procedure

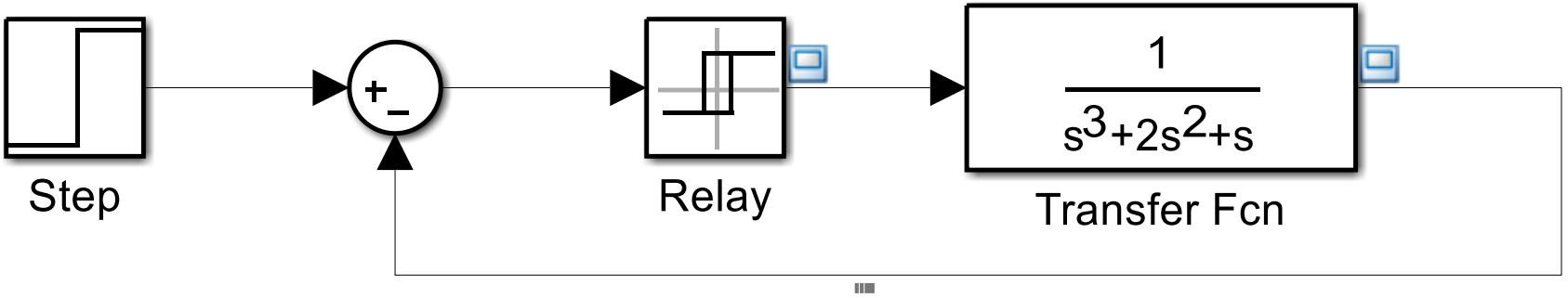
Closed-loop tuning procedure

Tuning chart

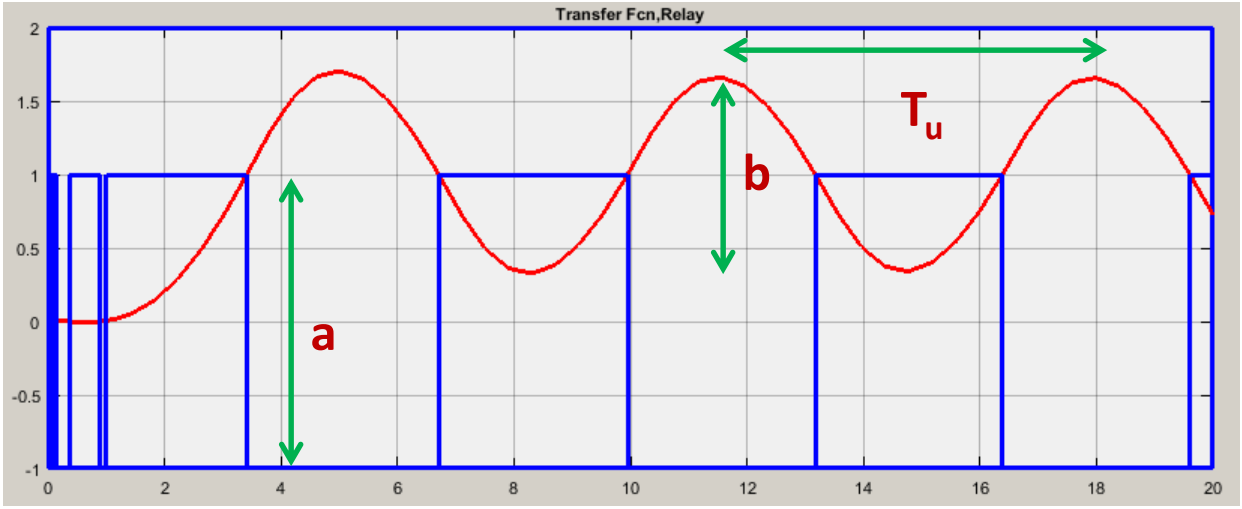
Ziegler-Nichols tuning chart: closed-loop data

Controller	K_p	K_i	K_d
P	$K_u/2$		
PI	$K_u/2.2$	$1.2/T_u$	
PID	$K_u/1.7$	$2/T_u$	$T_u/8$

Relay tuning technique

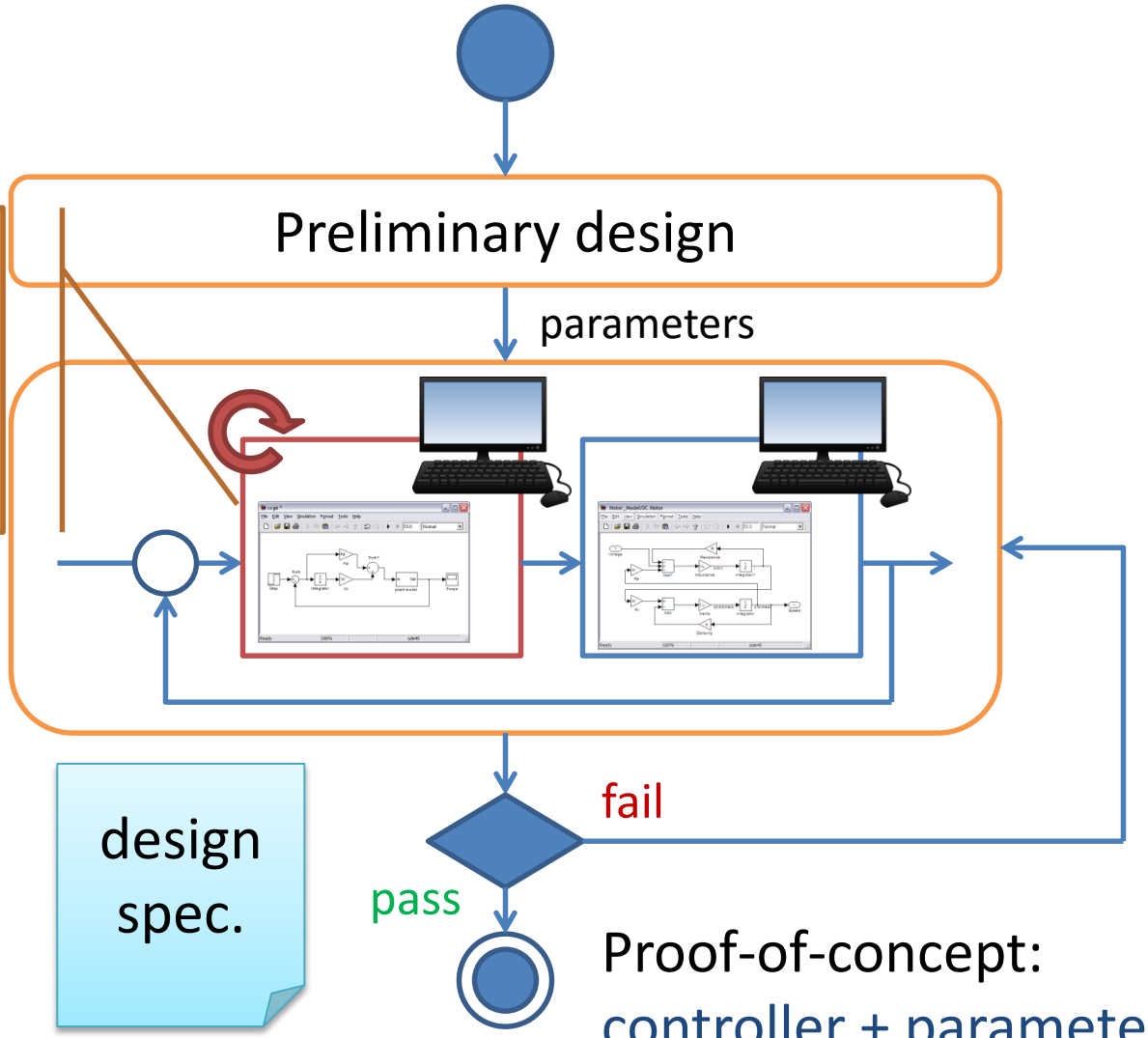


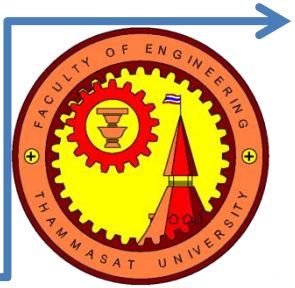
$$K_u = \frac{4}{\pi}\left(\frac{a}{b}\right)$$



Model-In-Loop Simulation

Differential Algebraic Equations
- variable-step solver
- zero-crossing detection
- IEEE 754 double precision

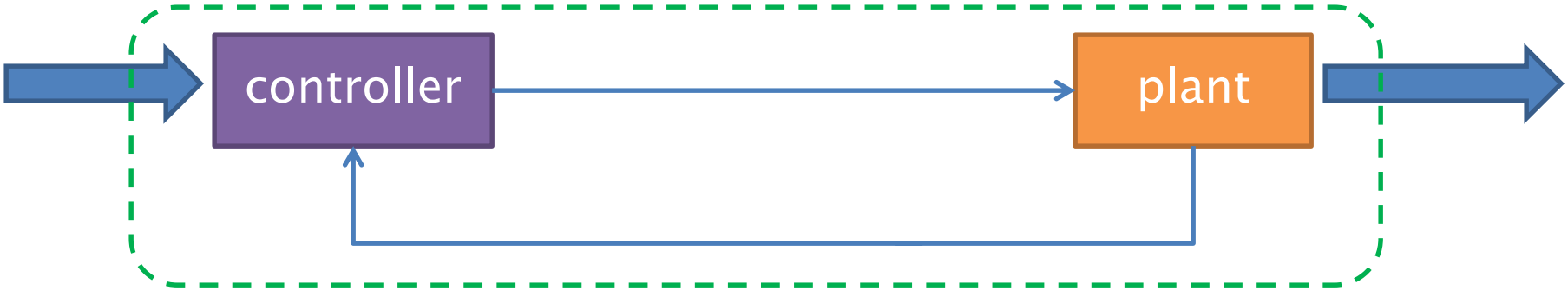




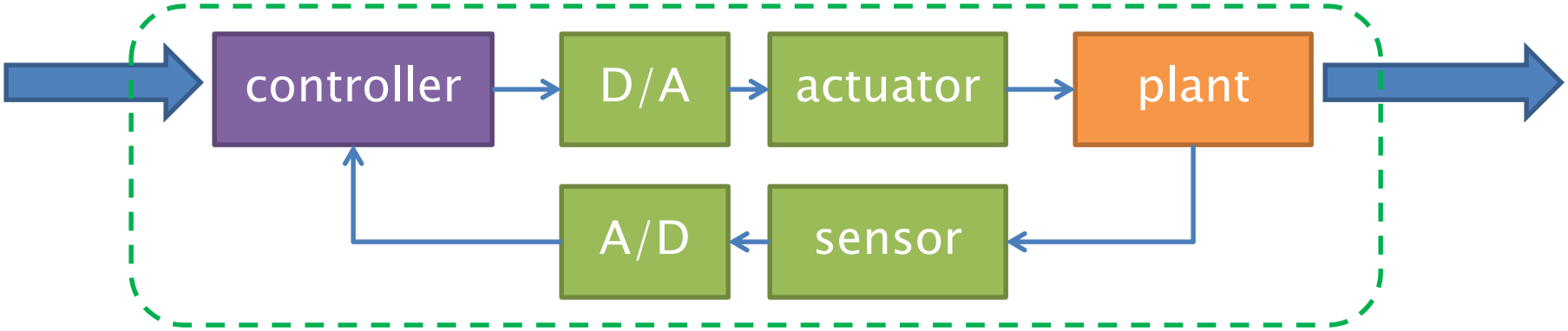
CONTROLLER REALIZATION

Controller design and synthesis

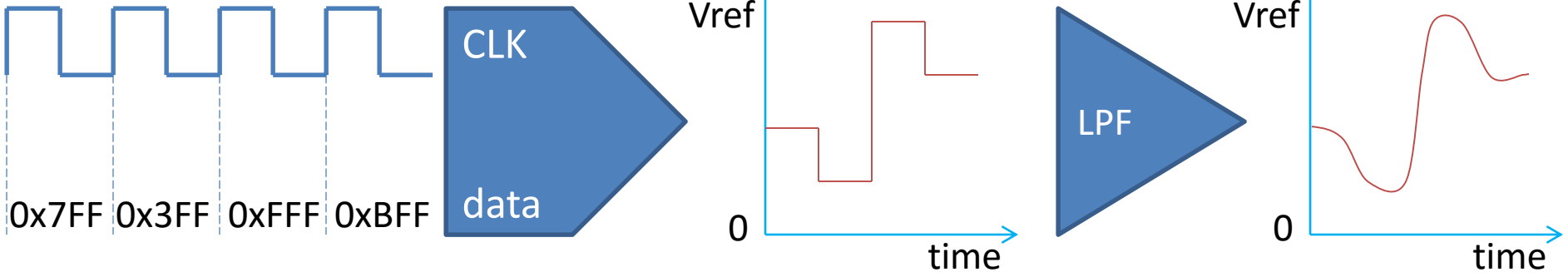
Theoretical viewpoint



Practical viewpoint

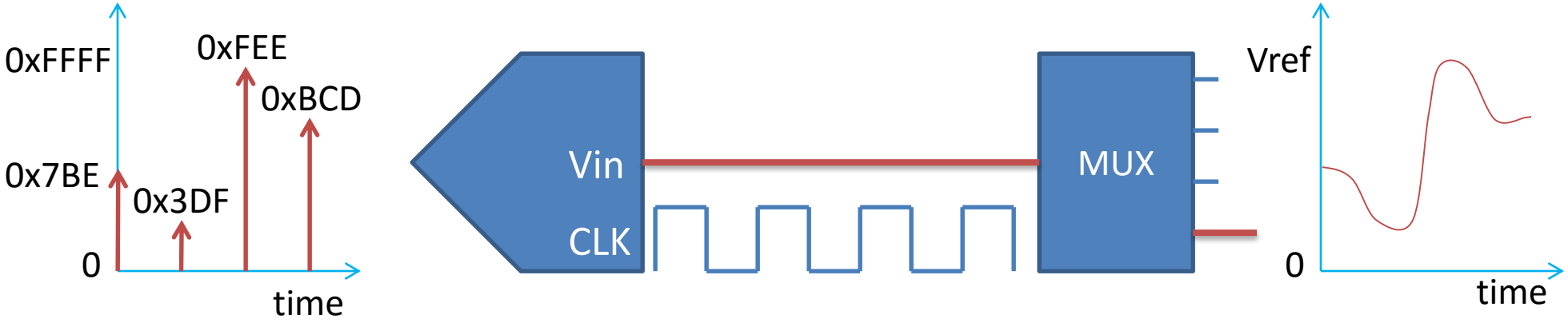


Discretization

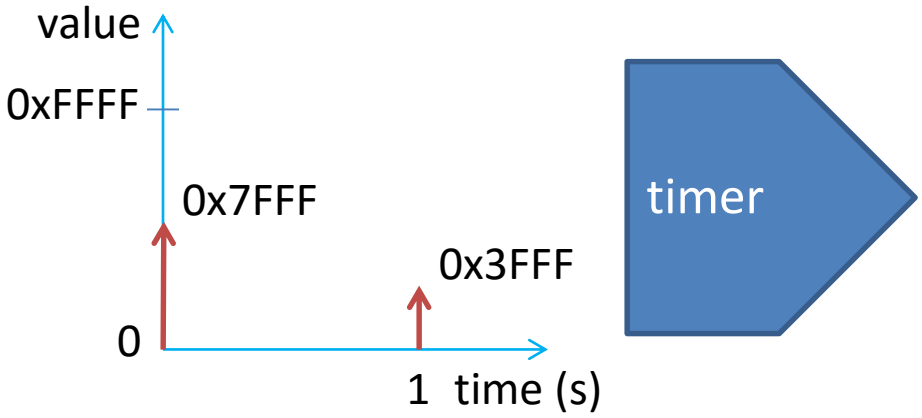


digital-to-analog conversion →

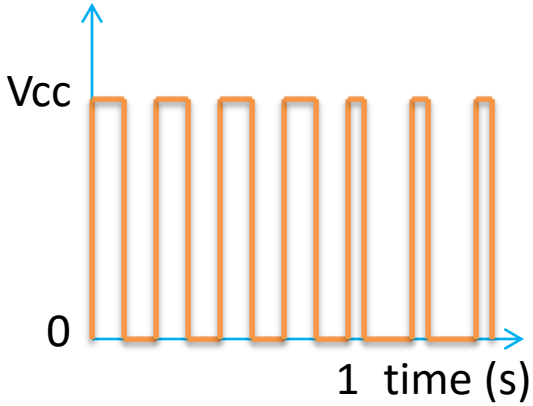
← analog-to-digital conversion

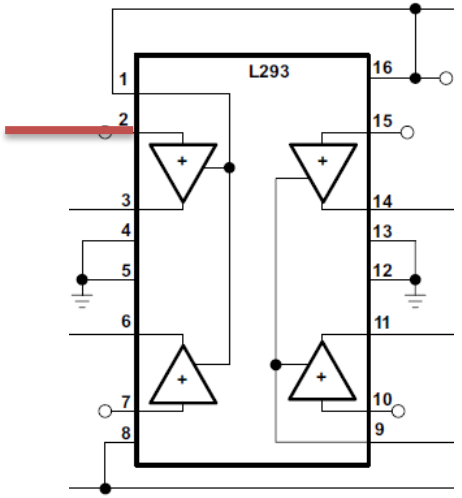


Discretization: motor control



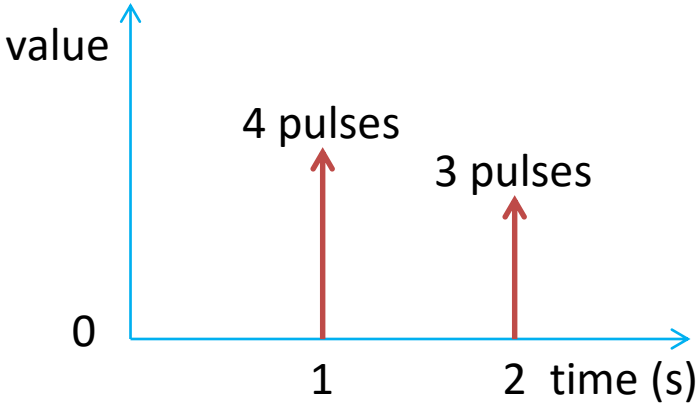
timer



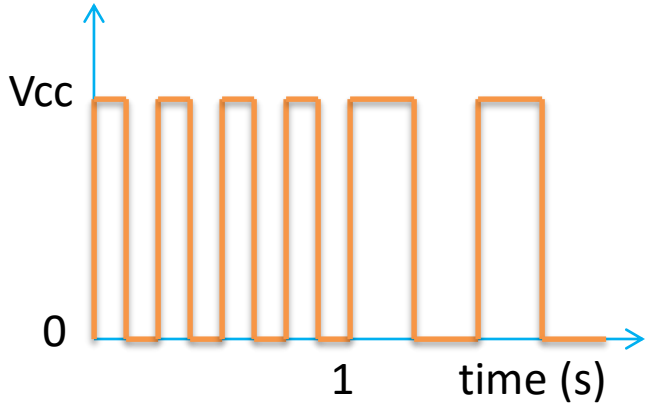


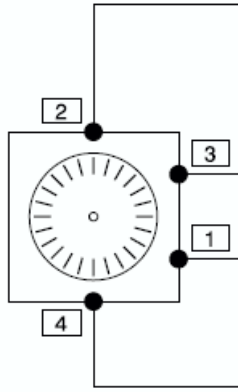
digital-to-analog conversion →

← analog-to-digital conversion

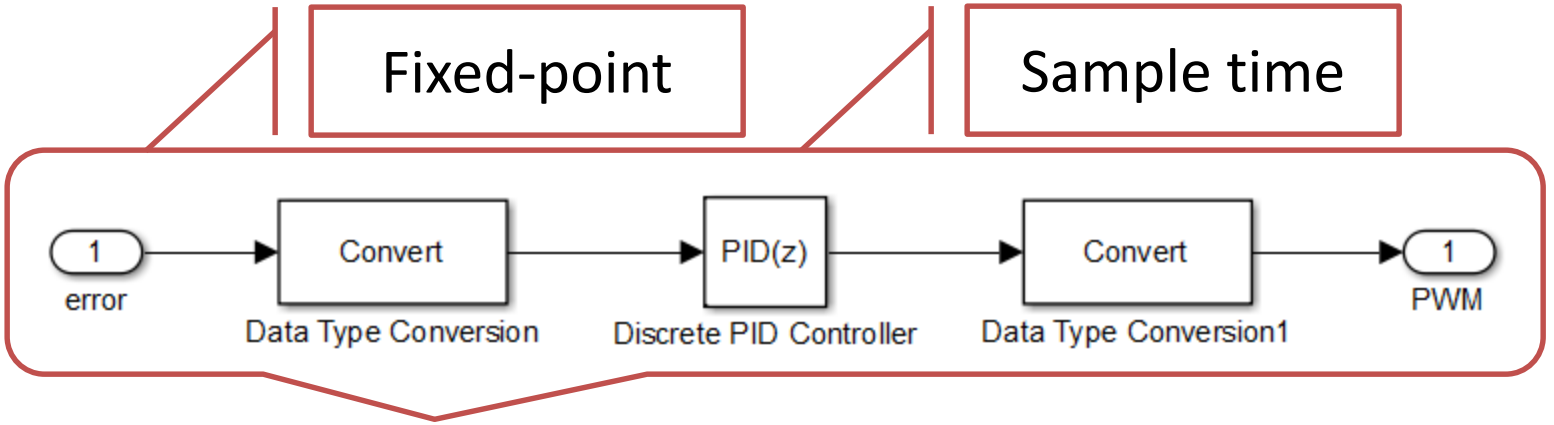


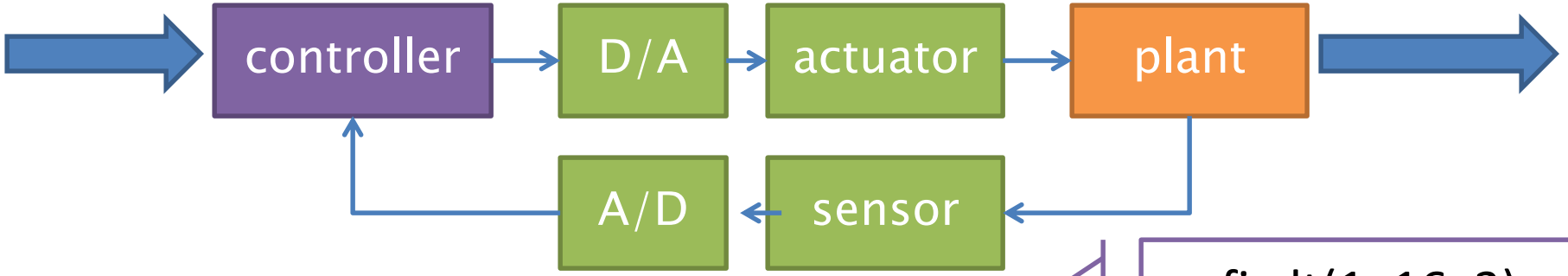
EINT





Discretization effects



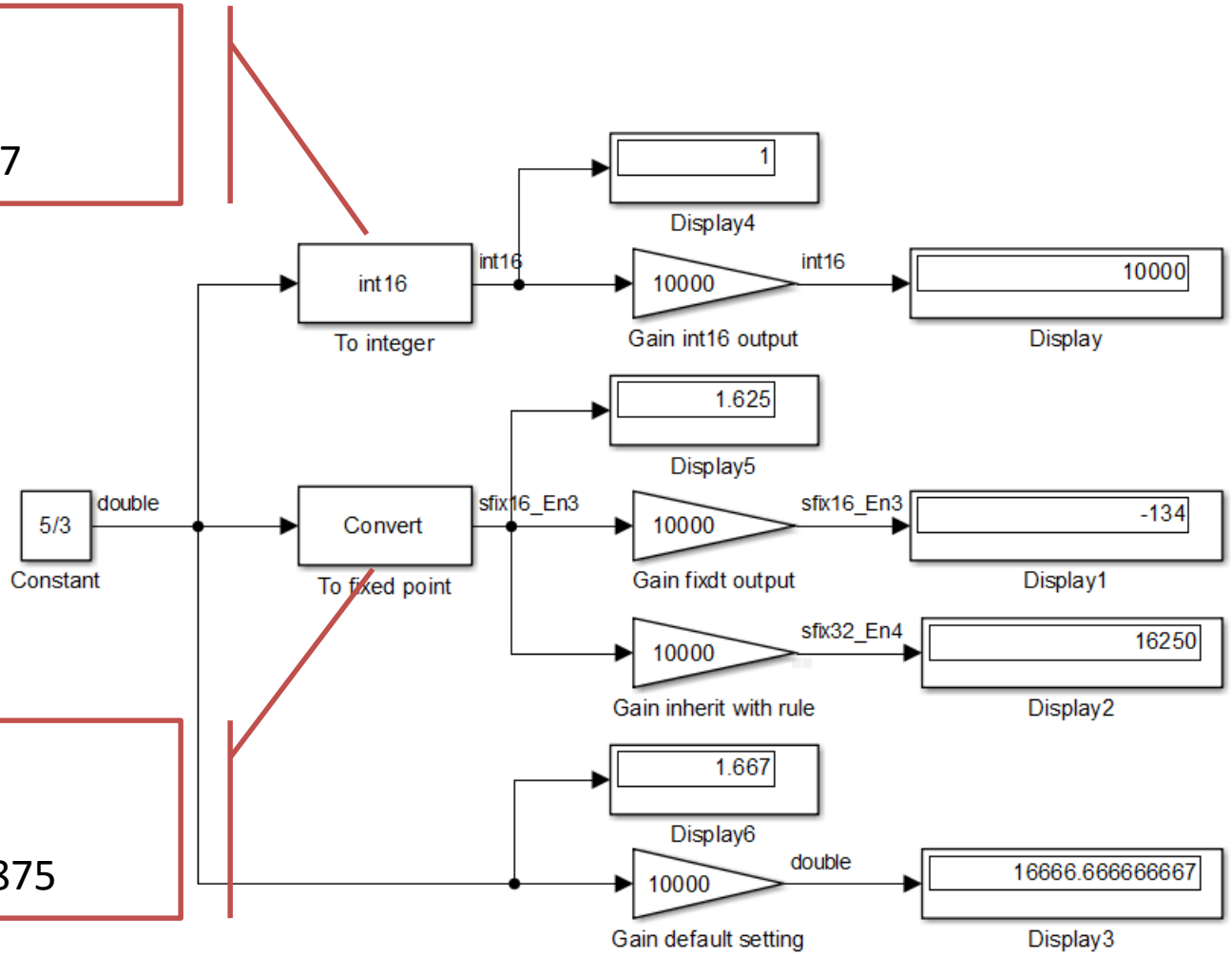


fixdt(1, 16, 3)

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau \qquad u(k) = K_p e(k) + K_i T_s \sum_{i=0}^{k} e(i)$$

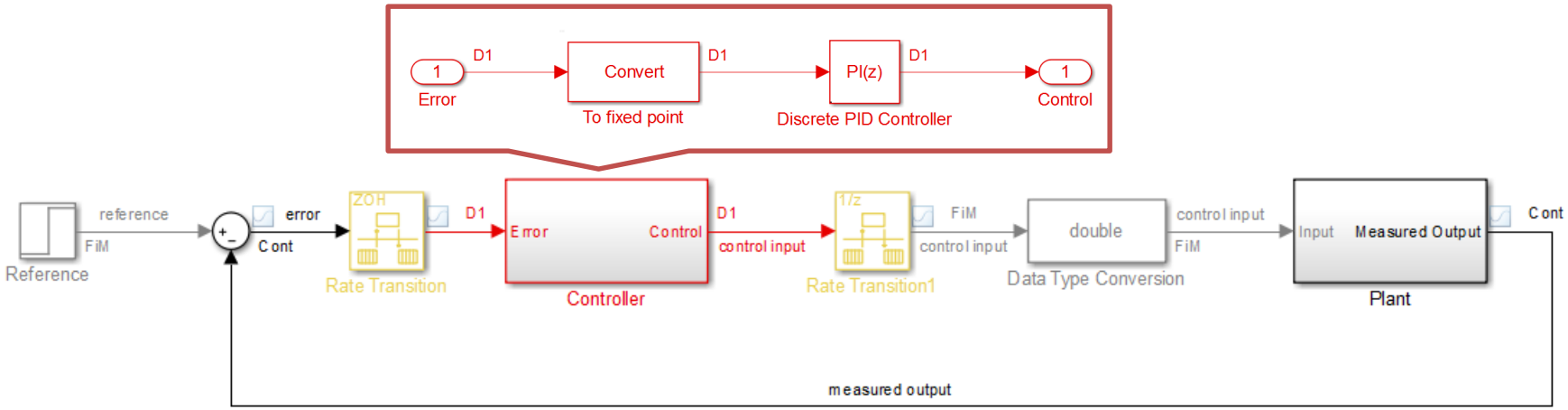
Fixed-point arithmetic

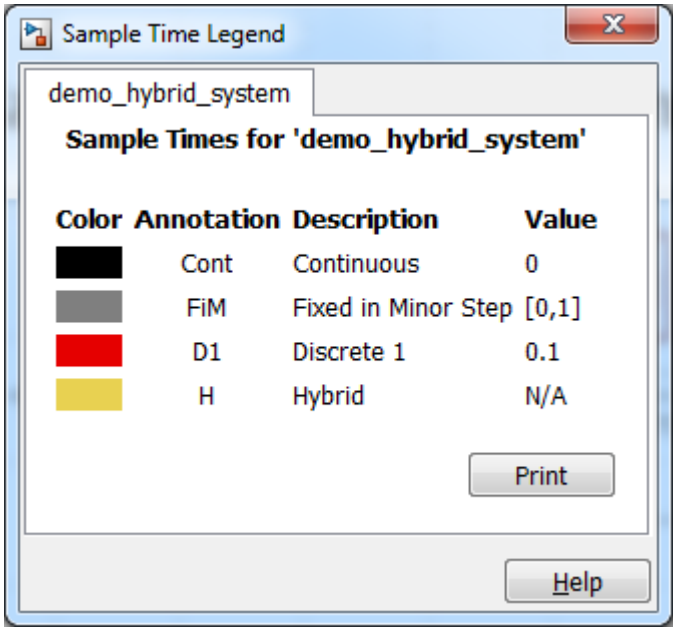
16-bit integer
resolution: 1
range: -32768 ~ 32767

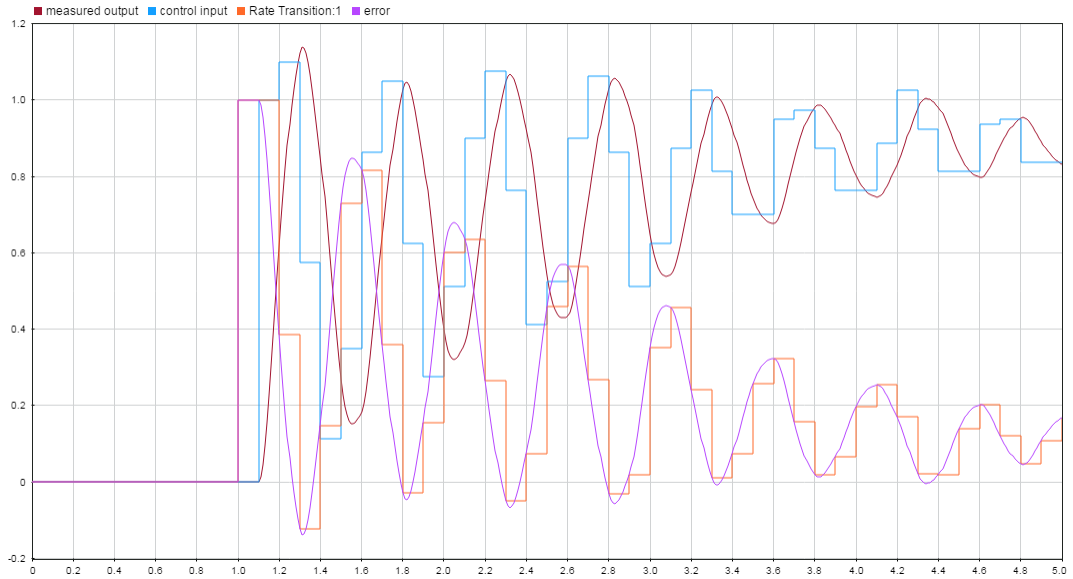


fixdt(1, 16, 3)
resolution: 0.125
range: -4096 ~ 4095.875

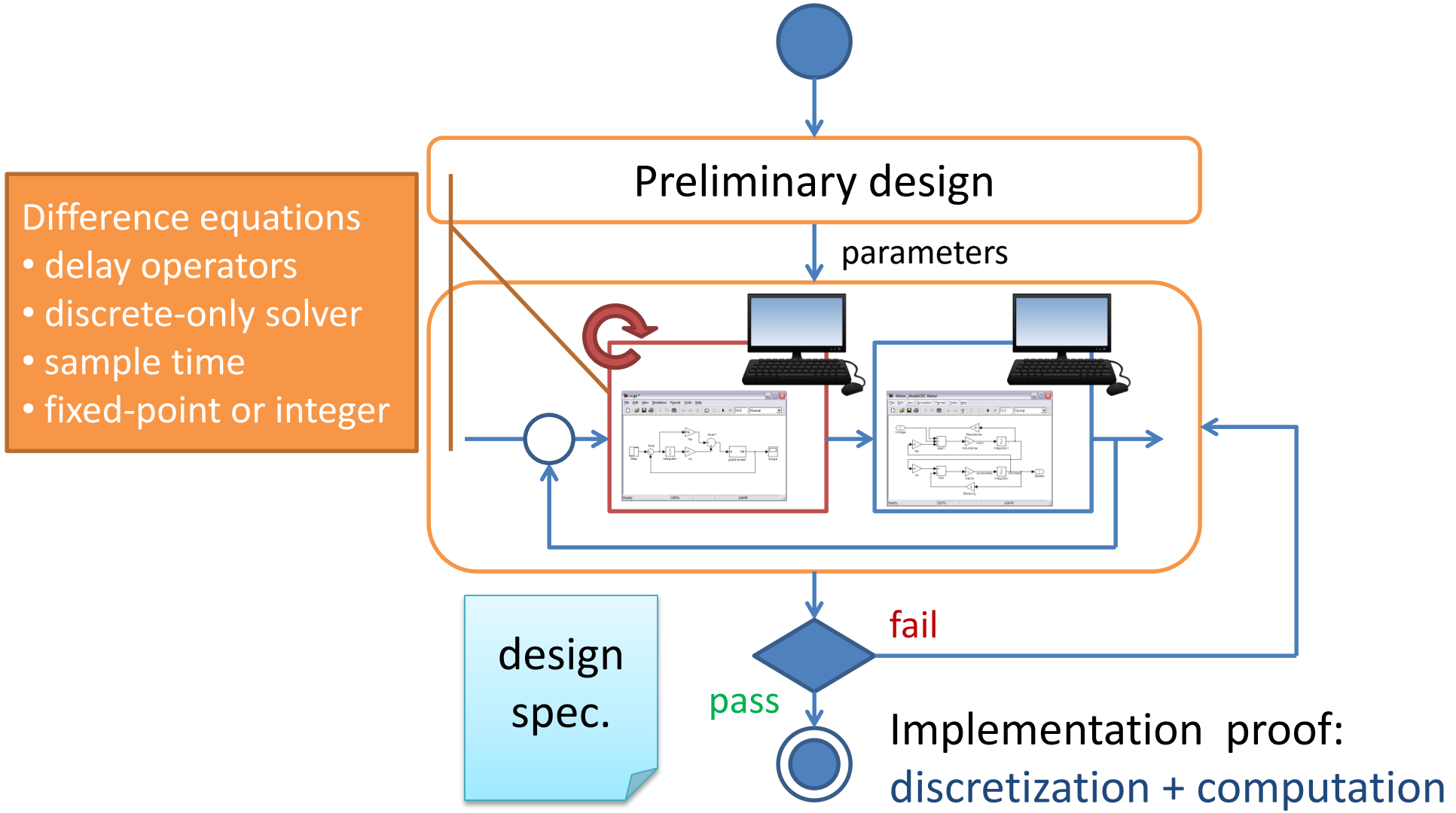
Simulation of hybrid systems

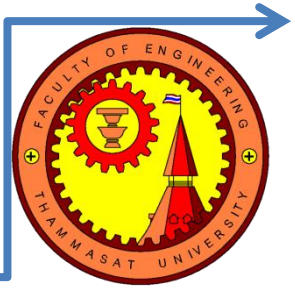






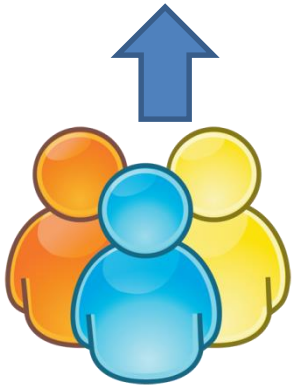
Software-In-Loop Simulation

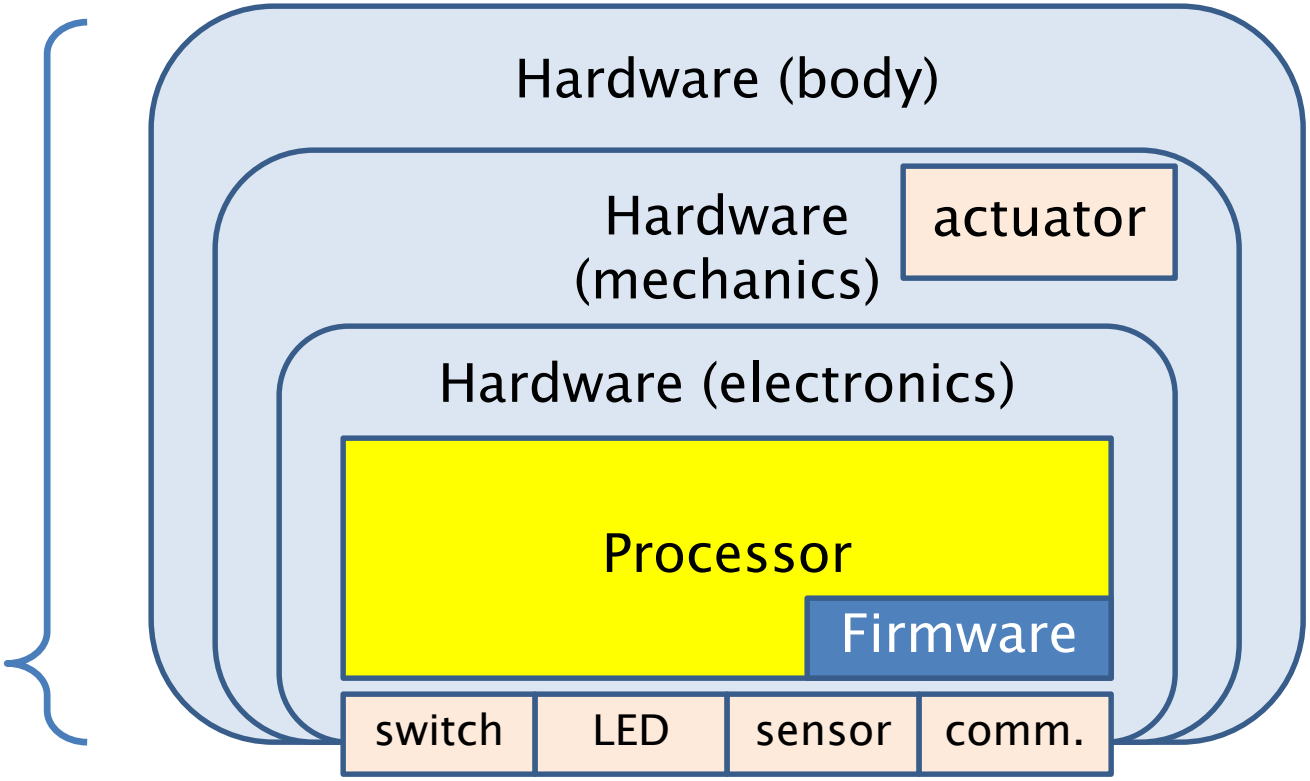




EMBEDDED HARDWARE

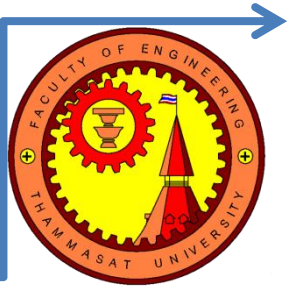
Embedded hardware





Nature | Timing | Constraint | Reliability

Embedded processor



System cost

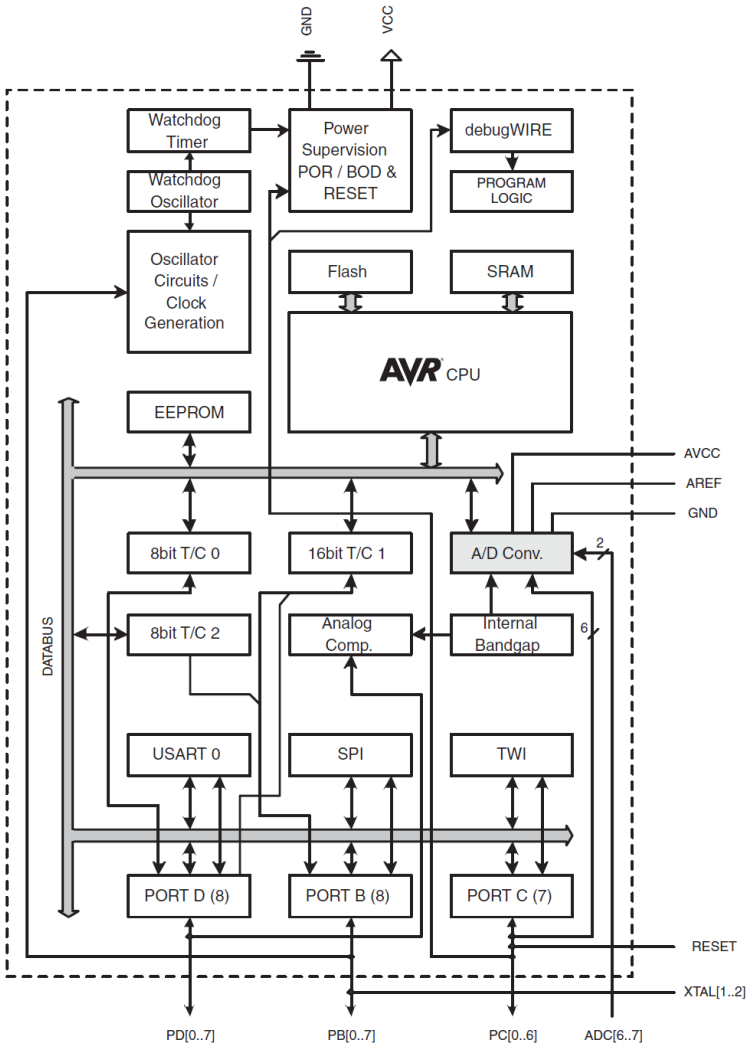
Production volume

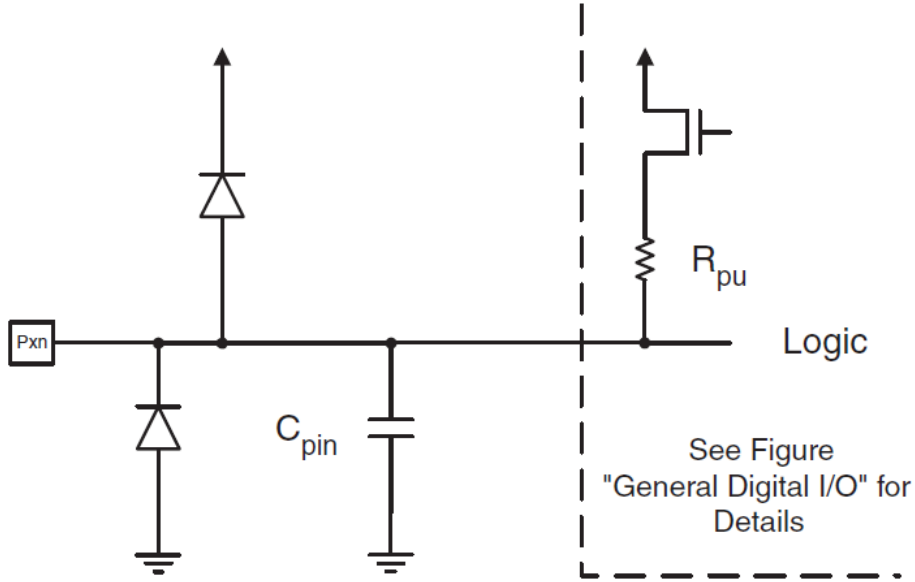
Digital Signal Processor
Microprocessor

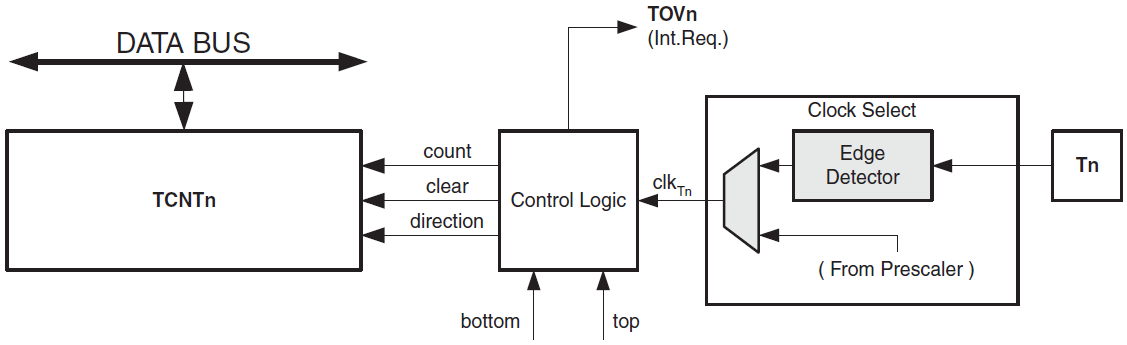
Microprocessor
System-on-Chip

Microcontroller
- Processor + memory + peripherals + I/O
- Limited computing resources
- Cost effective

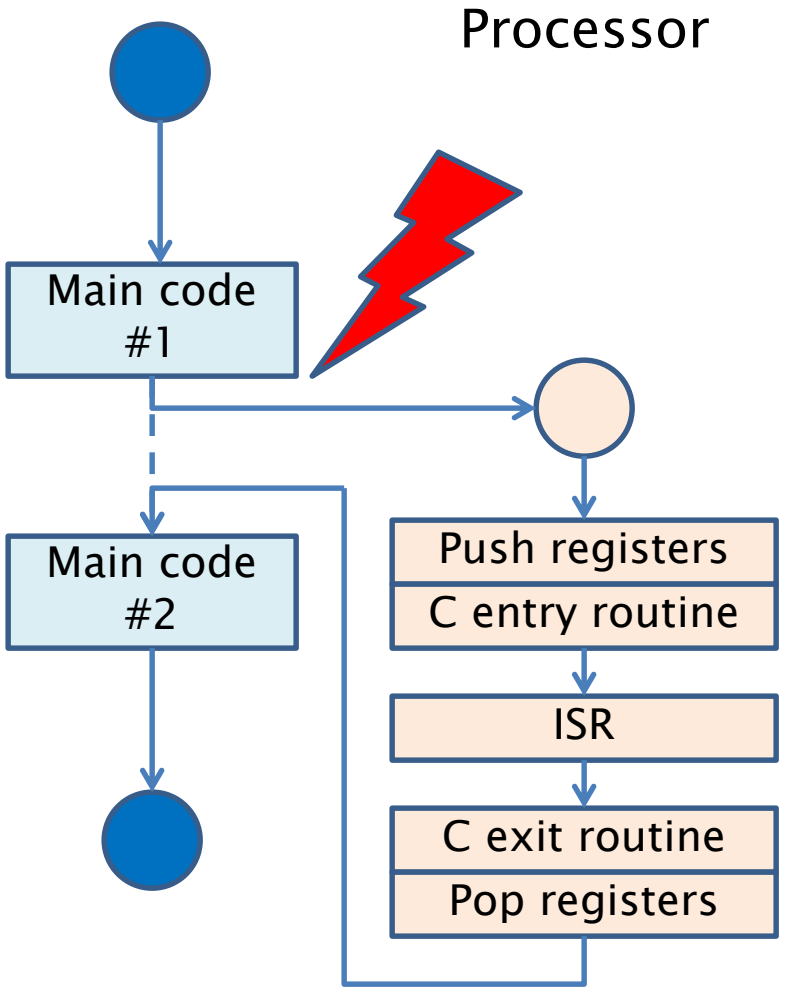
Microcontroller

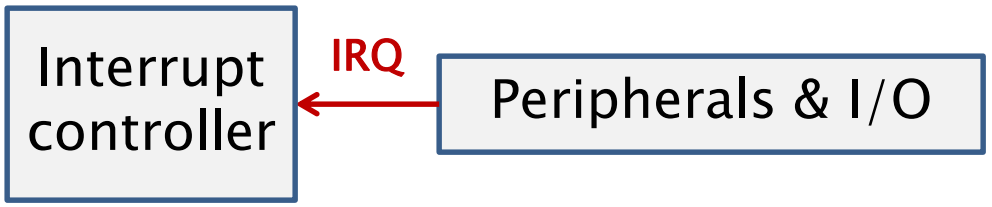






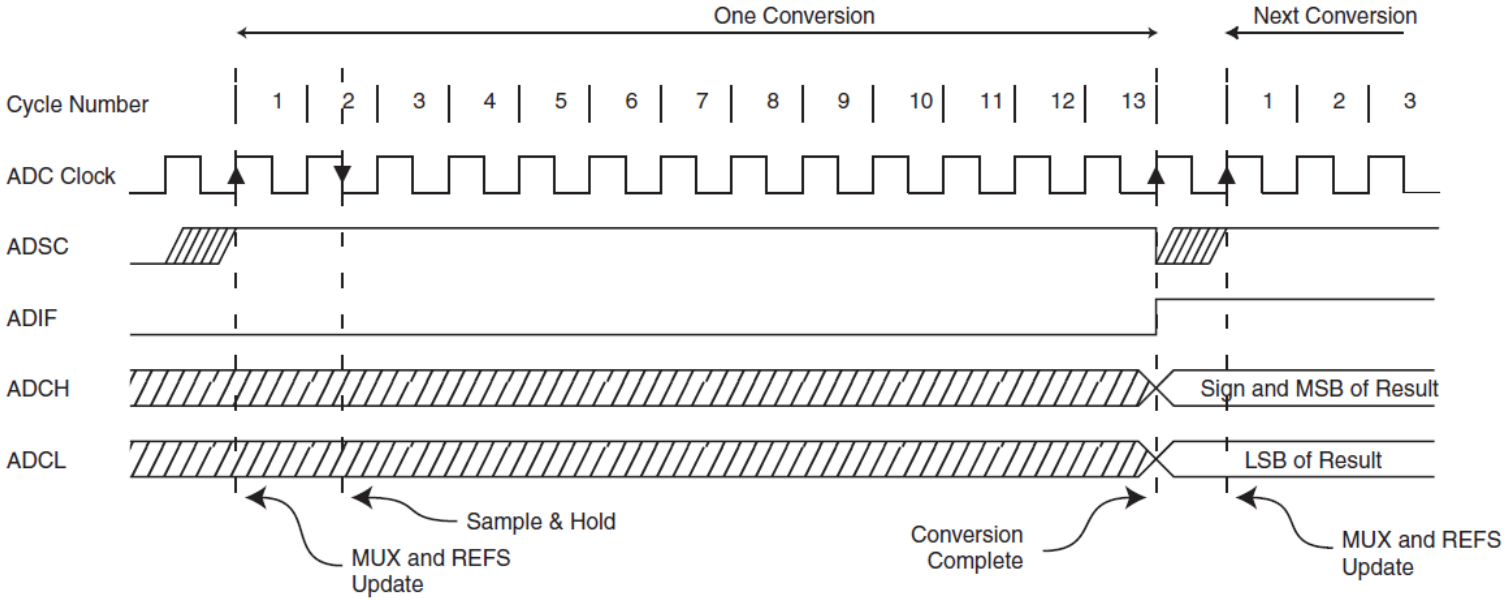
Hardware interrupt

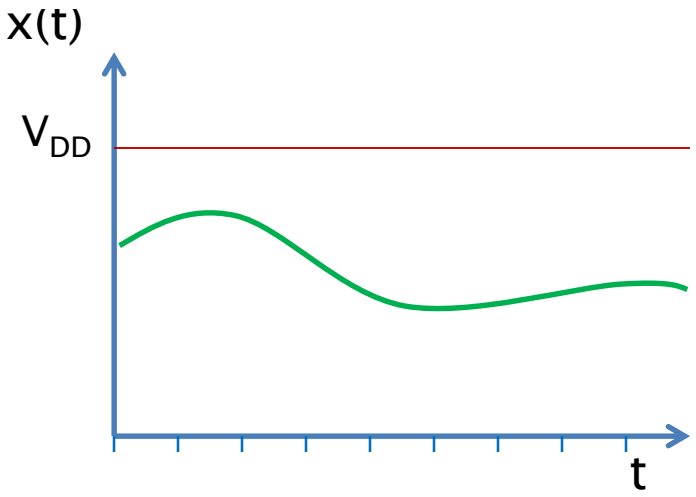


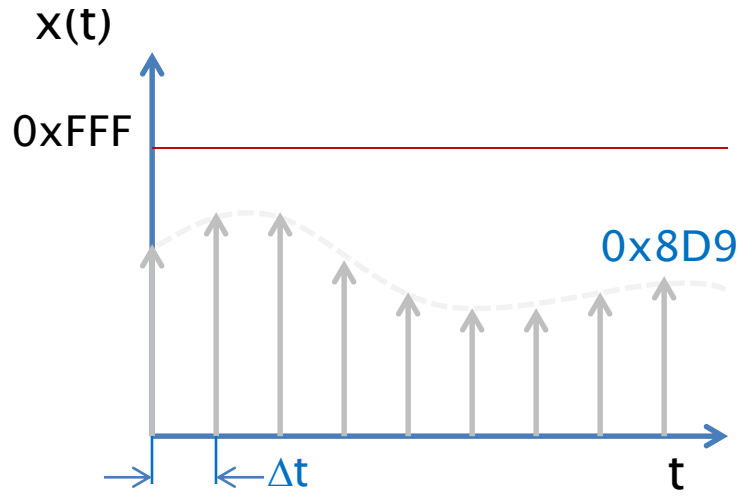


No.	Address	Source
1	0x000	RESET
2	0x001	INT0
3	0x002	INT1
4	0x003	PCINT0
5	0x004	PCINT1
6	0x005	PCINT2
…	…	…
26	0x019	SPM READY

Real-world interfaces

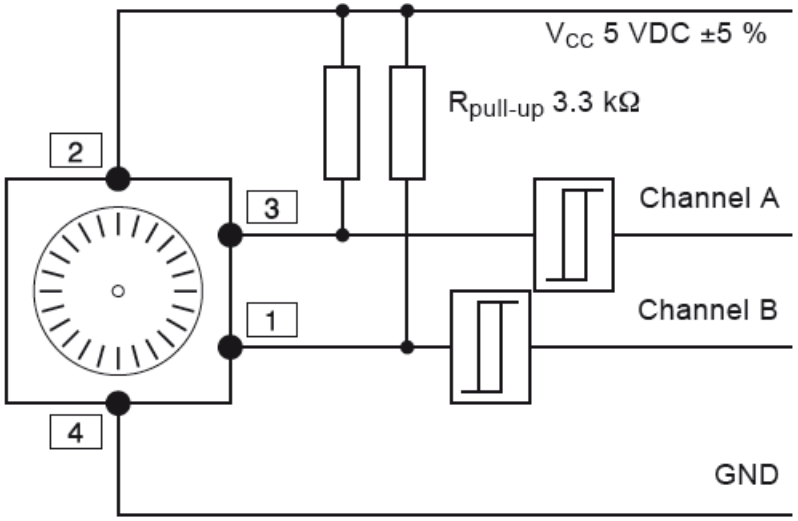


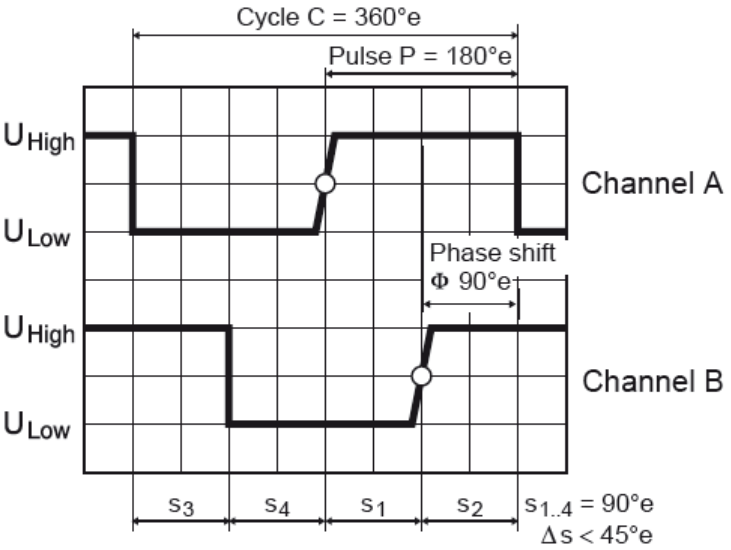


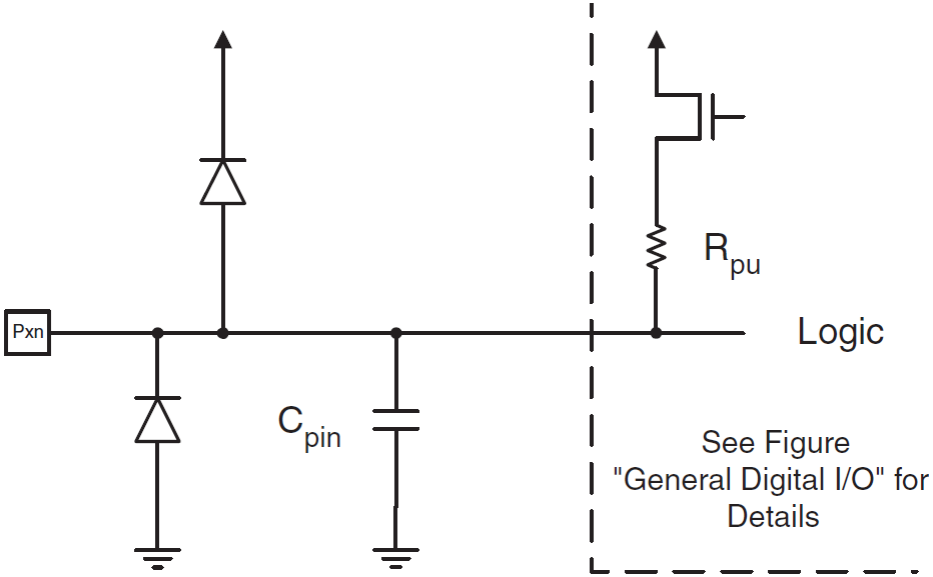


x[0] = 2730
x[1] = 3195
x[2] = 2730
x[3] = 2511
x[4] = 2048
x[5] = 1829
x[6] = 1965
x[7] = 2047
x[8] = 2265

Real-world interfaces

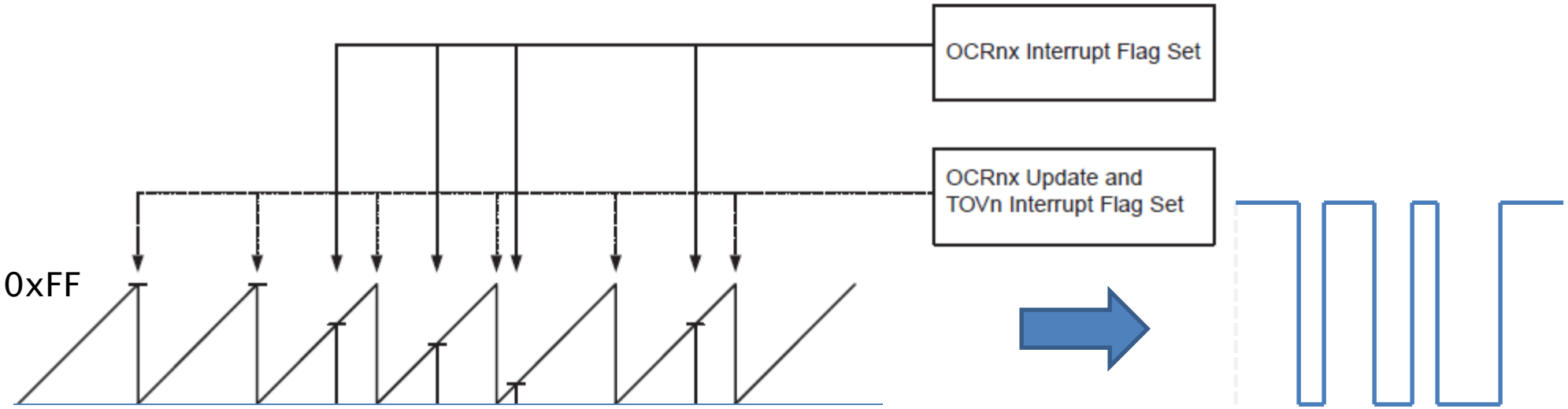


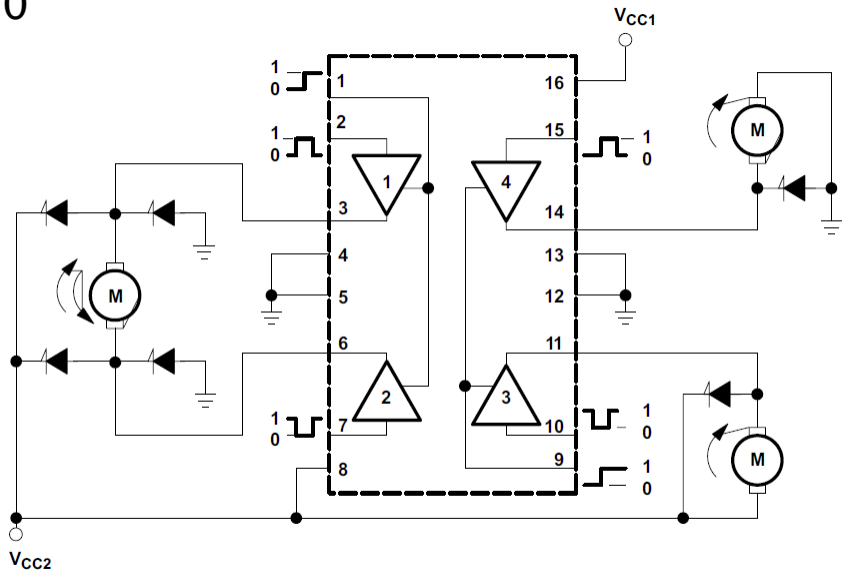


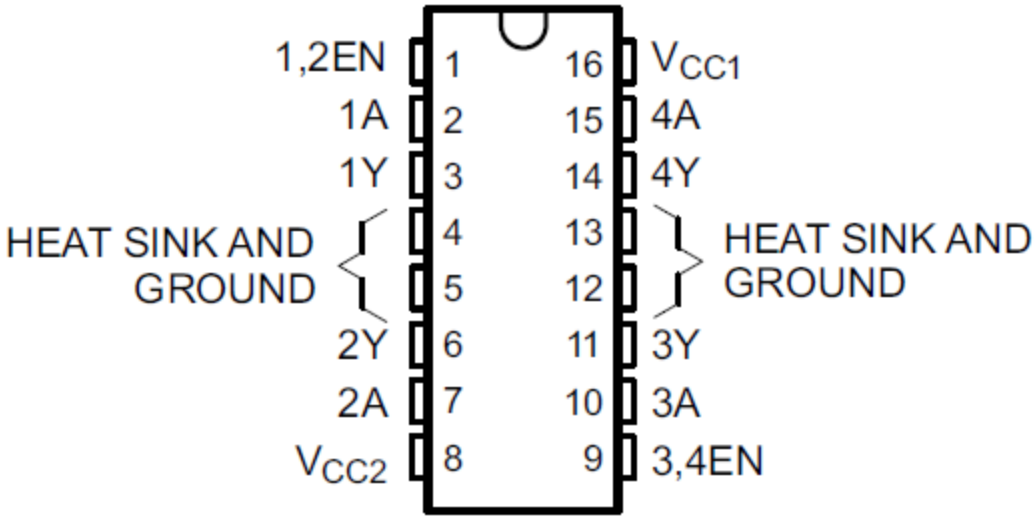


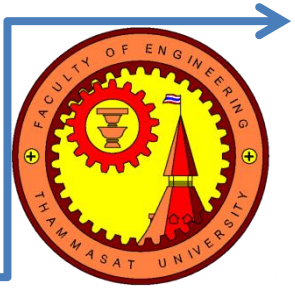
ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

Real-world interfaces







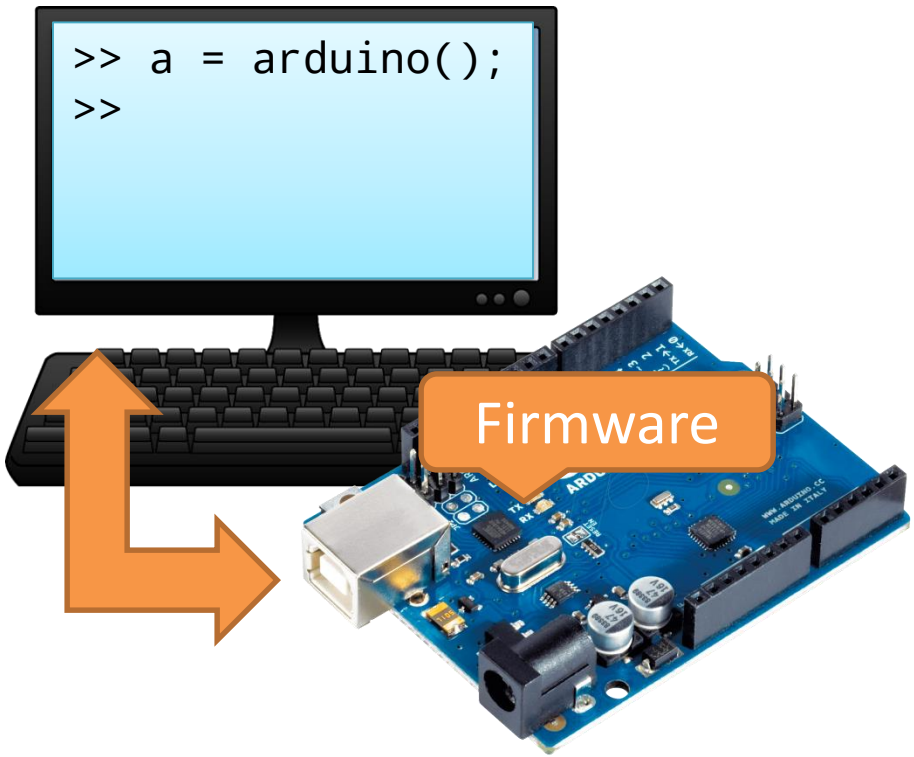


HARDWARE TARGET

Hardware support package

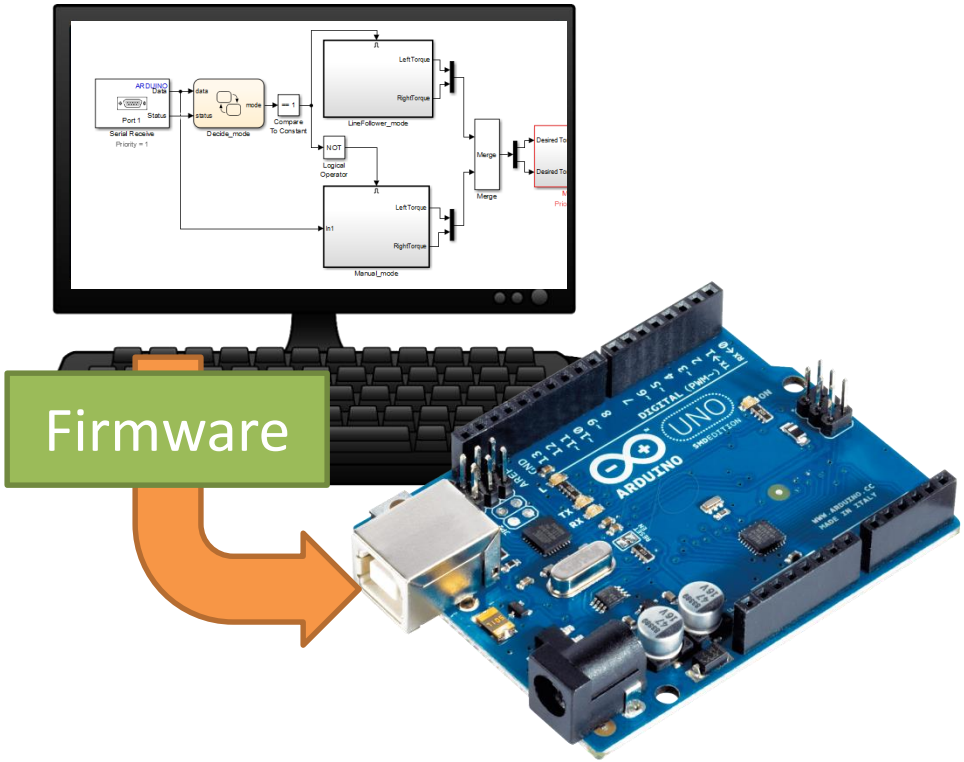
Hardware as I/O

- Receive and send data to hardware

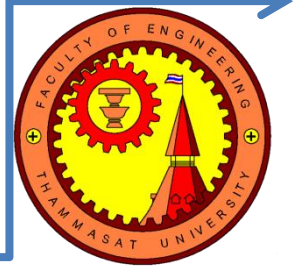


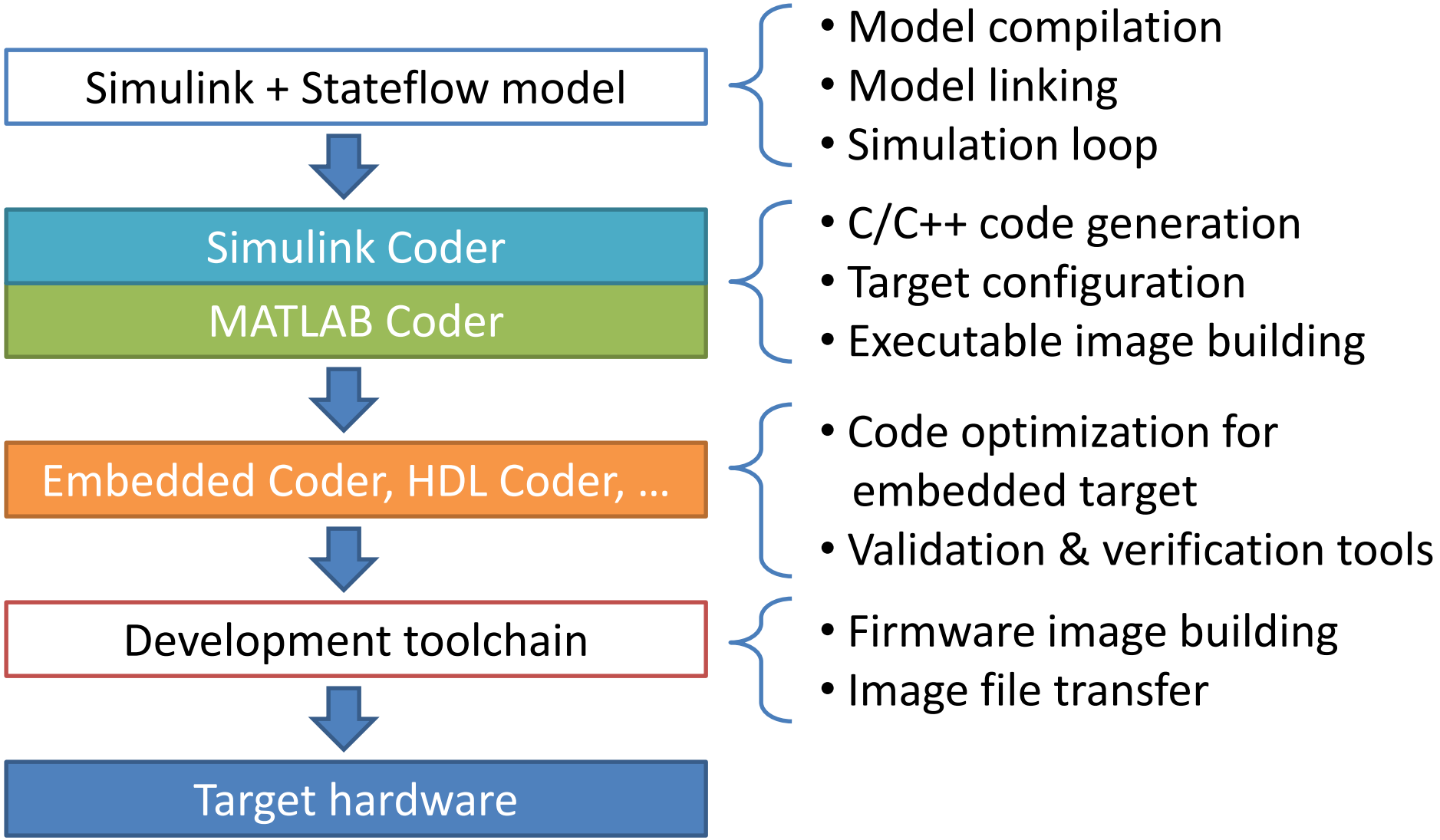
Hardware target

- Generate C code, build and run on hardware

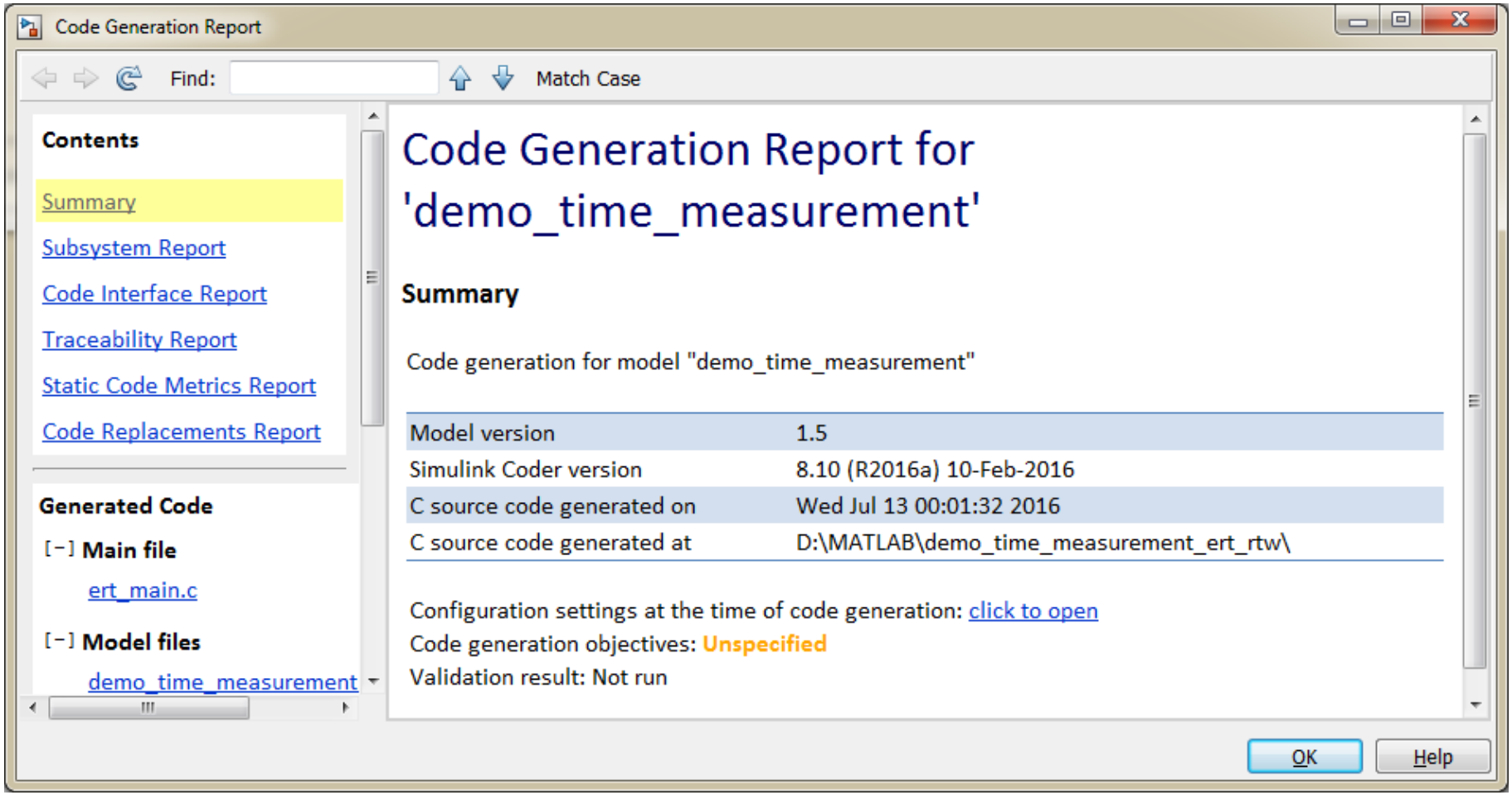


Code generation

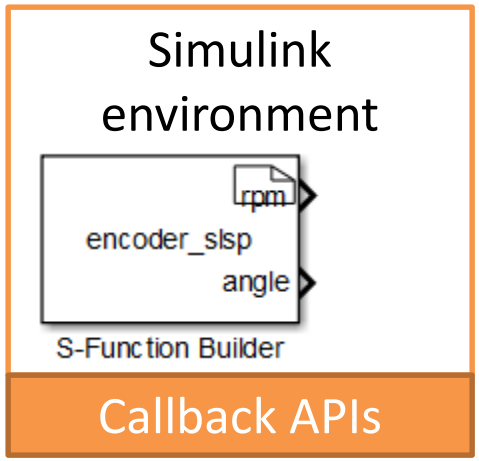




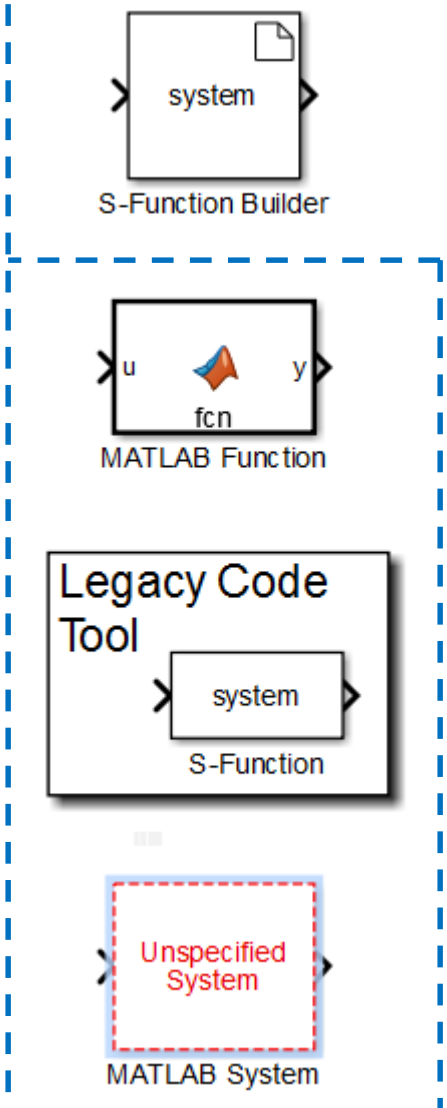
Code generation report



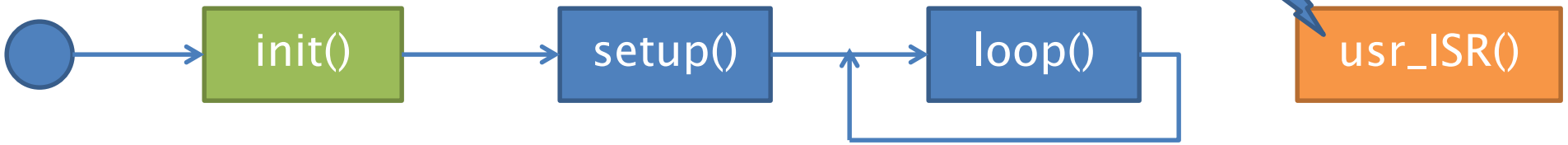
Device driver block

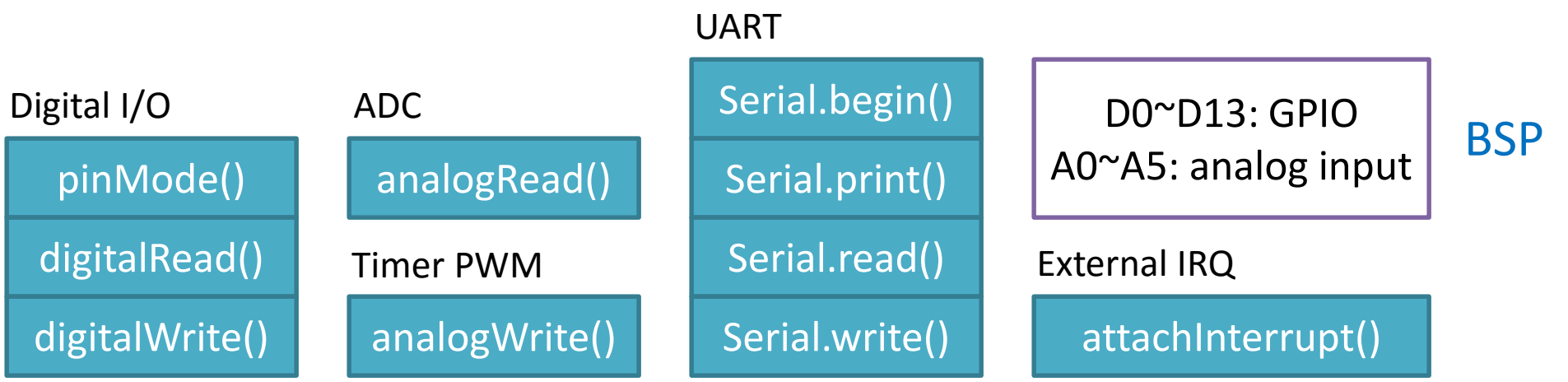


- C code
- Access HW



Arduino library

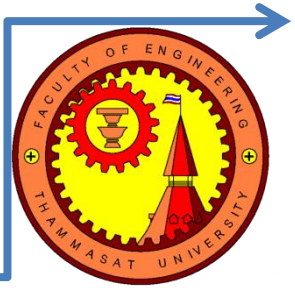




14.4.2 PORTB – The Port B Data Register

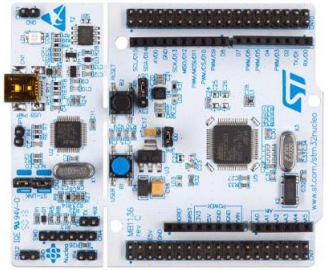
Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

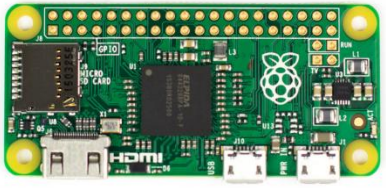
MSP



PERFORMANCE LIMITATIONS

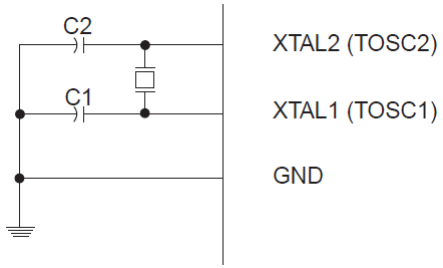
Processor resources

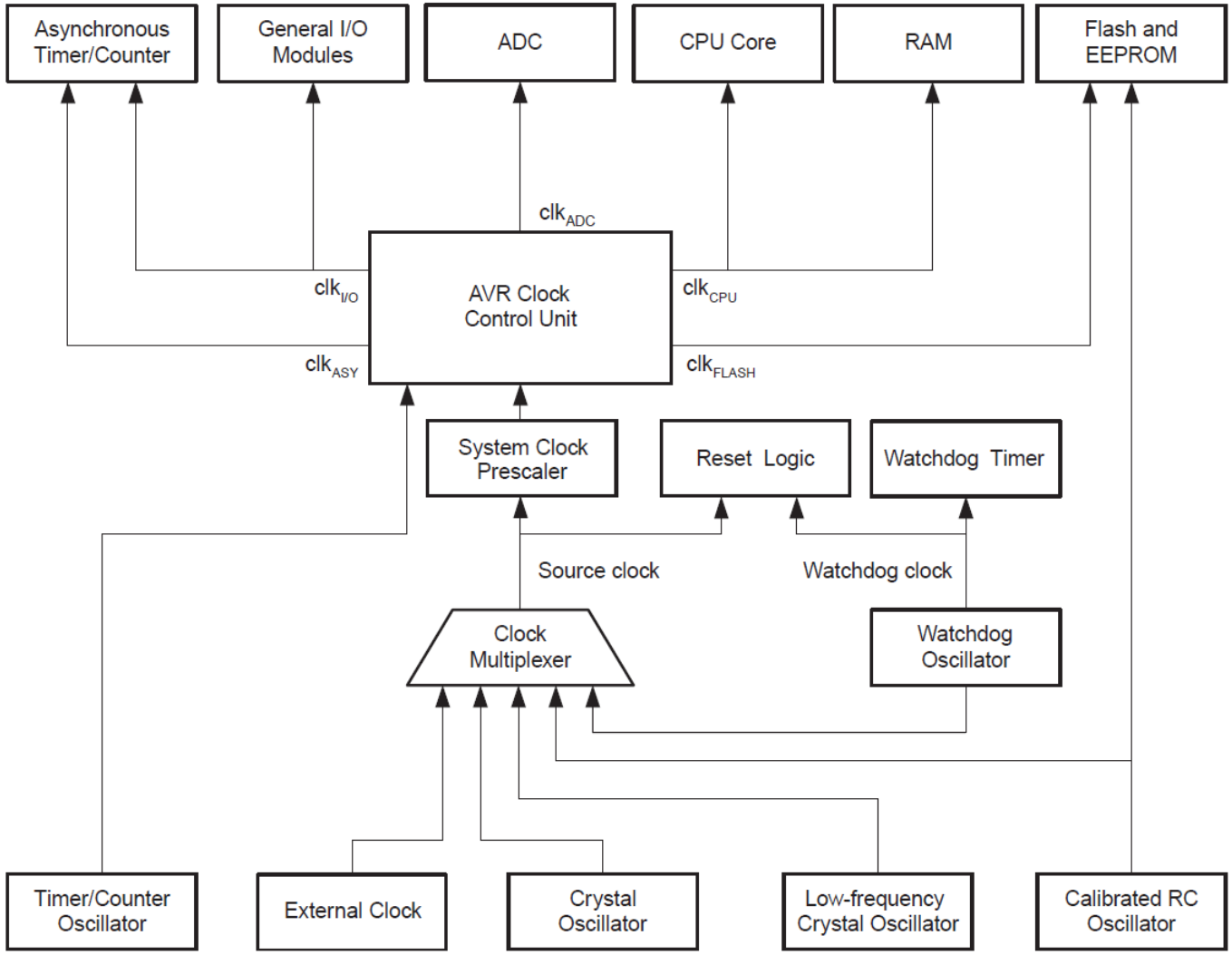




Processor	AVR 8-bit	ARM 32-bit	ARM 32-bit
Clock	16 MHz	84 MHz	1 GHz
Program	32 kB Flash	512 kB Flash	SD card
RAM	2 kB SRAM	96 kB SRAM	512 MB DRAM
I/O pins	20 GPIO pins	50 GPIO pins	26 GPIO pins
Peripherals	UART, I2C, SPI, ADC, PWM	UART, I2C, SPI, ADC, PWM, USB	UART, I2C, SPI

Clock system

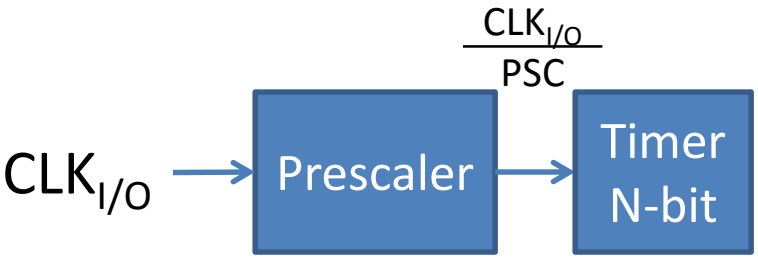




Time-triggered systems

Wikipedia

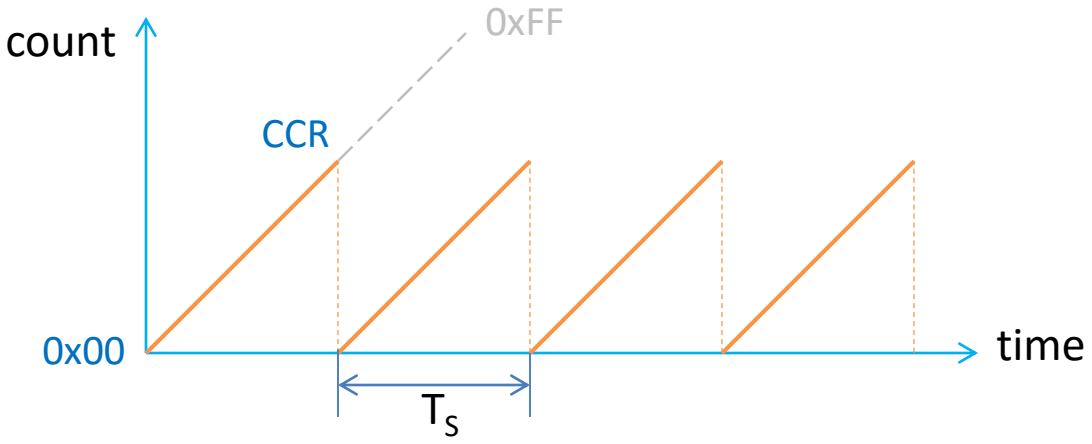
A computer system that executes one or more sets of tasks according to a pre-determined task schedule.

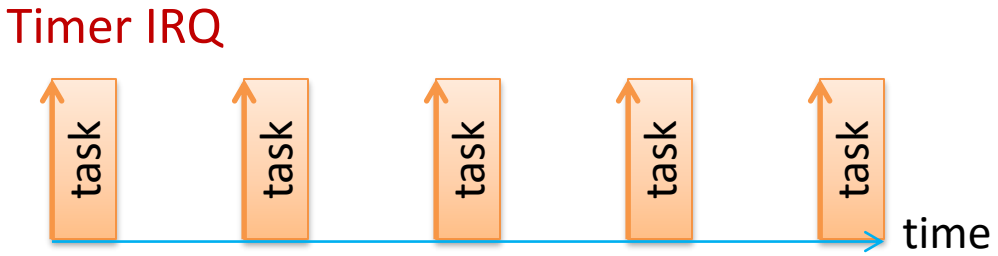


$$\text{Resolution} = \frac{PSC}{CLK_{I/O}}$$

$$\text{Range} = 2^N \times \frac{PSC}{CLK_{I/O}}$$

$$\text{Period} = CCR \times \frac{PSC}{CLK_{I/O}}$$



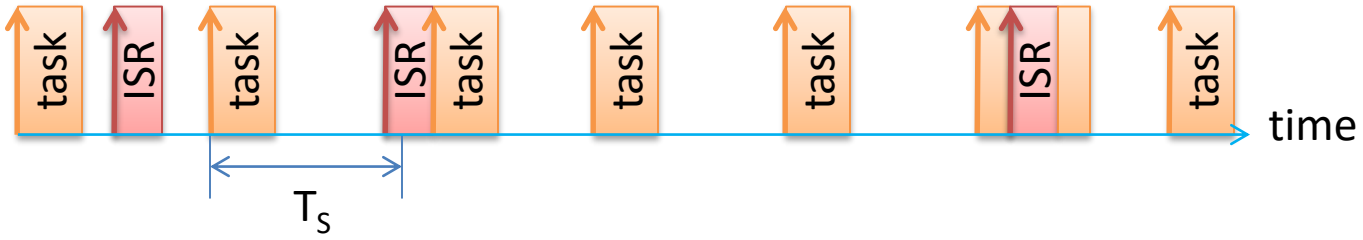


Real-time computing

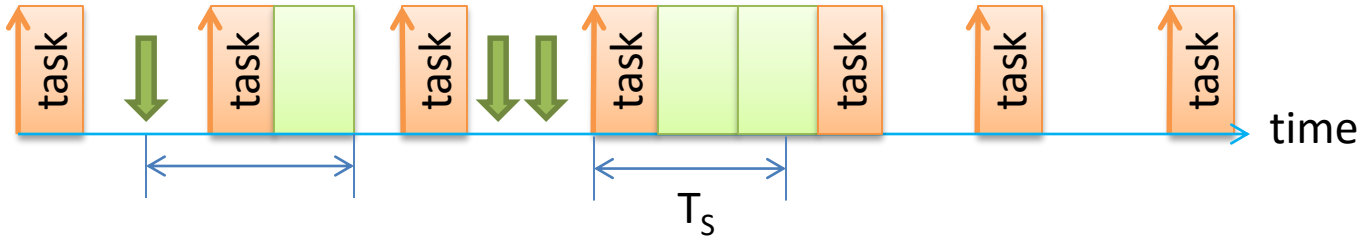
Wikipedia

Hardware and software systems subject to real-time constraints, often referred to as "deadlines".

Timing accuracy

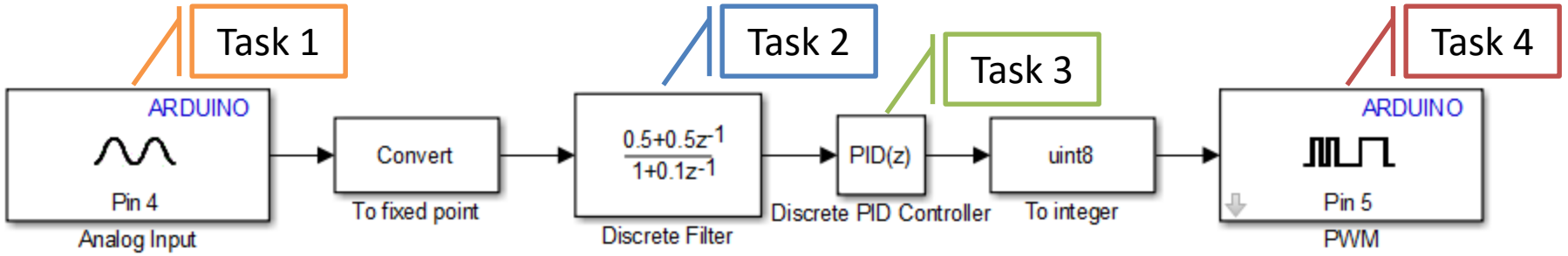


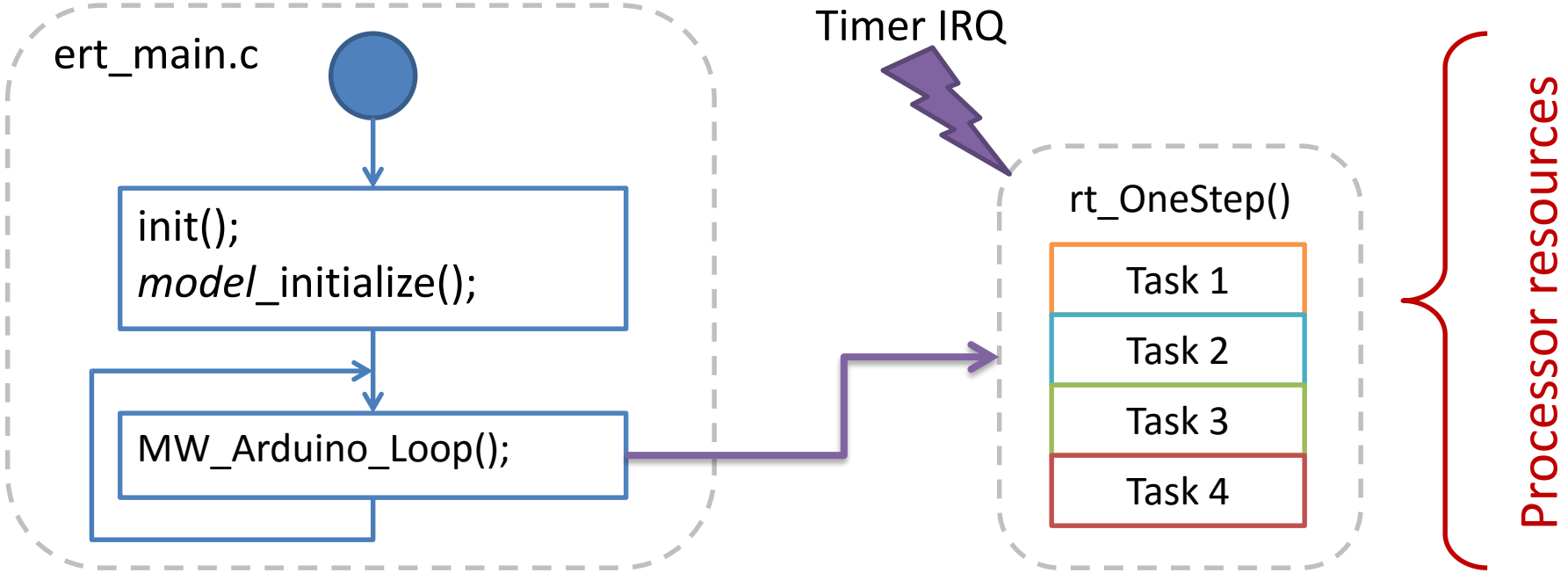
Response time



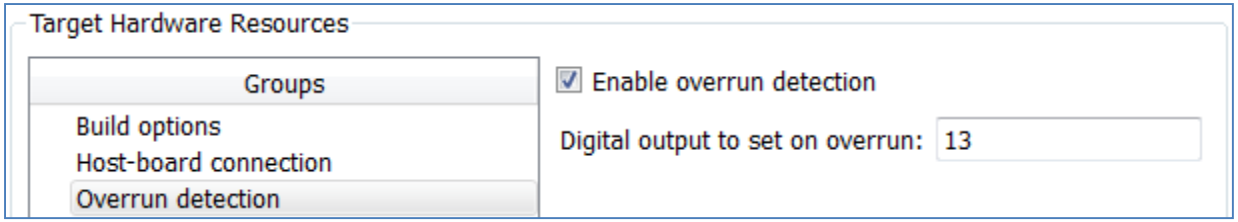
Nature	Timing	Constraint	Reliability

Runtime operations





Overrun detection




```
#ifdef _RTT_OVERRUN_DIGITAL_PIN_
if (IsrOverrun == 1) { // Overrun detected
  digitalWrite(_RTT_OVERRUN_DIGITAL_PIN_, HIGH);
}
#endif
rt_OneStep();
```

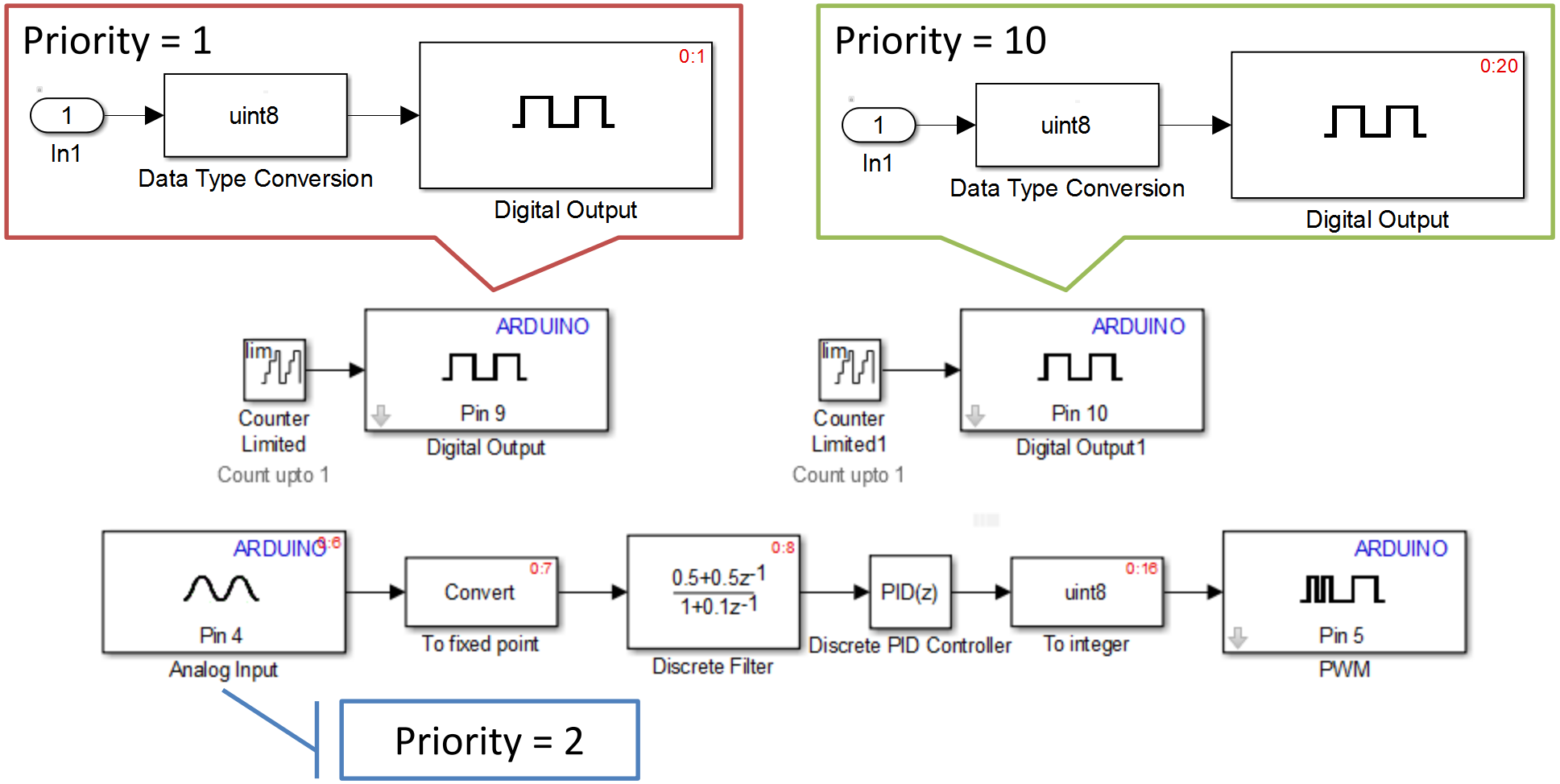
MW_Arduino_Loop()

```
if (OverrunFlag++) {
  IsrOverrun = 1;
  OverrunFlag--;
  return;
}
...
OverrunFlag--;
```

rt_OneStep()

Trigger: NORMAL
Level: 1V ~ 3V
Slope: Rising

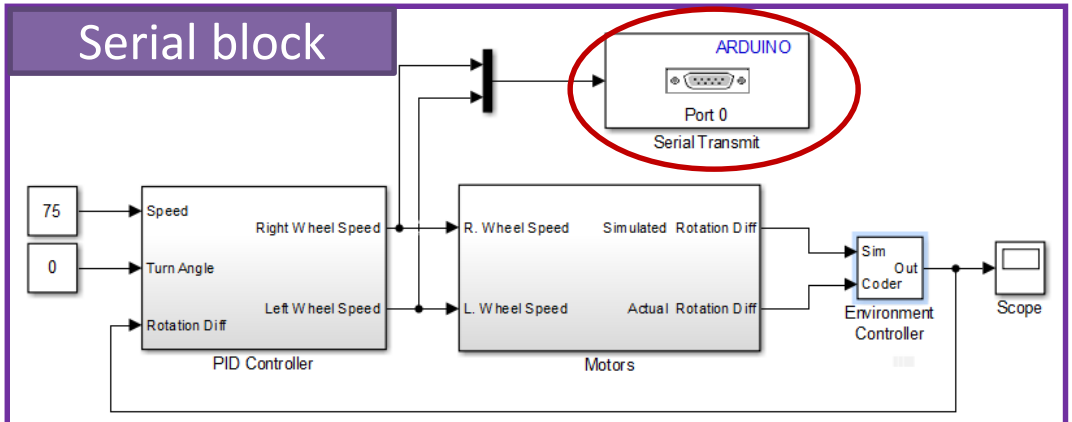
Software timing



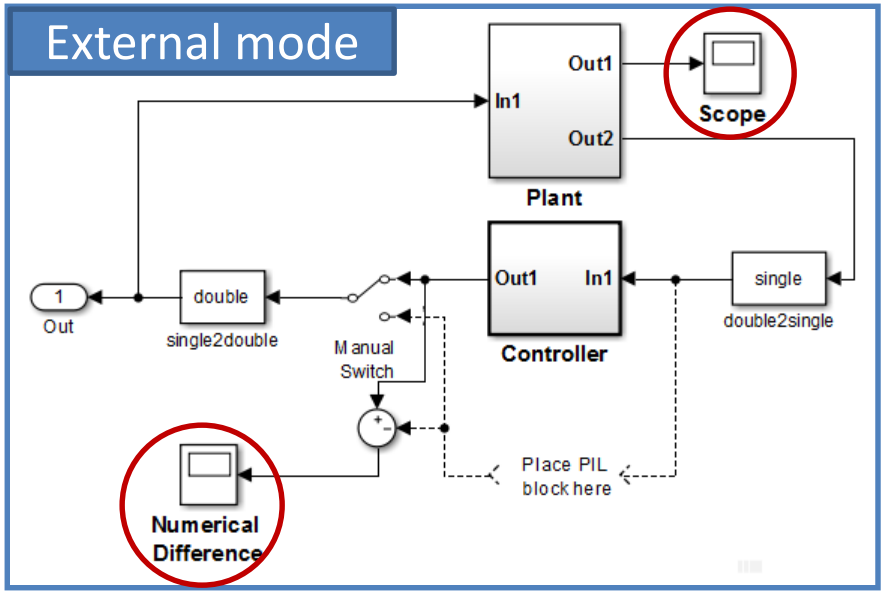
Status monitoring

USB serial

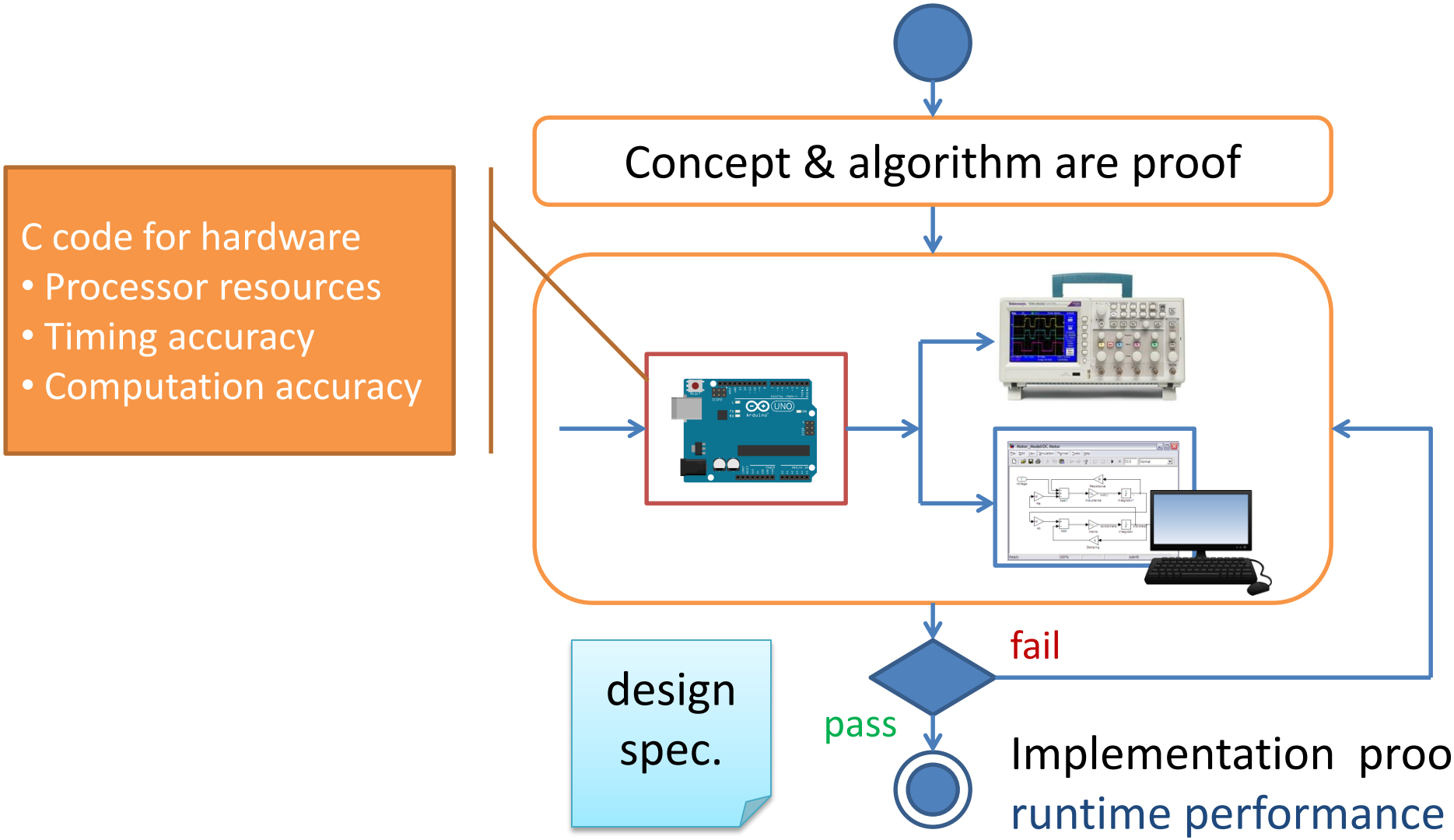
Serial block

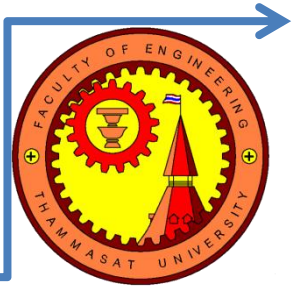


External mode



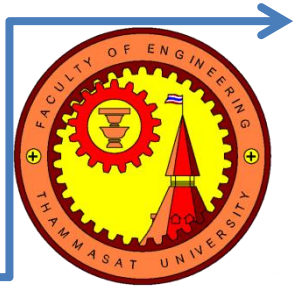
Processor-In-Loop Simulation





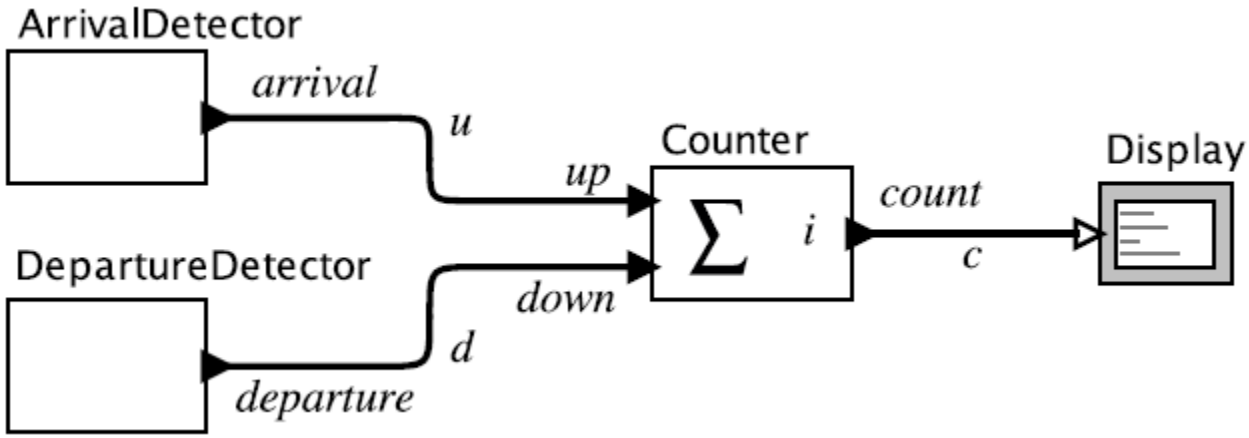
REACTIVE SYSTEMS

Discrete dynamics



Intro to Embedded Systems

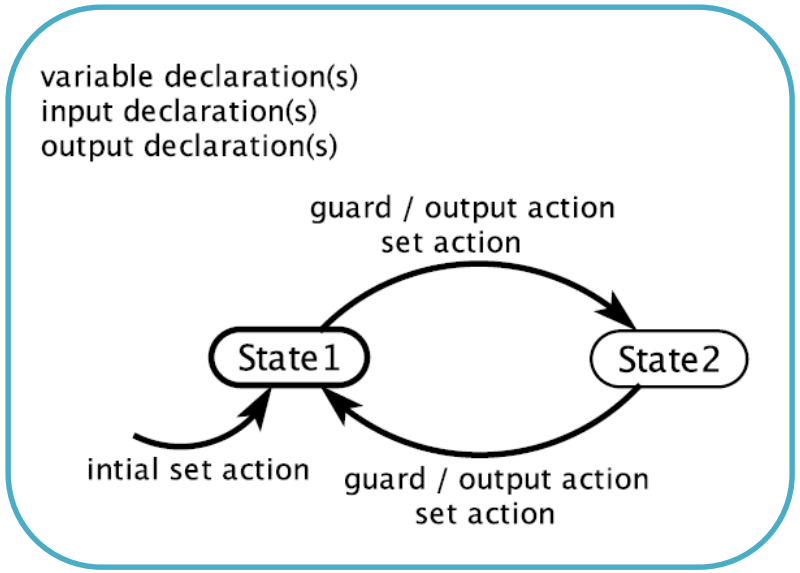
Discrete dynamics can be described as a sequence of steps that we call reactions.
Reactions of a discrete system are triggered by the environment, i.e. by events.
The state of a system is its condition at a particular point in time.



Finite state machine

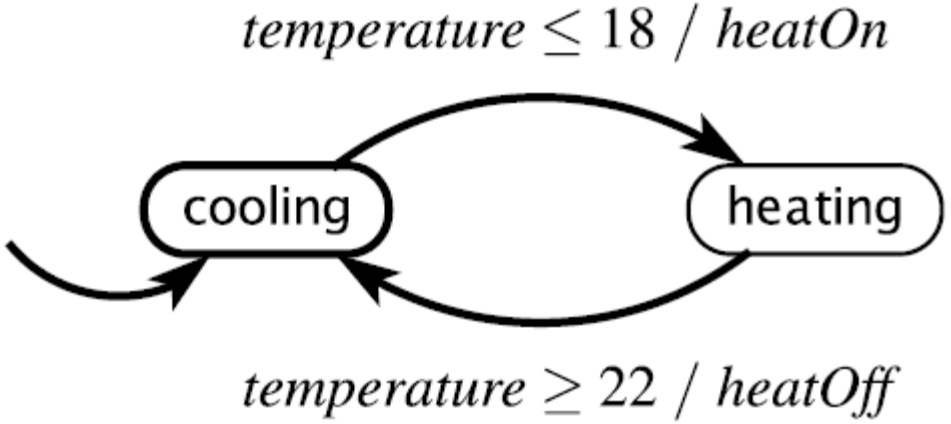
Intro to Embedded Systems

A model of a system with discrete dynamics that at each reaction maps valuations of the inputs to valuations of the outputs, where the map may depend on its current state from a finite set.



input: *temperature* : $\mathbb{R}$

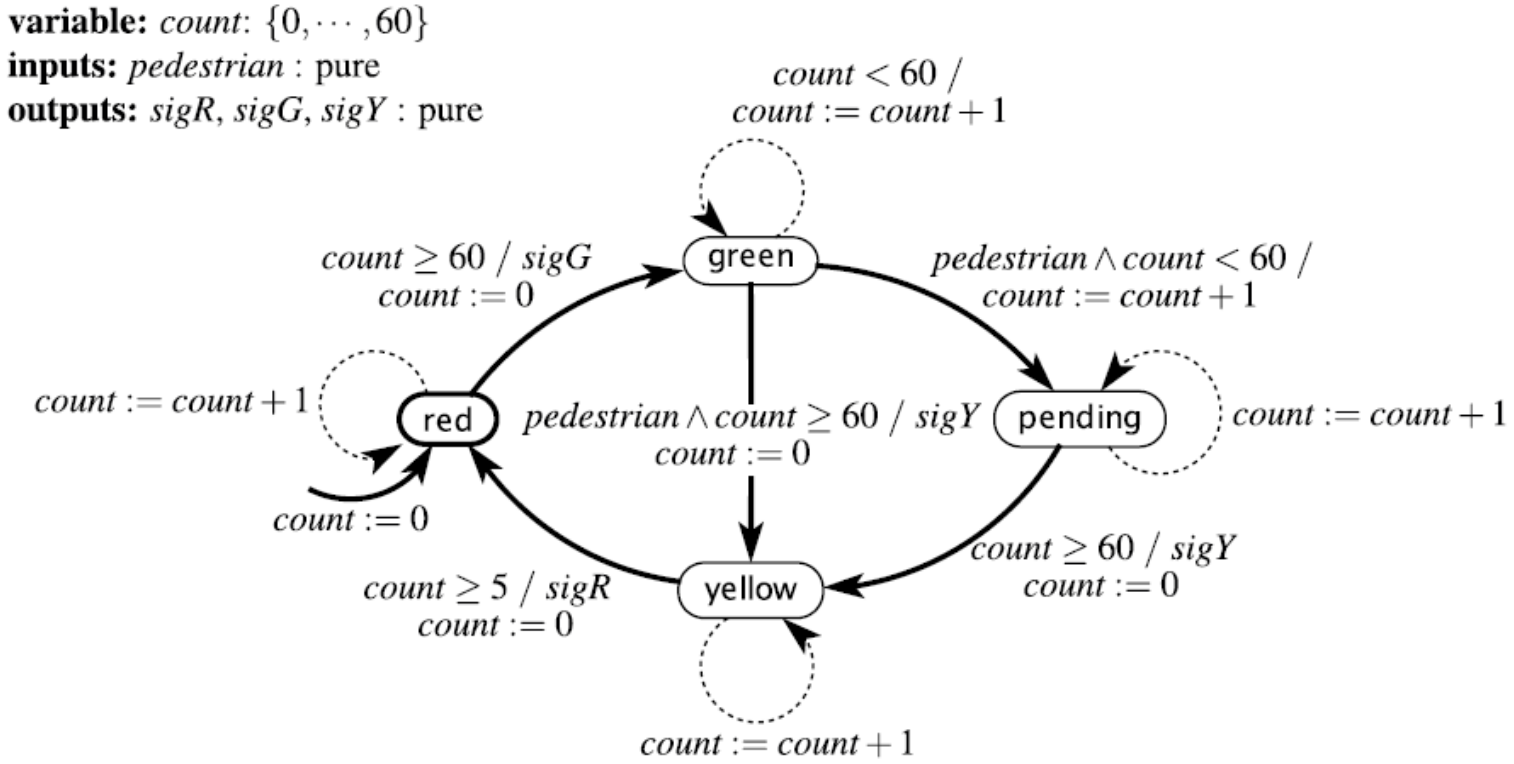
outputs: *heatOn*, *heatOff* : pure



Temporal logic

Wikipedia

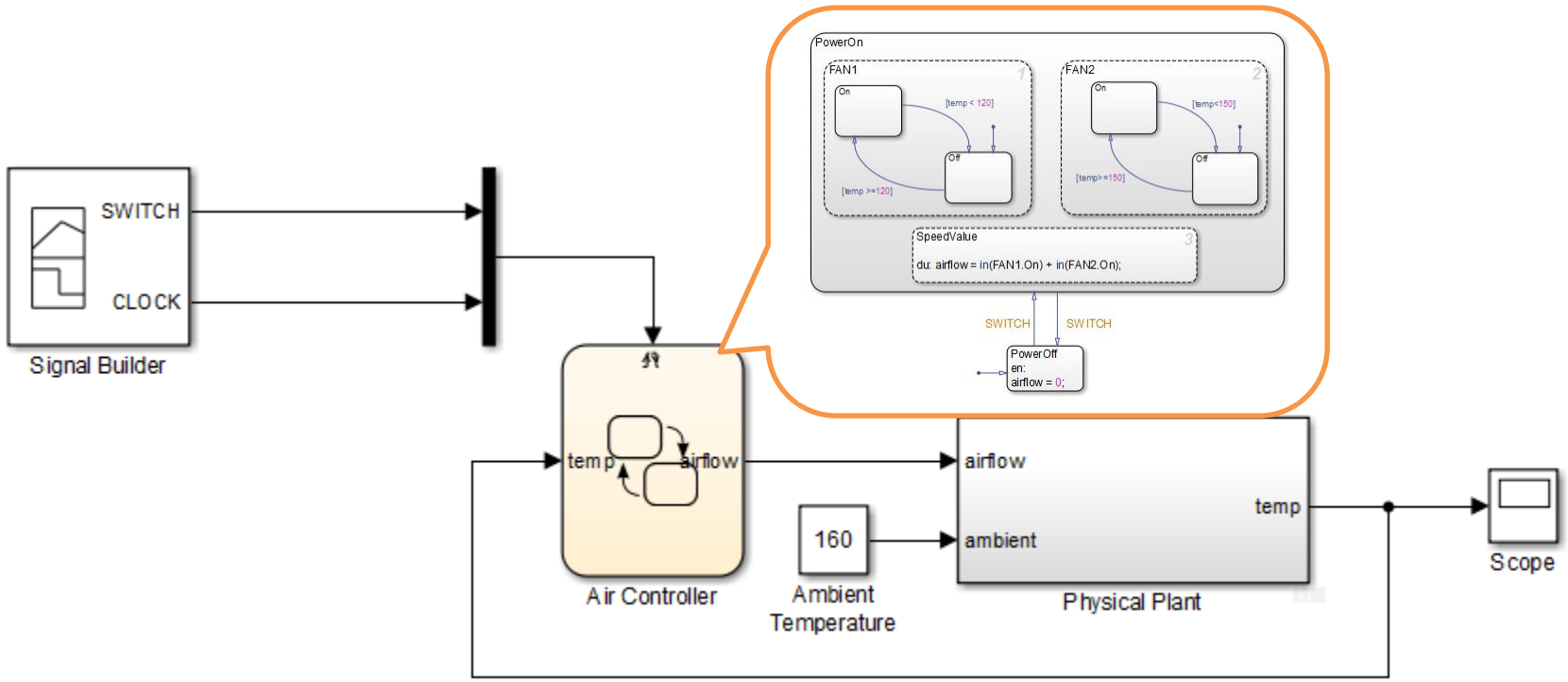
Any system of rules and symbolism for representing, and reasoning about, propositions qualified in terms of time.



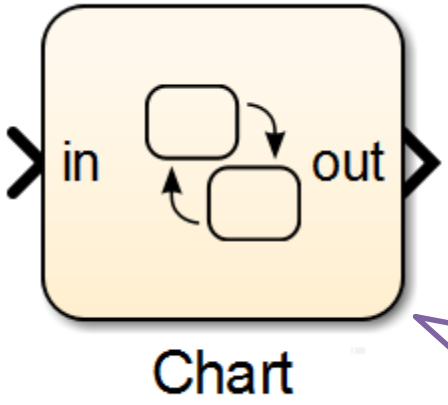
Stateflow

Wikipedia

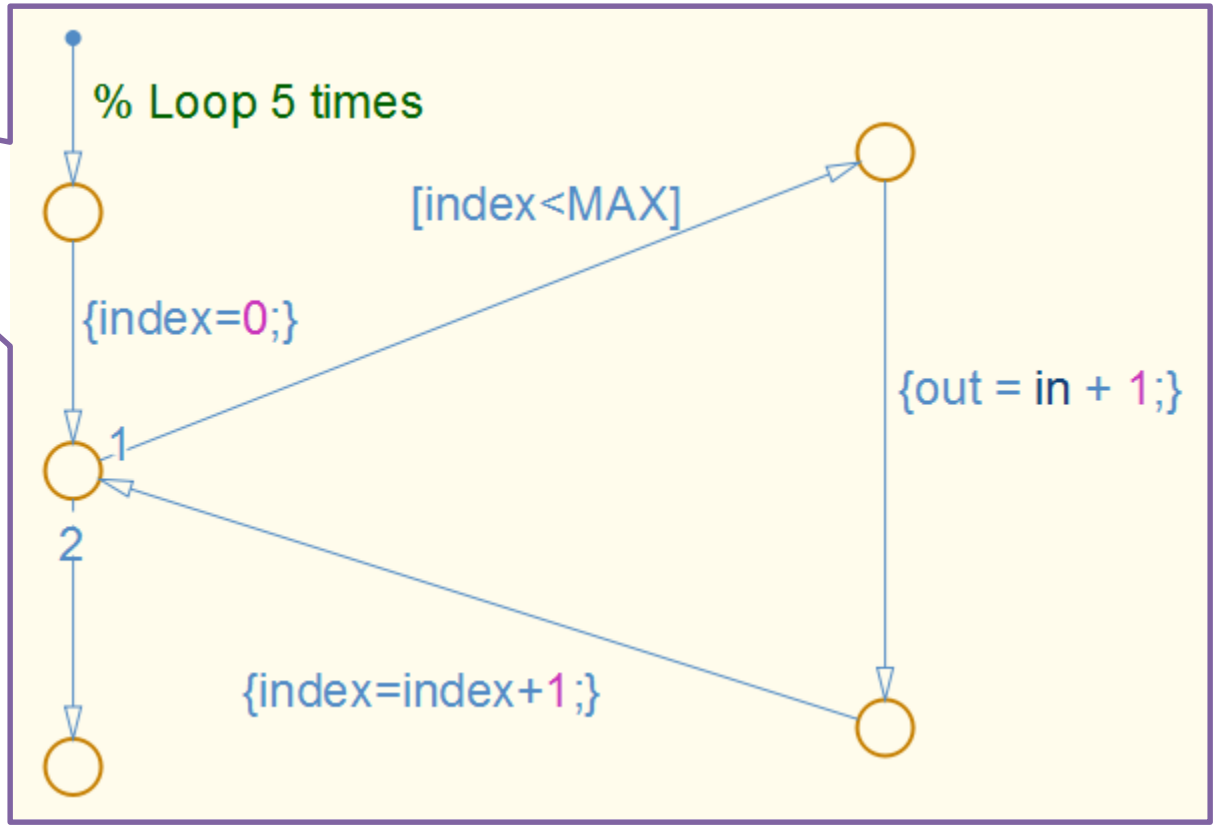
Stateflow is an environment for modeling and simulating combinatorial and sequential decision logic based on state machines and flow charts.



Flow charts



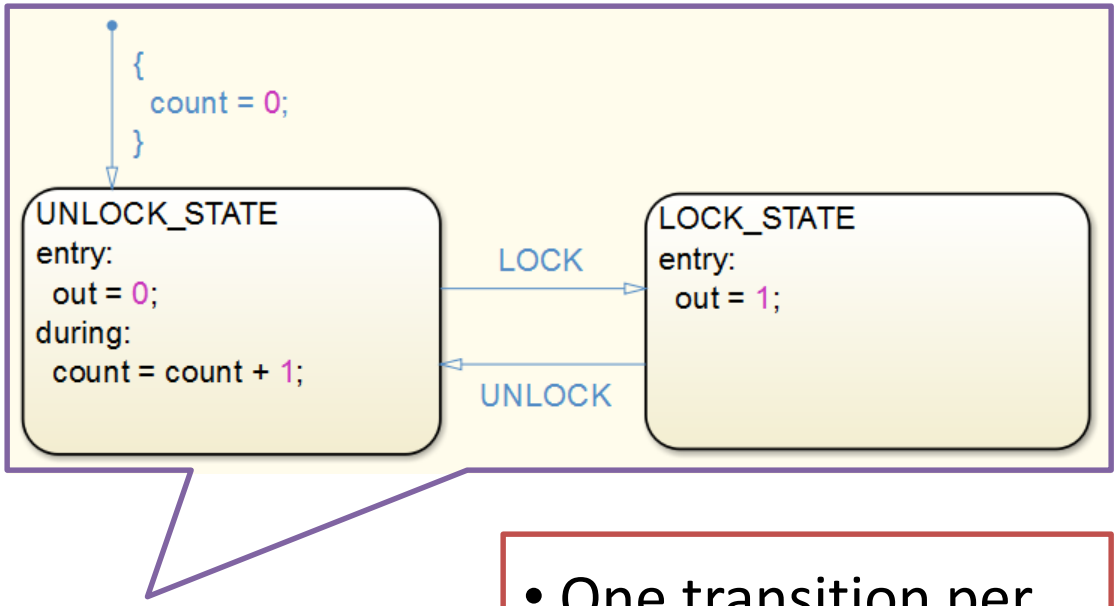
	Name	Scope	Port	Resolve Signal	DataType	Size
	in	Input	1		Inherit: Same as Simulink	-1
	out	Output	1		Inherit: Same as Simulink	-1

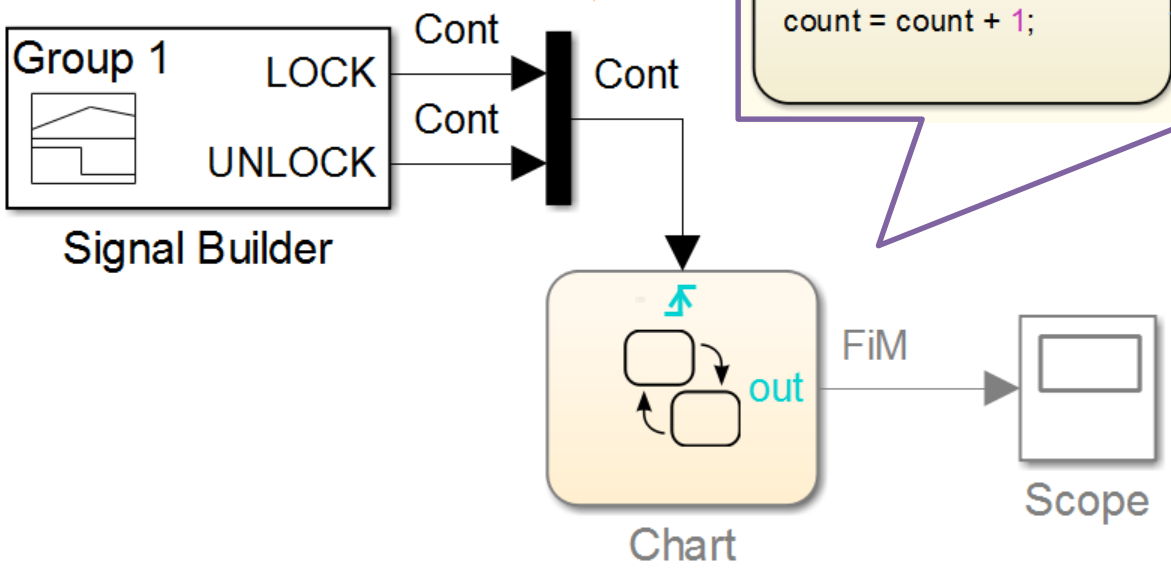


- Start and end in one step
- Terminating node has no output

State machine

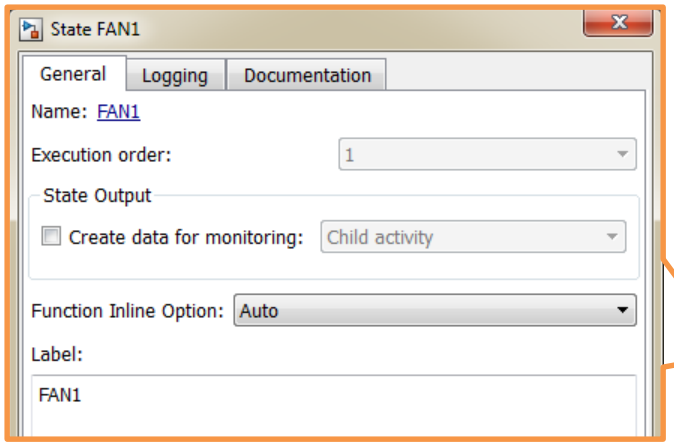
Data: persist value
Event: edge → enable
Message: queue value



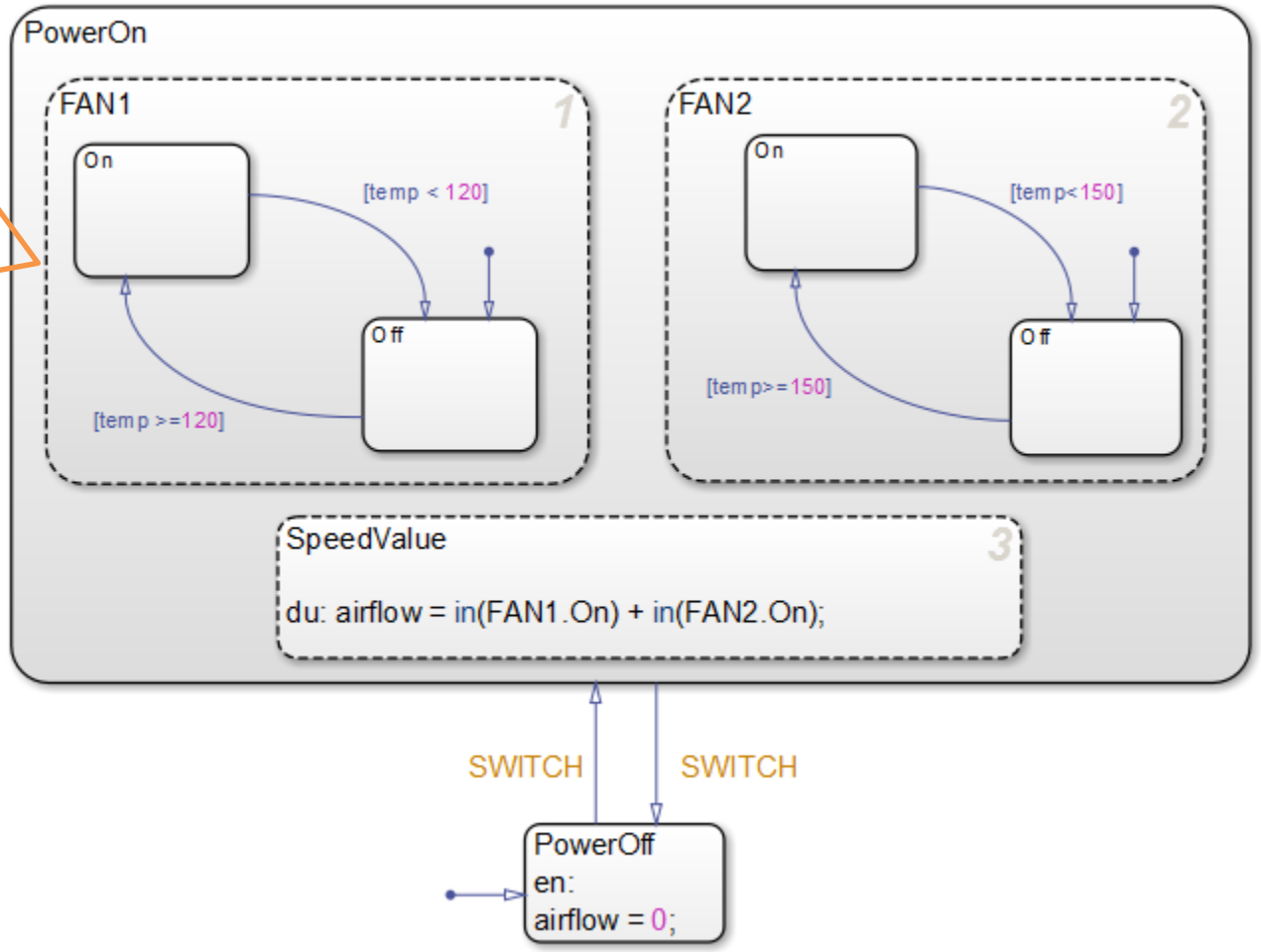


- One transition per step
- Action on transition
- State actions: entry, during, exit

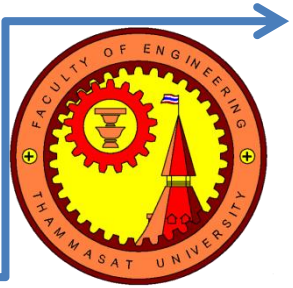
Hierarchical state machine



- Parallel (AND) states
- Exclusive (OR) states



Stateflow ↔ MATLAB/Simulink



A

during:
out = compute(in)

Simulink Fcn
y = compute(u);

B

B1
during: out1 = in;

B2
during: out2 = 2*in;

Simulink Fcn
y = compute(u);

C

C1
during: out = compute(in);

after(3,sec)

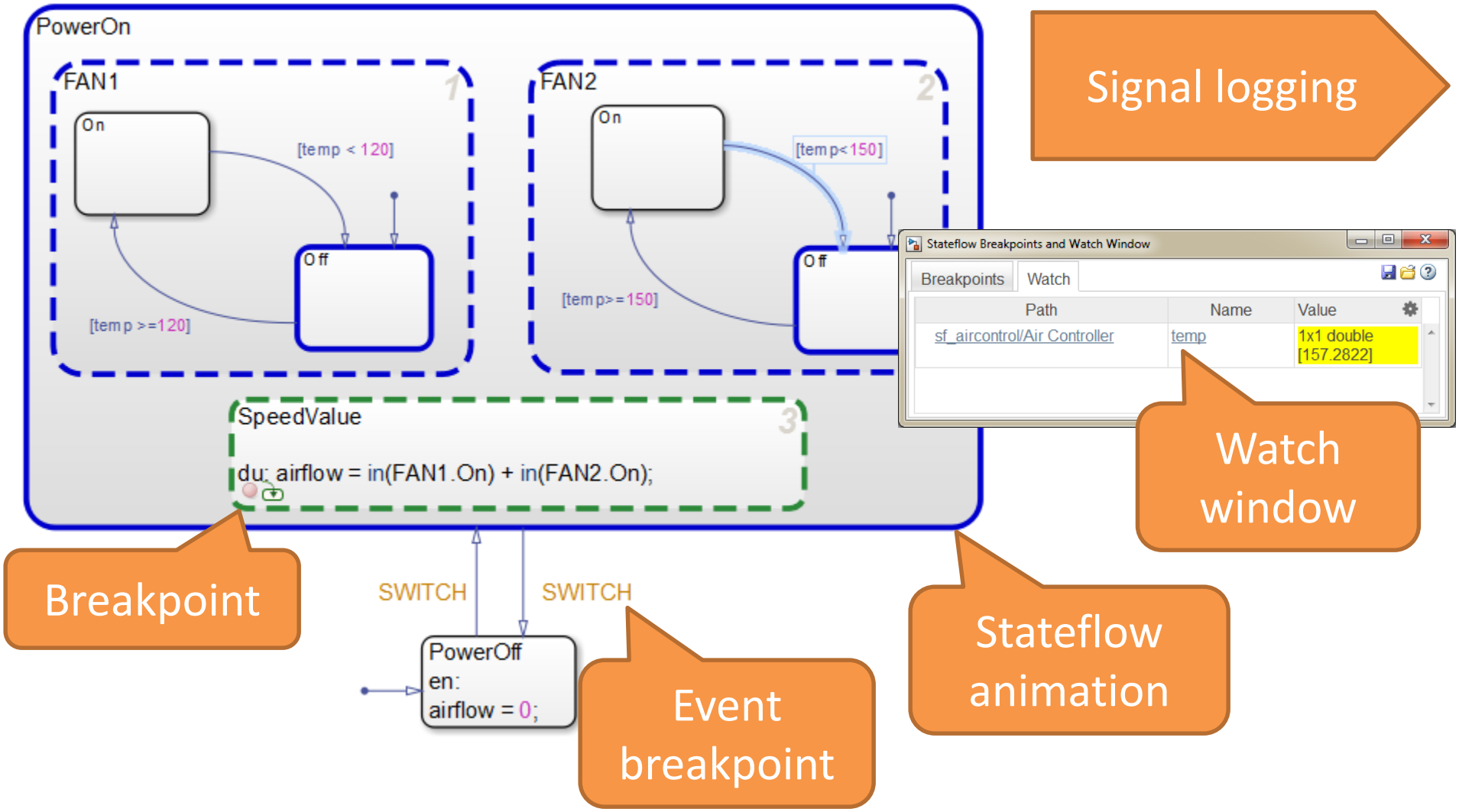
after(1,sec)

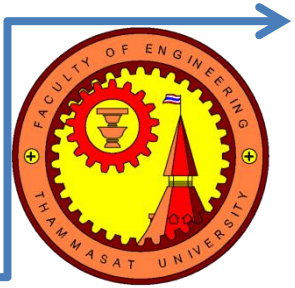
C2
during: out = calc(in);

Simulink Fcn
y = compute(u);

Simulink Fcn
y = calc(u);

Development with Stateflow



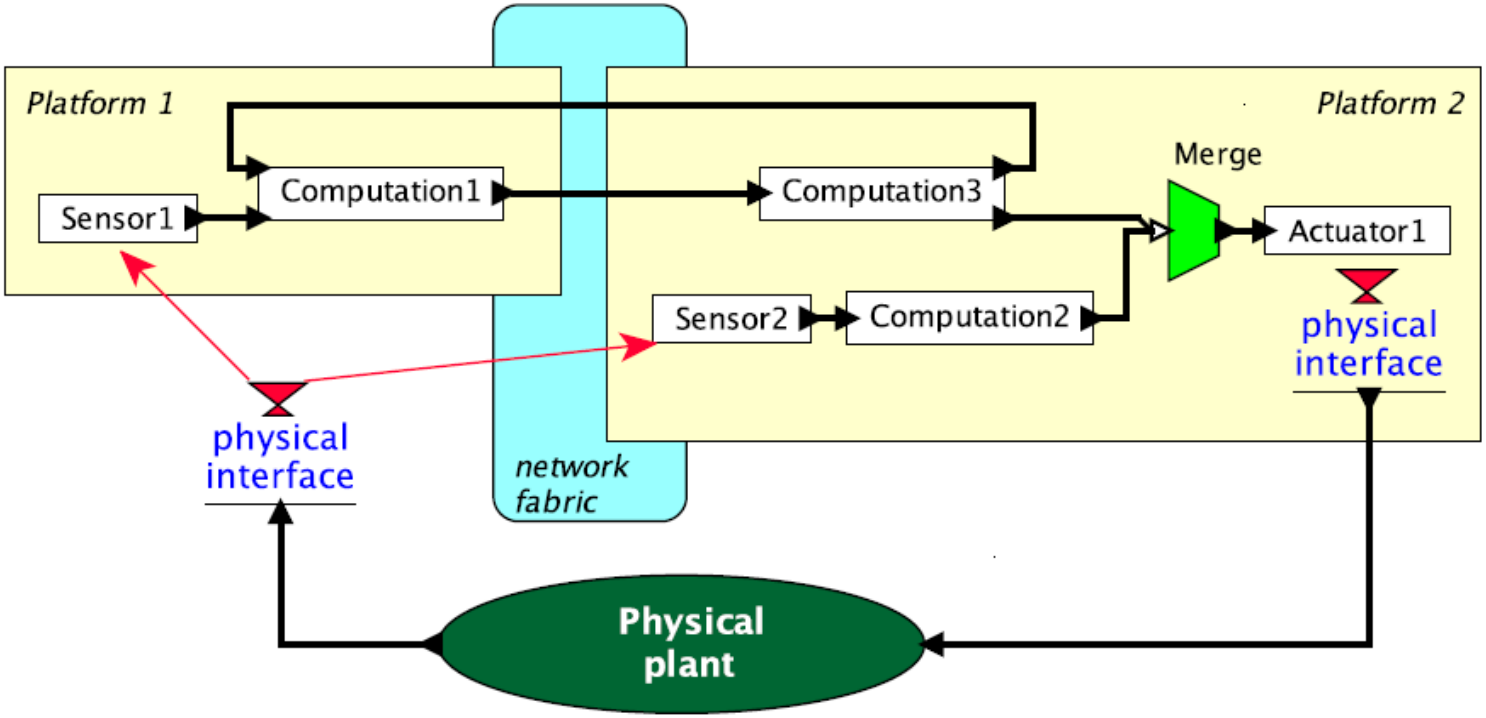


CYBER-PHYSICAL SYSTEMS

Cyber-physical systems

CyberPhysicalSystems.org

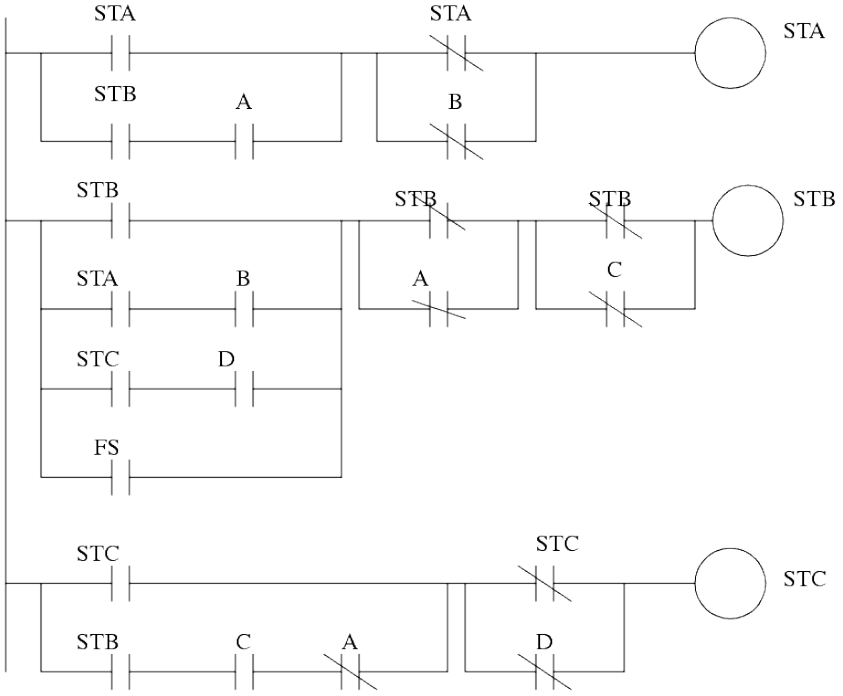
Integration of computation, networking, and physical processes



Sequence control

Wikipedia

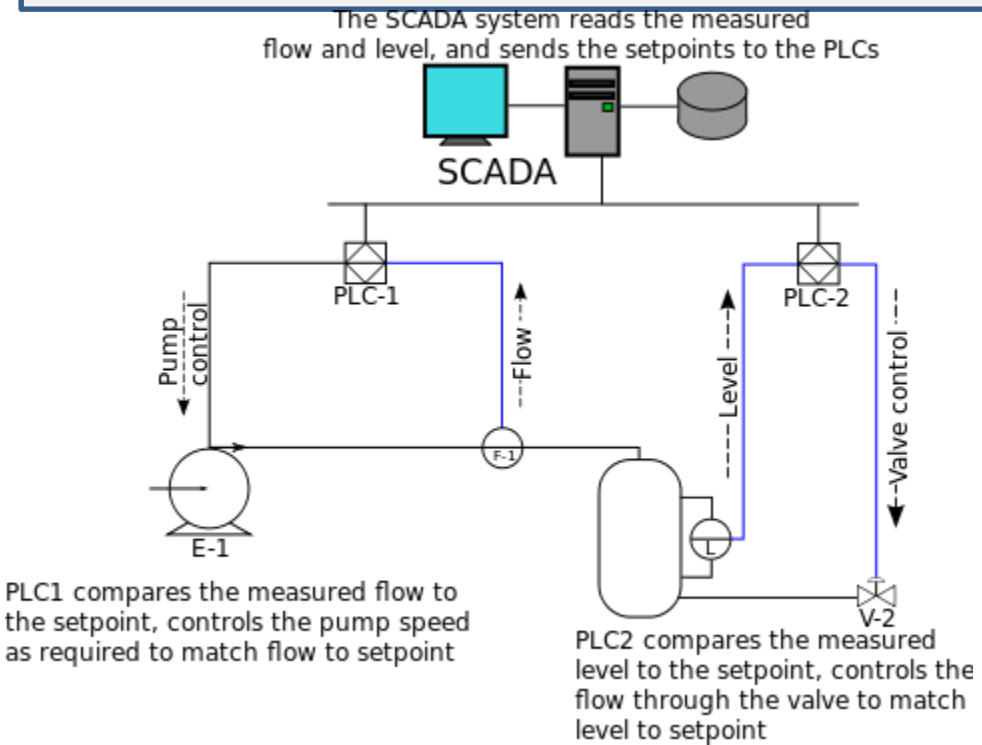
Programmable logic controller or PLC is a digital computer used for automation of typically industrial electromechanical processes



Supervisory control

Wikipedia

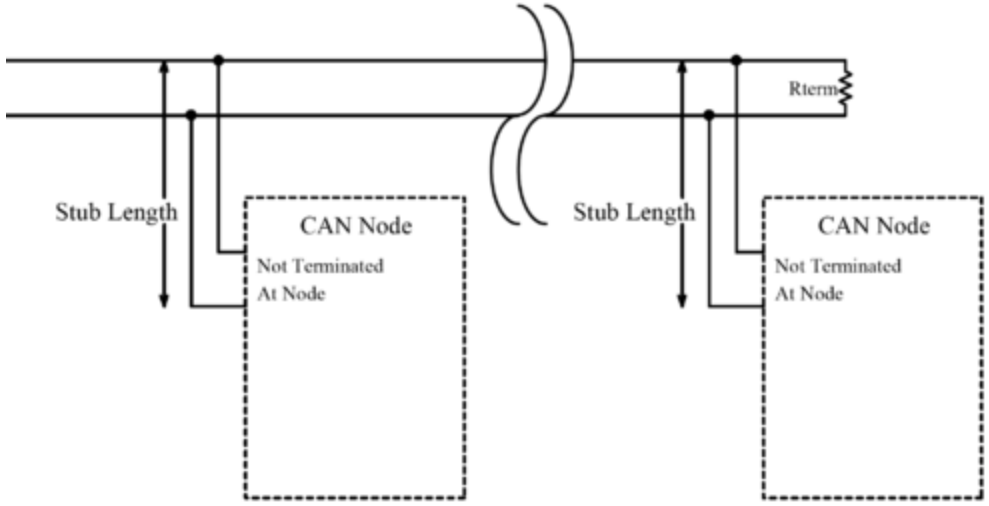
Supervisory control and data acquisition (SCADA) is a system for remote monitoring and control that operates with coded signals over communication channels

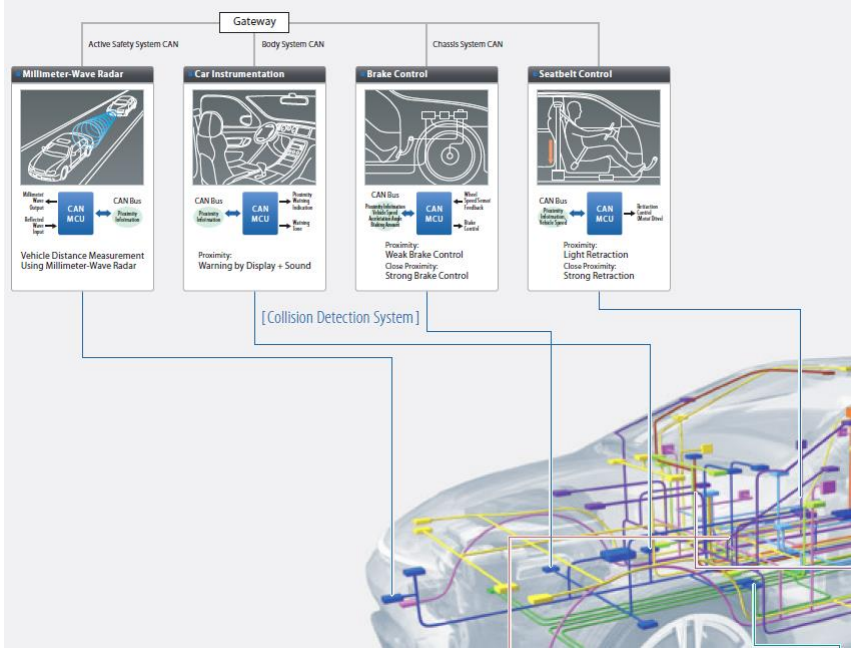


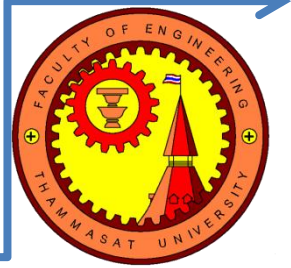
Automotive network

Wikipedia

Controller Area Network (CAN bus) is a vehicle bus standard designed to allow devices to communicate with each other in applications without a host computer.

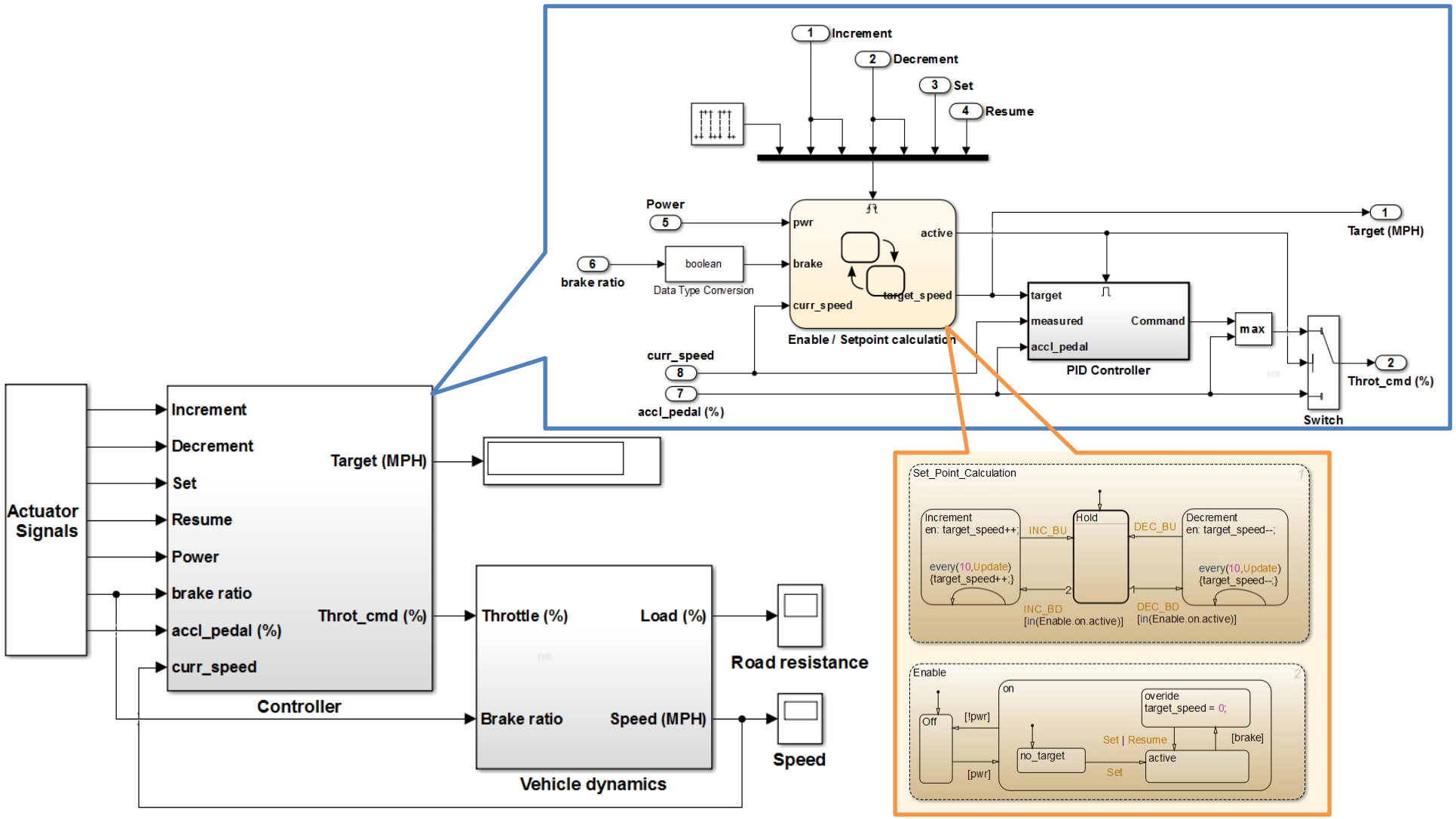




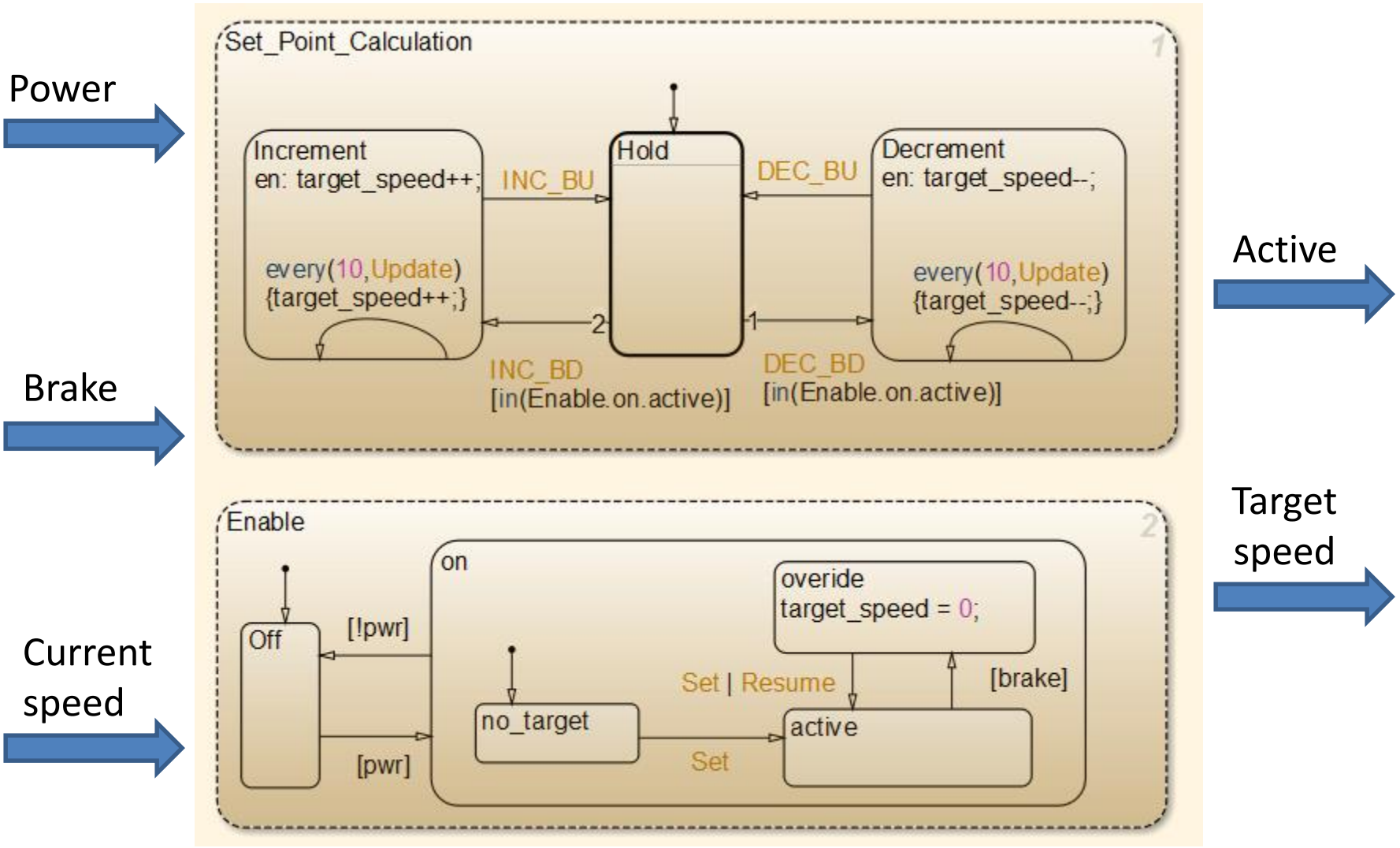


STATEFLOW APPLICATIONS

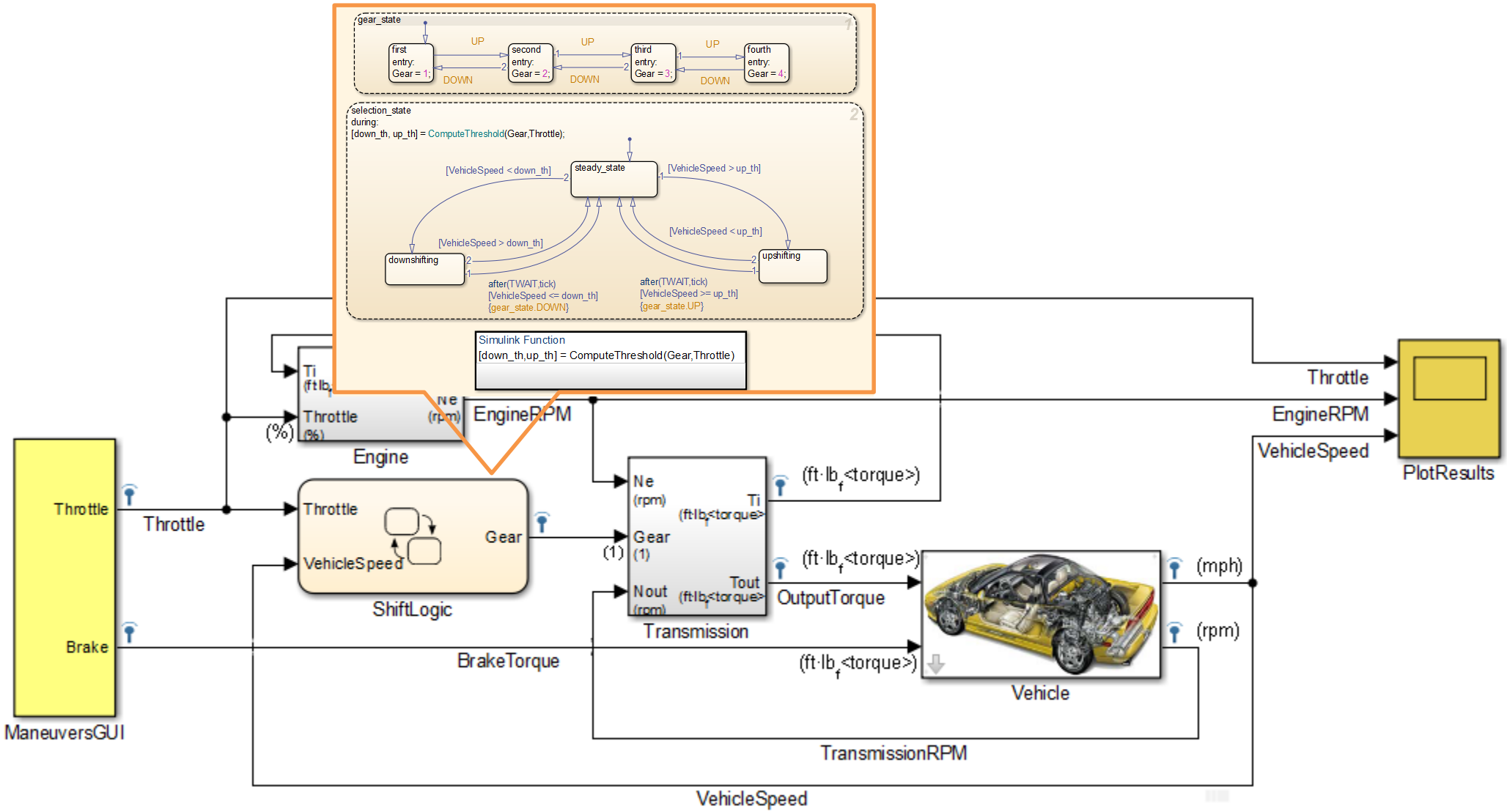
Cruise control



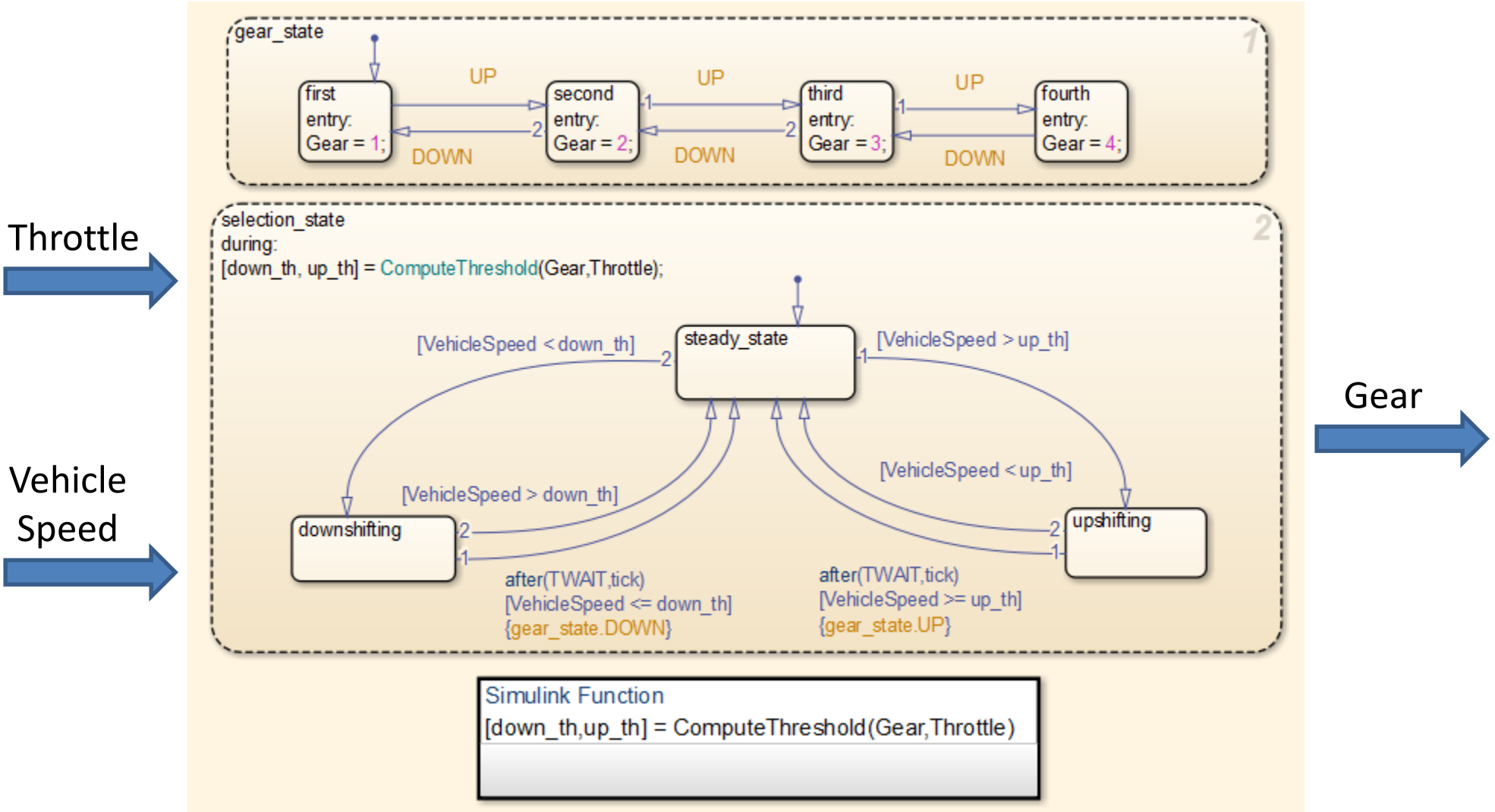
Speed controller

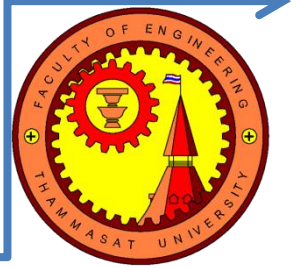


Automatic transmission



Gear-shifting logic





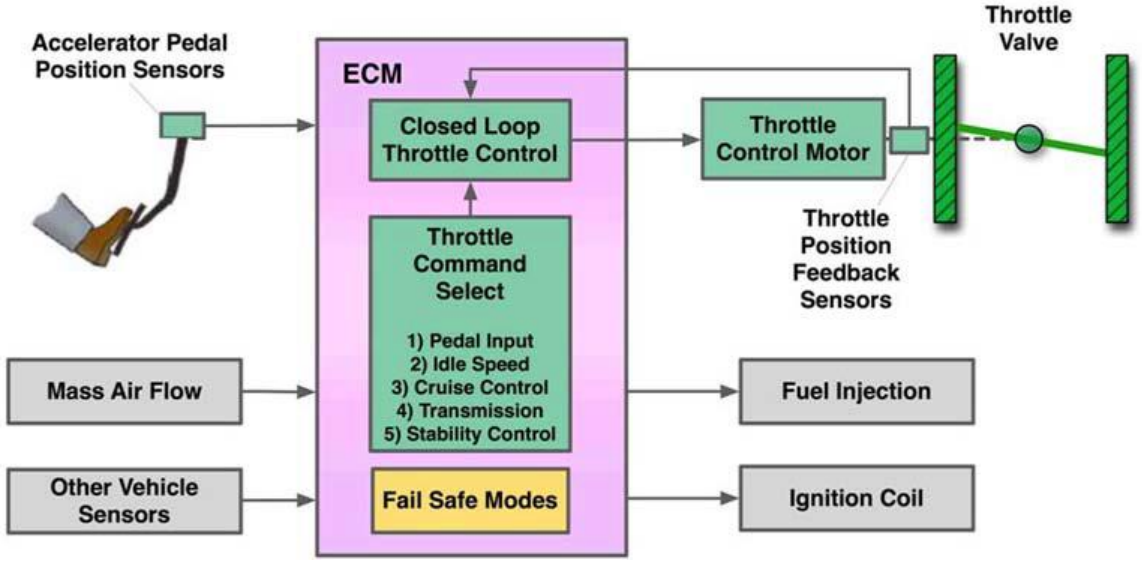
AUTOMOTIVE SAFETY

Automotive safety

NASA Report for NHTSA

The combined work of NASA and NHTSA identified no **electronic** cause of Unintended Acceleration incidents involving large throttle openings and no reason to believe that any failure of the ETC system would affect a vehicle's braking system.

Toyota ETCS-i (Electronic Throttle Control System-intelligent)

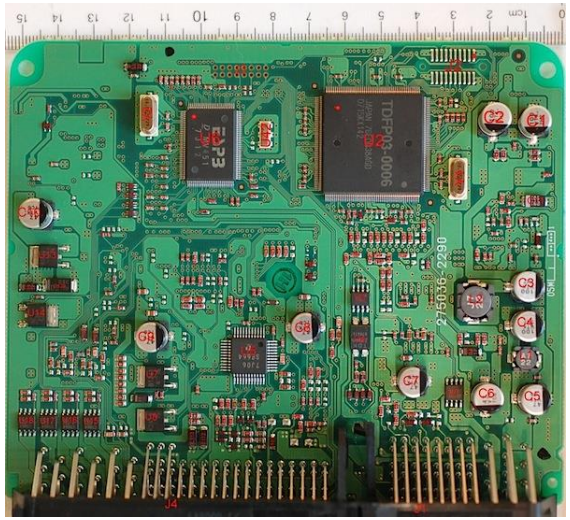


Automotive safety

Michael Barr

"Memory corruption as little as one bit flip can cause a task to die."
"The fail safes Toyota did install have gaps in them and are inadequate to detect all of the ways UA can occur via software."

Fault injection

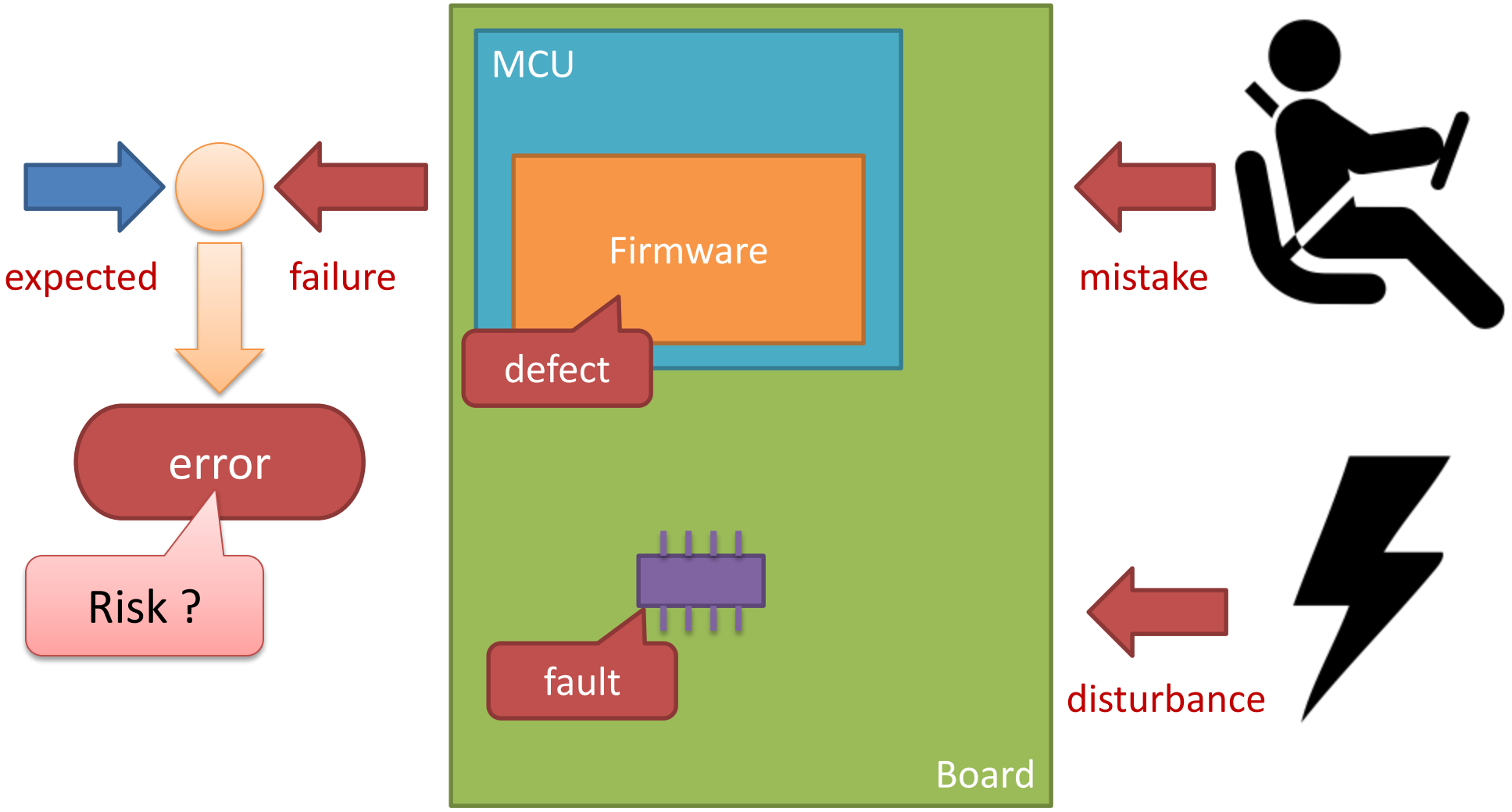


Wide Open Throttle

$3M settlement in Oklahoma courtroom

$1.2 billion economic loss settlement

Reliability engineering



ISO 26262

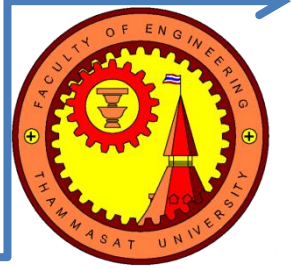
Wikipedia

An international standard for functional safety of electrical and/or electronic systems in production automobiles defined by ISO

		C1	C2	C3
S0	E1-4	QM	QM	QM
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	ASIL A
	E4	QM	ASIL A	ASIL B
S2	E1	QM	QM	QM
	E2	QM	QM	ASIL A
	E3	QM	ASIL A	ASIL B
	E4	ASIL A	ASIL B	ASIL C
S3	E1	QM	QM	ASIL A
	E2	QM	ASIL A	ASIL B
	E3	ASIL A	ASIL B	ASIL C
	E4	ASIL B	ASIL C	ASIL D

Risk =

Severity × Exposure × Controllability



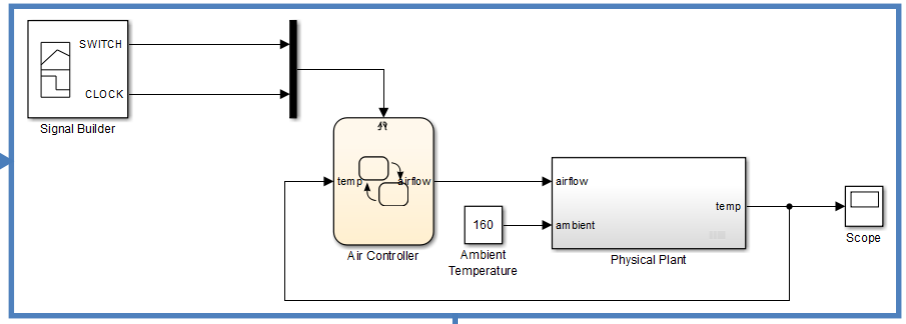
VALIDATION & VERIFICATION

V & V concepts

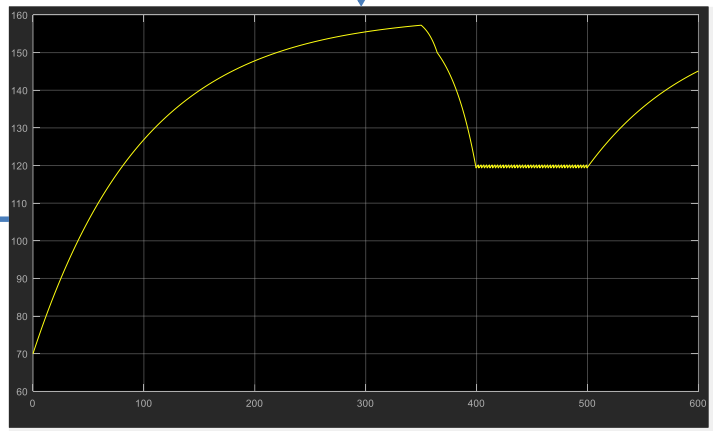
validation activity verification

- Device shall do jobs according to user manual
- Device shall be protected from abnormal usages

modeling



simulation



code generation

V & V concepts

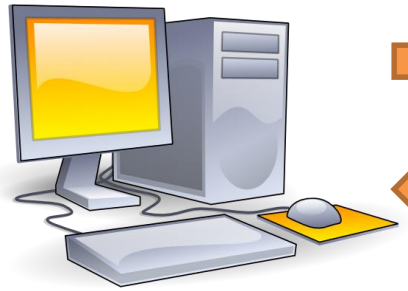
Haptic wheel movement shall be constrained within ±100 degree.

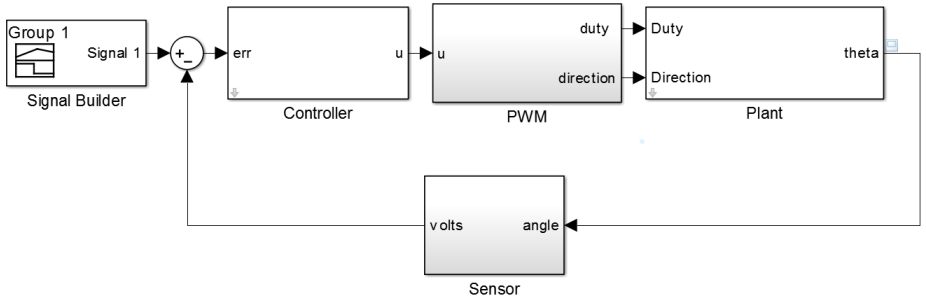
Static

Review
Walkthrough
Inspection

Rule checker

```
#include <Arduino.h>
int main() {
  setup();
  while(1) {
    loop();
  }
}
```





Dynamic

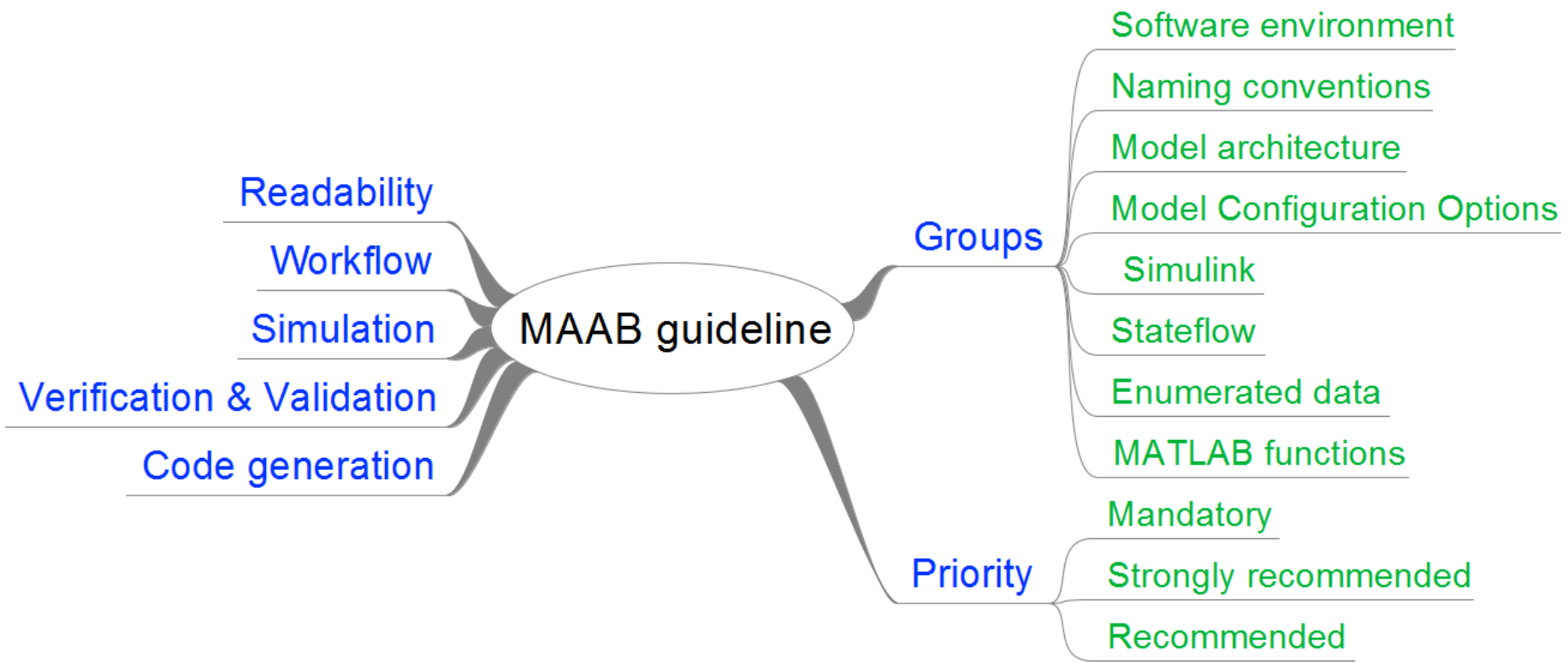
Image transfer

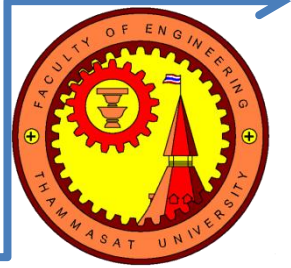
Debug, logging

Modeling guidelines

MAAB

An independent group that develops guidelines for using MATLAB, Simulink, Stateflow, and Embedded Coder





SOFTWARE TESTING

Software testing

Art of Software Testing

Testing is the process of executing a program with the intent of finding errors.

Test input
- valid
- invalid

Test mockup

Source code

driver | stub

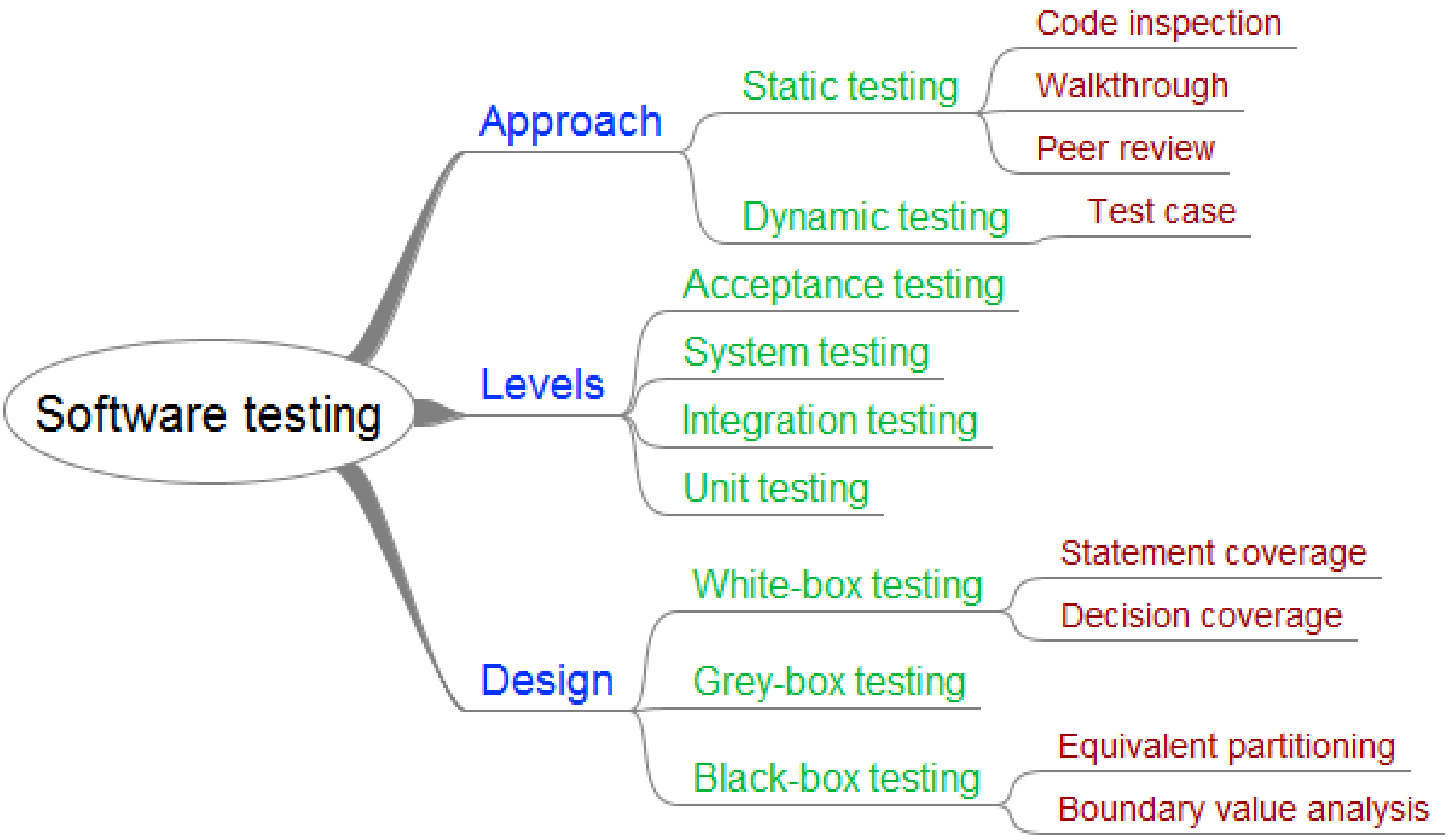
Actual output

Pass

Fail

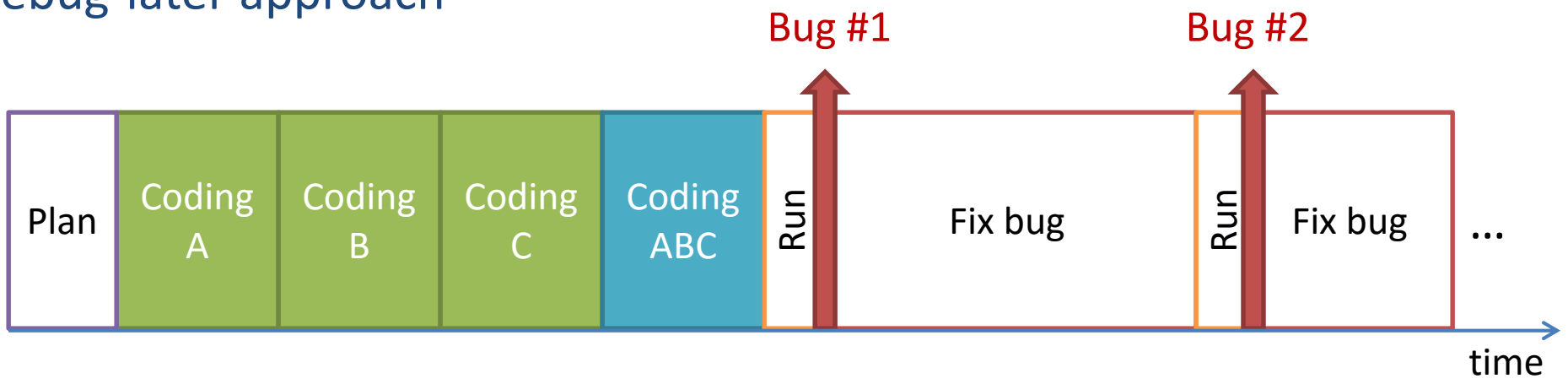
Expected output

Software testing

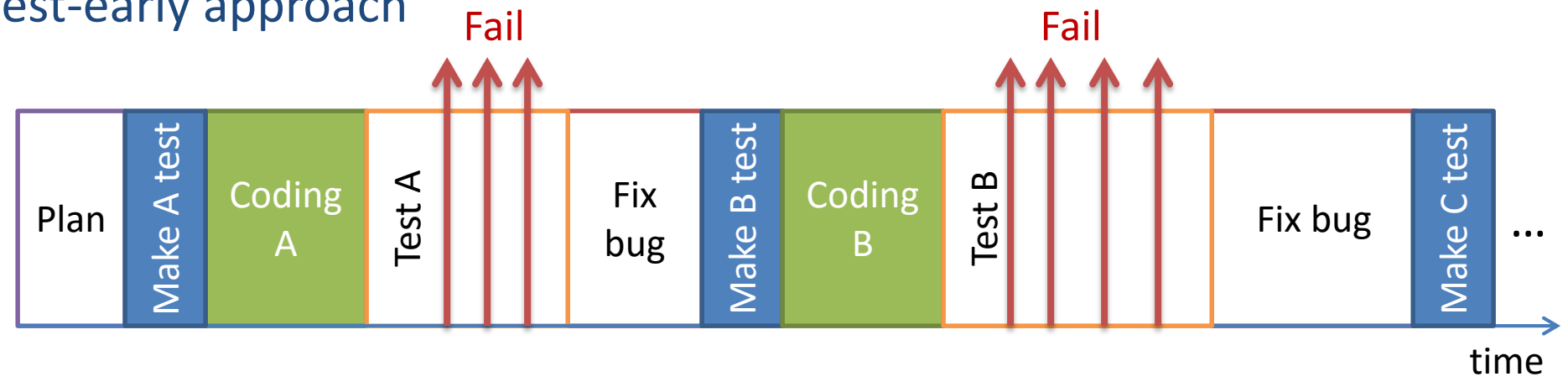


Test-driven development

Debug-later approach



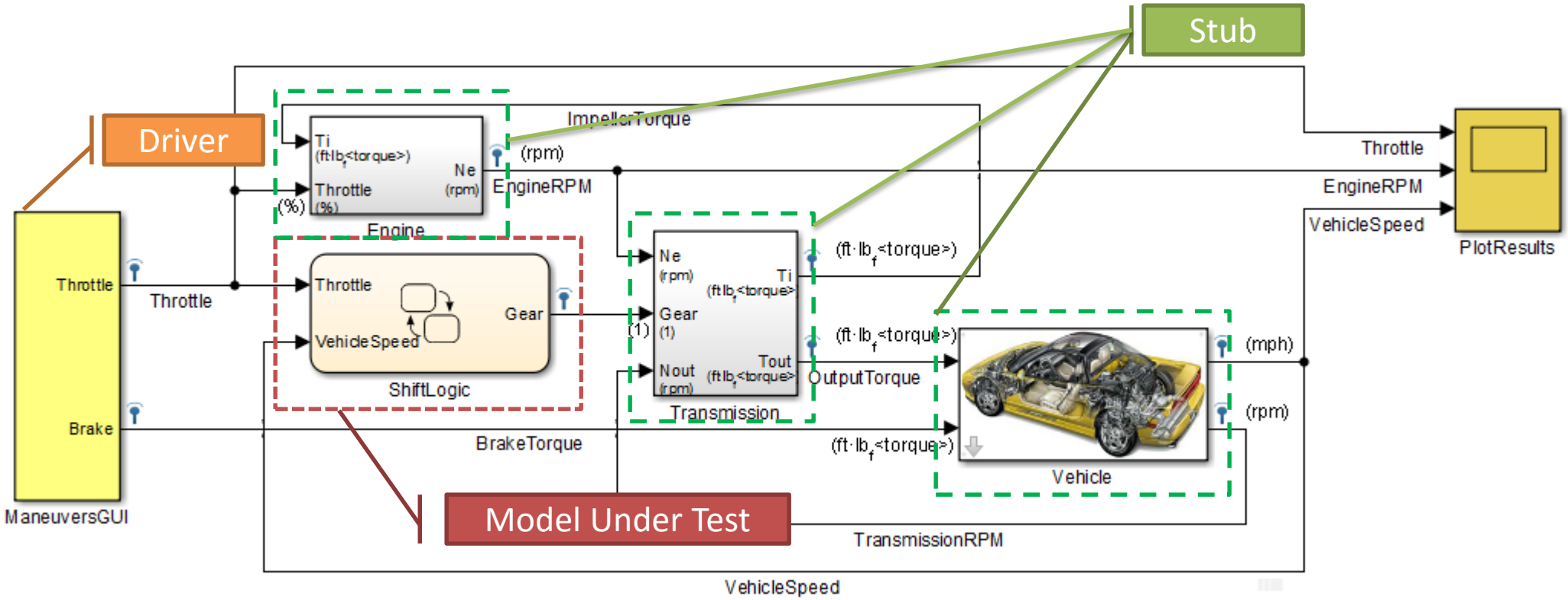
Test-early approach



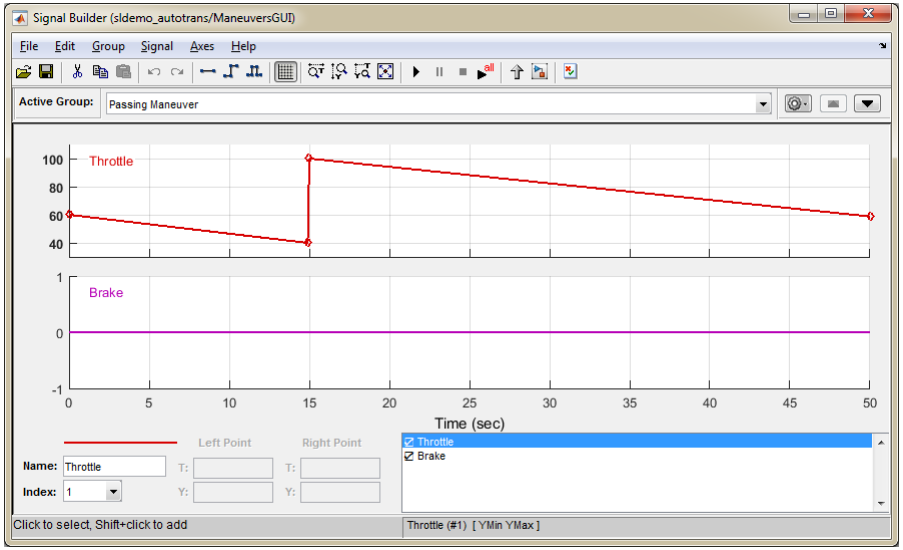
TDD, MBD, V-model

Add a test | Run test to fail | Write the code | Run tests | Refactor

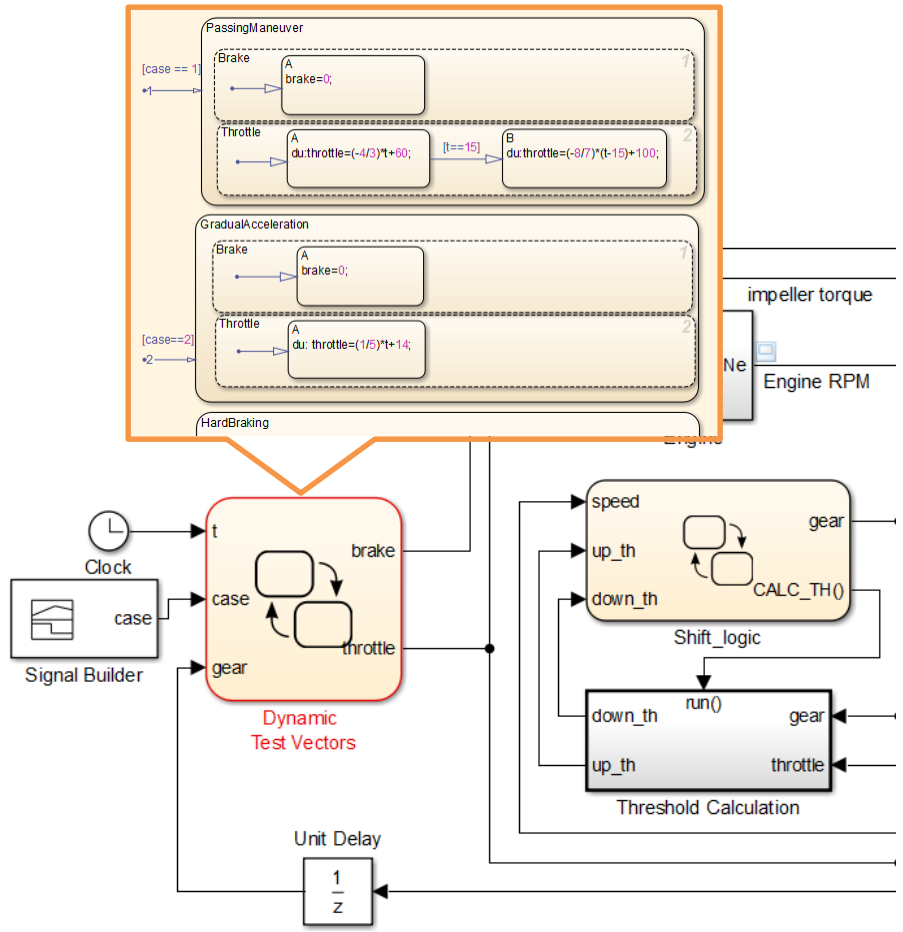
Create test model → Design test inputs → Add controller model → Simulate → Refine model



Test case design



- Input that force controller to operate in each mode.
- Input to drive the plant into specific behaviors.
- Input to test coverage of operations



Fault injection

Wikipedia

A technique for improving the coverage of a test by introducing faults to test code paths, in particular error handling code paths, that might otherwise rarely be followed.

Memory violation

```
uint16_t buf[50];
for (i=0;i<=50;i++) {
  buf[i] = readADC();
}
```

```
uint16_t *pbuf;
*pbuf = readADC();
```

Stack overflow