Berkeley Engineering

Educating Leaders. Creating Knowledge. Serving Society.

UC Berkeley
College of Engineering

International and Corporate Partnerships

Anthony St. George, Ph.D.
Assistant Dean
International and Corporate Partnerships
College of Engineering

Summary Agenda

1. College of Engineering International Partnerships

2. College of Engineering Executive and Professional Education

Berkeley Engineering

UC Berkeley: A Tradition of Access & Excellence

Educating Leaders. Creating Knowledge. Serving Society.

Berkeley Engineering

UC Berkeley Overview

Educating Leaders. Creating Knowledge. Serving Society.

UC Berkeley by the Numbers:

- Founded in 1868
- 14 Colleges and Schools
- 170 departments and interdisciplinary research units
- 48 of 52 doctoral programs ranked in top 10.
- 8 current Nobel Laureates; 29 Alumni Nobel Laureates
- 1620 Full-time Faculty

Student Life

- 27,126 Undergraduates
- 10,455 Graduate students

Research Funding

$730 Million in FY14-15

Berkeley Engineering Overview

Educating Leaders. Creating Knowledge. Serving Society.

Academics

- 7 departments
- 215 faculty
- 3,100 undergraduates
- 1,800 graduate students
- 9 undergraduate, 7 graduate programs ranked in US News & World Report Top 5

2015 USNWR:

#1 Graduate Programs:

- **Civil Engineering**
- **Computer Science**
- **Environmental Engineering**
- **Computer Engineering**

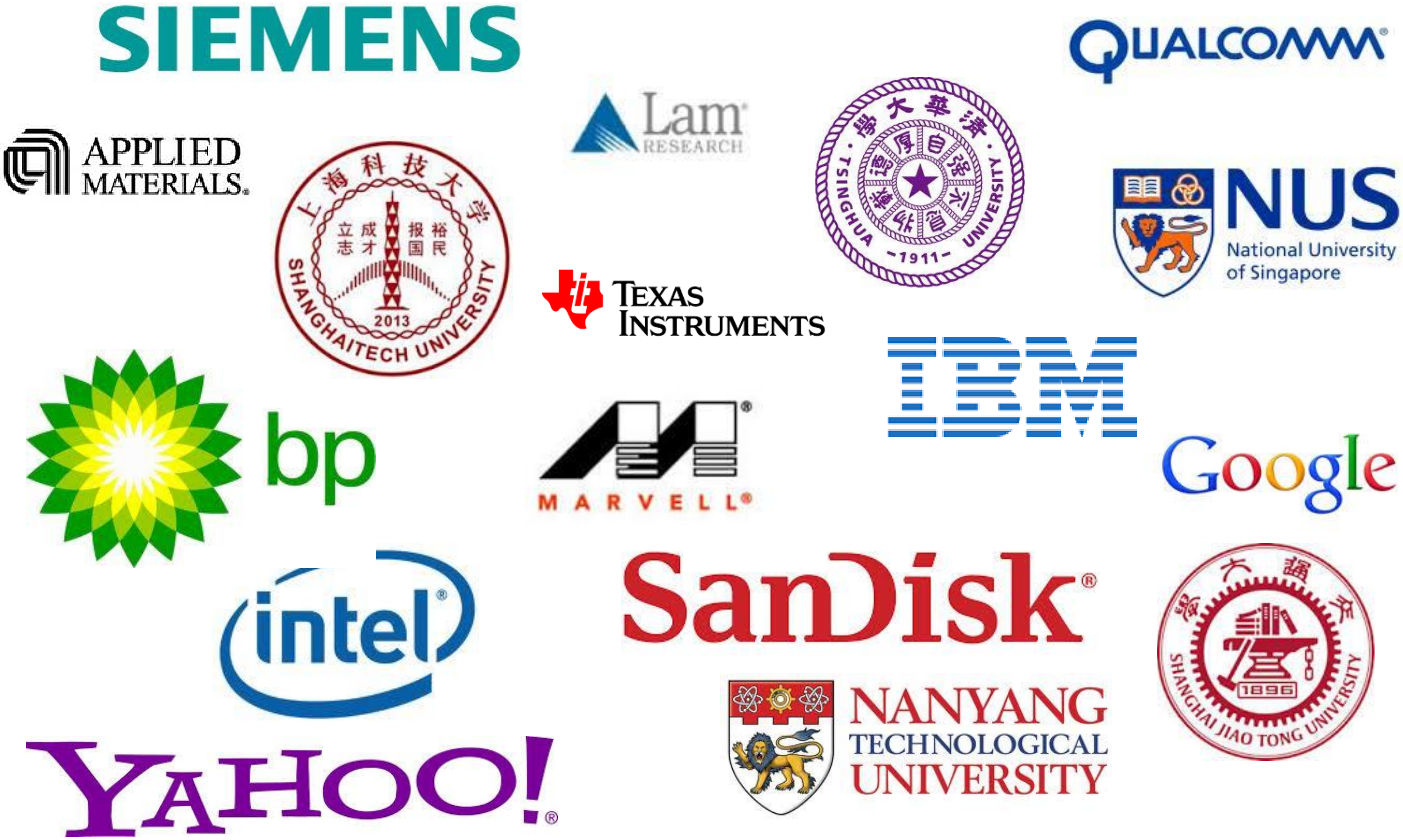
Research

- Cutting-edge work in semiconductor, optoelectronic, MEMS devices, wireless communications, parallel computing, synthetic biology, networks, robotics, …
- More than 20 research centers
- 78 faculty in National Academy of Engineering
- 3 Turing Award Recipients

Berkeley Engineering

UC Berkeley College of Engineering
Industry & International Academic Partnerships

Educating Leaders. Creating Knowledge. Serving Society.



Our Students by Department (Fall 2014)

3,136 undergraduate + 1,804 graduate students = 4,943 total enrolled students

DEPARTMENT (in order by undergraduate student count)	Undergraduate	Graduate	Total
Electrical Engr & Computer Sciences	1,258	549	1,807
Mechanical Engineering	624	348	972
Bioengineering	360	204	564
Civil & Environmental Engineering	298	375	673
Materials Science & Engineering	146	96	242
Industrial Eng. & Operations Research	124	122	246
Nuclear Engineering	72	74	146
Other (Undgrad: Eng Science, Grad: App Sci & Tech)	254	39	293
Total COE	3,136	1,807	4,943
L&S Comp. Sci and Oper Res & Mngt Sci*	501		501
TOTAL	3,637	1,807	5,444

Berkeley Engineering

Educating Leaders. Creating Knowledge. Serving Society.

Recent Highlights

Educating Leaders:

- Jacobs Hall and Design institute: Opened August 20, 2015
- Certificate/Minor in Design Innovation, launched Fall 2015
- **Center for Entrepreneurship and Technology**, re-launched and named by Pantas and Ting Sutardja, led by Ikhlaq Sidhu and Phil Kaminsky

Creating Knowledge:

- New Siebel Energy Institute, partnership with MIT, UIUC, CMU, Princeton, Ecole Polytechnique, University of Tokyo, Politecnico Di Torino. First projects selected, August 2015.
- **Tsinghua Berkeley Shenzen Institute** for research in precision medicine and healthcare, energy and environmental technologies, and information technology and data science. Official launch, October 20, 2015. Faculty Director Connie Chang-Hasnain.

Serving Society:

- "Girls in Engineering" – Middle school girls pipeline building, led by Claire Tomlin and Annie Averitt, funded by NSF + other partners – Summer 2015
- Designated Emphasis in Development Engineering, led by Alice Agogino and Clair Brown, Launched in Fall 2014, hosted TechCon, December 2014
- Several startups associated with USAID's Development Innovation Laboratory (DIL): Cellscope, Endaga, Gram Power, Tarana Wireless, …

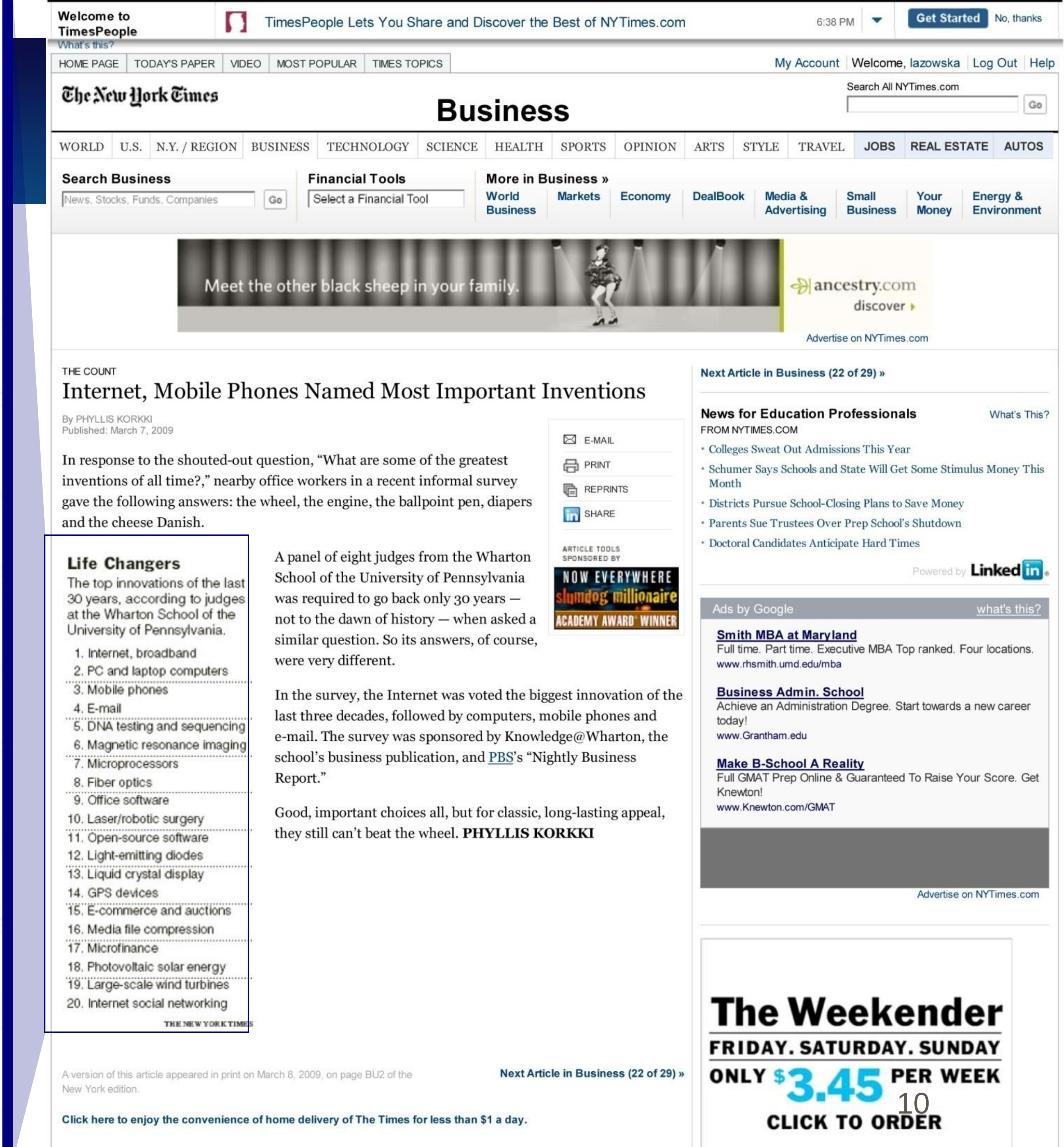
Industry-Academia Engagement

Life Changers

The top innovations of the last 30 years, according to judges at the Wharton School of the University of Pennsylvania.

- ● Internet, broadband
- ● PC and laptop computers
- ● Mobile phones
- ● E-mail
- ● DNA testing and sequencing
- ● Magnetic resonance imaging
- ● Microprocessors
- 8. Fiber optics
- ● Office software
- 10. Laser/robotic surgery
- ● Open-source software
- 12. Light-emitting diodes
- 13. Liquid crystal display
- ● GPS devices
- ● E-commerce and auctions
- ● Media file compression
- 17. Microfinance
- 18. Photovoltaic solar energy
- 19. Large-scale wind turbines
- ● Internet social networking

THE NEW YORK TIMES



HOME PAGE | TODAY'S PAPER | VIDEO | MOST POPULAR | TIMES TOPICS
My Account | Welcome, lazowska | Log Out | Help

The New York Times

Business

Search All NYTimes.com Go

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION | ARTS | STYLE | TRAVEL | JOBS | REAL ESTATE | AUTOS

Search Business
News, Stocks, Funds, Companies Go

Financial Tools
Select a Financial Tool

More in Business »
World Business | Markets | Economy | DealBook | Media & Advertising | Small Business | Your Money | Energy & Environment

THE COUNT

Internet, Mobile Phones Named Most Important Inventions

By PHYLLIS KORKKI
Published: March 7, 2009

E-MAIL
PRINT
REPRINTS
SHARE

In response to the shouted-out question, "What are some of the greatest inventions of all time?," nearby office workers in a recent informal survey gave the following answers: the wheel, the engine, the ballpoint pen, diapers and the cheese Danish.

Life Changers

The top innovations of the last 30 years, according to judges at the Wharton School of the University of Pennsylvania.

1. Internet, broadband
2. PC and laptop computers
3. Mobile phones
4. E-mail
5. DNA testing and sequencing
6. Magnetic resonance imaging
7. Microprocessors
8. Fiber optics
9. Office software
10. Laser/robotic surgery
11. Open-source software
12. Light-emitting diodes
13. Liquid crystal display
14. GPS devices
15. E-commerce and auctions
16. Media file compression
17. Microfinance
18. Photovoltaic solar energy
19. Large-scale wind turbines
20. Internet social networking

THE NEW YORK TIMES

A panel of eight judges from the Wharton School of the University of Pennsylvania was required to go back only 30 years — not to the dawn of history — when asked a similar question. So its answers, of course, were very different.

In the survey, the Internet was voted the biggest innovation of the last three decades, followed by computers, mobile phones and e-mail. The survey was sponsored by Knowledge@Wharton, the school's business publication, and PBS's "Nightly Business Report."

Good, important choices all, but for classic, long-lasting appeal, they still can't beat the wheel. **PHYLLIS KORKKI**

A version of this article appeared in print on March 8, 2009, on page BU2 of the New York edition.

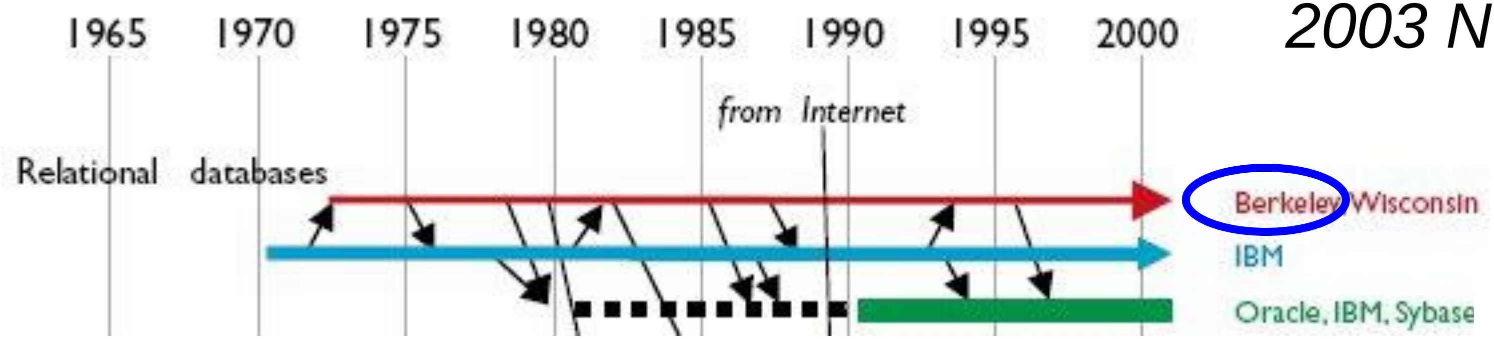
Next Article in Business (22 of 29) »

Next Article in Business (22 of 29) »

News for Education Professionals What's This?
FROM NYTIMES.COM
- Colleges Sweat Out Admissions This Year
- Schumer Says Schools and State Will Get Some Stimulus Money This Month
- Districts Pursue School-Closing Plans to Save Money
- Parents Sue Trustees Over Prep School's Shutdown
- Doctoral Candidates Anticipate Hard Times

Powered by LinkedIn

University and Company Research Creates $1B Industries



2003 NRC study

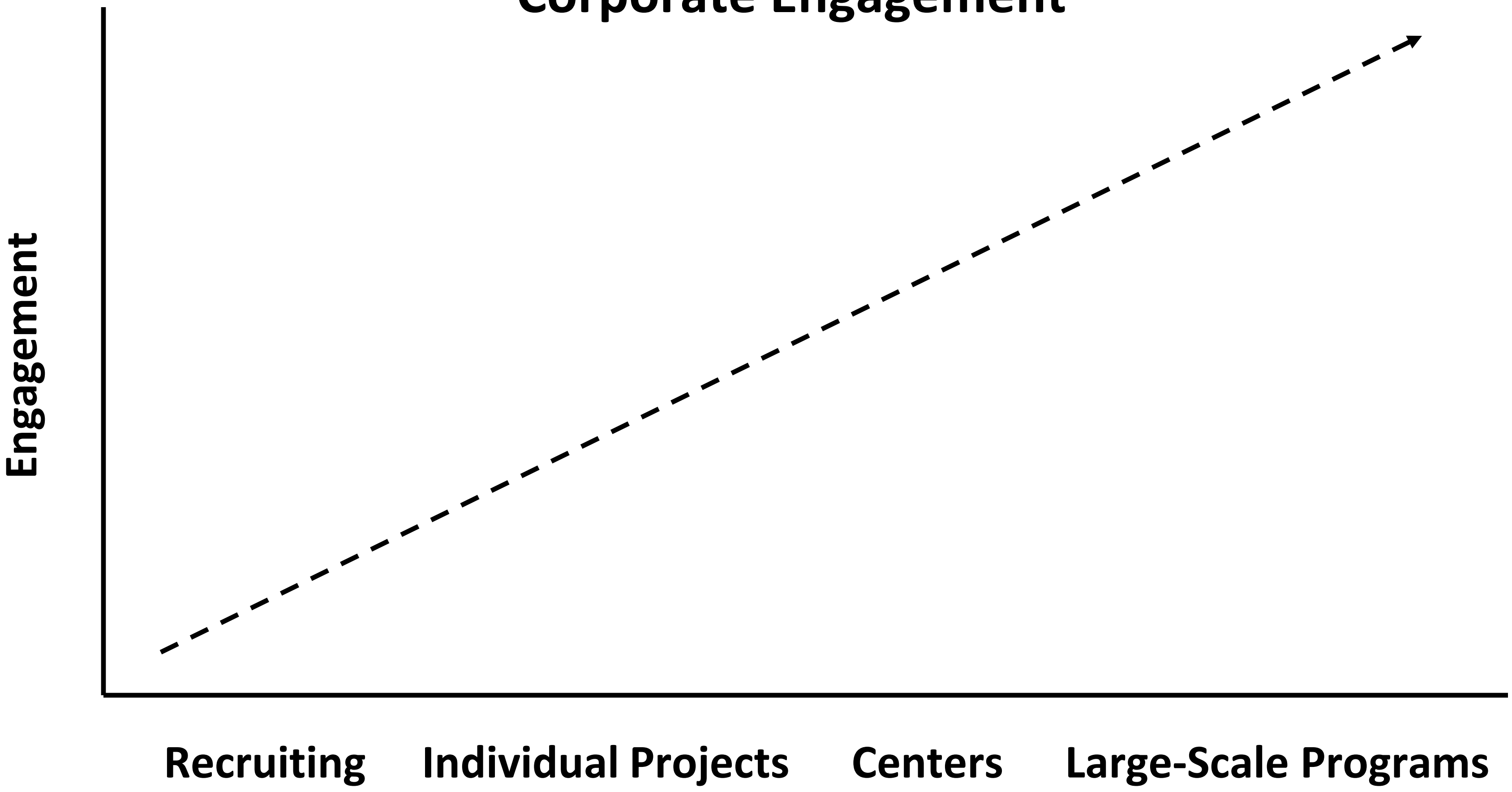
University

Company

$1B Industry

- Not a “pipeline” – lots of “back-and-forth”
- 10-15 years from idea to $1B industry
- Research puts ideas in storehouse for later use
- Unanticipated results often as important: hard to predict the next “big hit”

Corporate Engagement



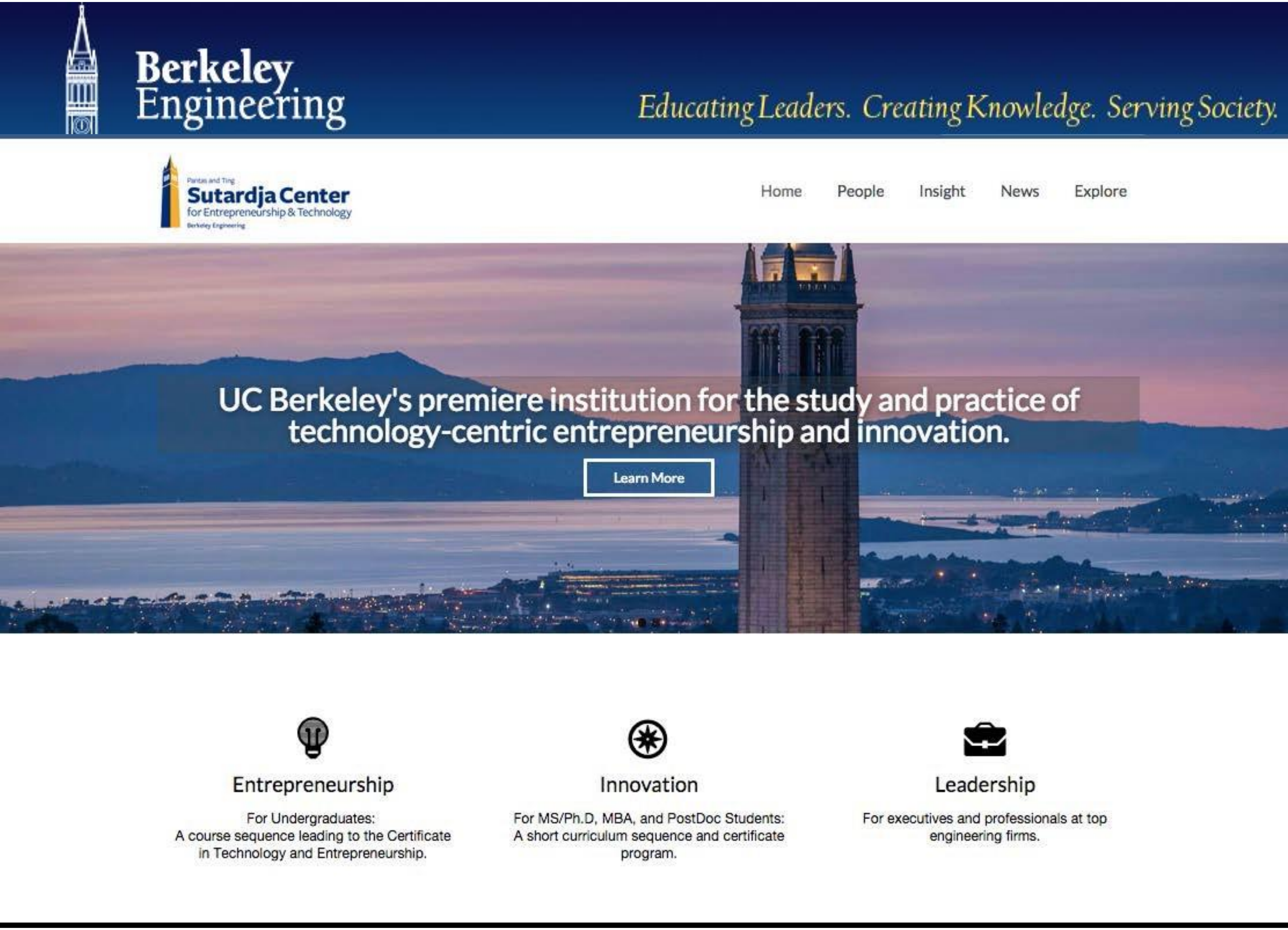
Corporate Engagement

1. **Recruiting: e.g., EECS Industrial Liaison Program ($15,000/year)**
   (includes annual symposium, career fair, internship open house, and infosessions)

2. **Individual Projects: ~$50K - $75K/year**
   Support of one project with one graduate student. Does not include new equipment or lab fees. Contract or gift funding possible.

3. **Center Membership or COE or Department Support: ~$50K-$500K/year**
   Corporate engagement examples include membership in a research center, executive education programs, gifts for new initiatives, student support, graduate fellowships, professorships, or capital improvements. Corporations may make several gifts across the college in different areas. Examples include: GM, SanDisk, Lam, or membership in BSAC, TRUST, SWARM Lab, etc.

4. **Large-scale Programs: $1M +/year**
   Combination of the above or support of multiple coordinate/thematic projects, such as Siemens, Intel, BP, Hyundai, etc.



Berkeley Engineering

Educating Leaders. Creating Knowledge. Serving Society.

Pantas and Ting Sutardja Center for Entrepreneurship & Technology Berkeley Engineering

UC Berkeley's premiere institution for the study and practice of technology-centric entrepreneurship and innovation.

Entrepreneurship

For Undergraduates:
A course sequence leading to the Certificate in Technology and Entrepreneurship.

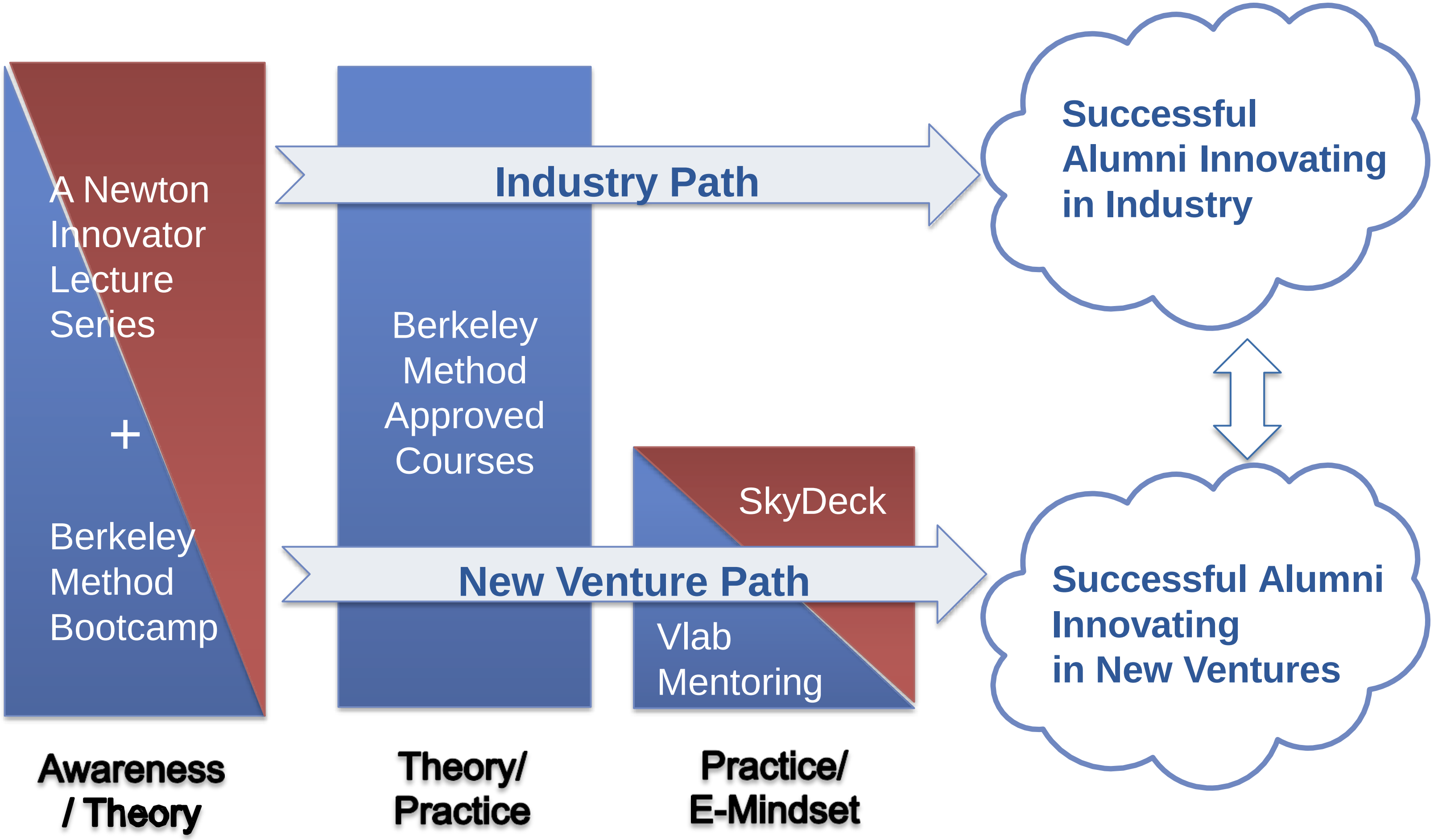
Innovation

For MS/Ph.D, MBA, and PostDoc Students:
A short curriculum sequence and certificate program.

Leadership

For executives and professionals at top engineering firms.

S-CET Curriculum



A Newton Innovator Lecture Series

+

Berkeley Method Bootcamp

Berkeley Method Approved Courses

SkyDeck

Vlab Mentoring

Industry Path

New Venture Path

Successful Alumni Innovating in Industry

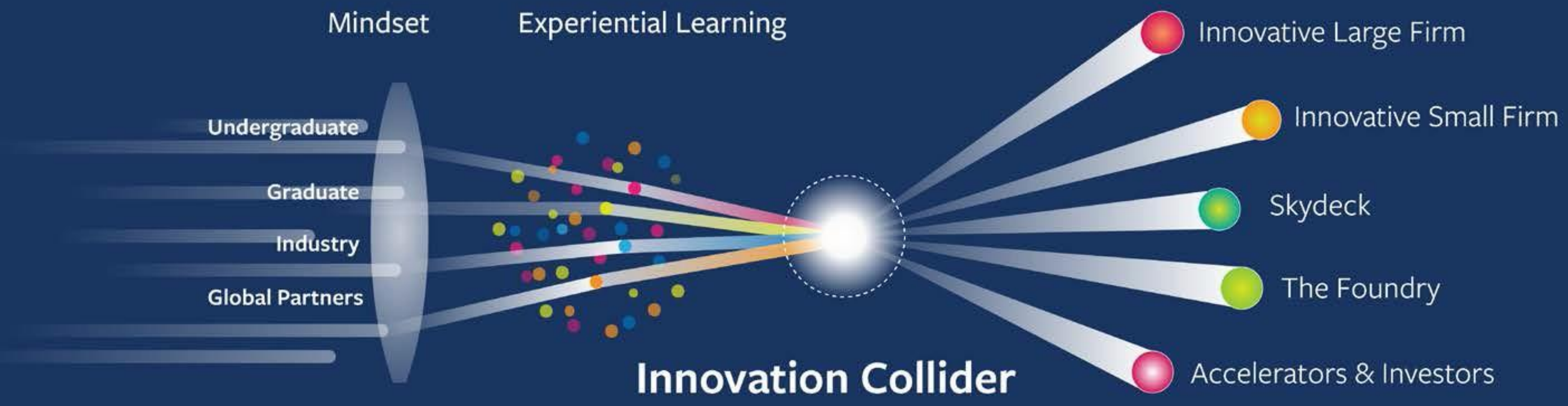
Successful Alumni Innovating in New Ventures

Awareness / Theory

Theory/ Practice

Practice/ E-Mindset

The Innovation Collider Model



Mindset

Experiential Learning

Undergraduate

Graduate

Industry

Global Partners

Innovation Collider

Innovative Large Firm

Innovative Small Firm

Skydeck

The Foundry

Accelerators & Investors

Innovation Collider Examples

Venture Creation

Recruiting Talent and Industry Application

Innovation and Translational Research

Foundation CAPITAL

Nosocom Solutions

hitmap

amazon

Supply Chain Logistics

Radius Machine Learning

KABAM!

Games and Data Analytics

GEISINGER HEALTH SYSTEM

Data and Healthcare

AUTODESK – NATIONAL INSTRUMENTS

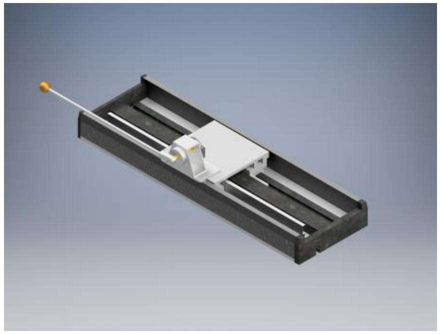
AUTOMATING WORKFLOW FROM CAD TO CONTROLS

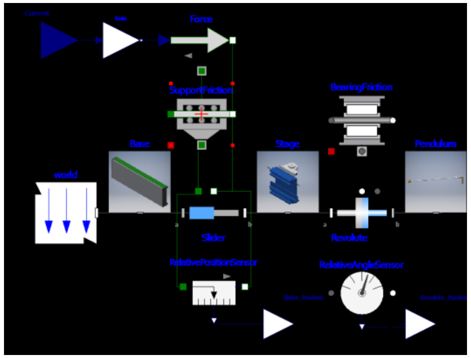
Professor David Auslander
Dr. George Anwar

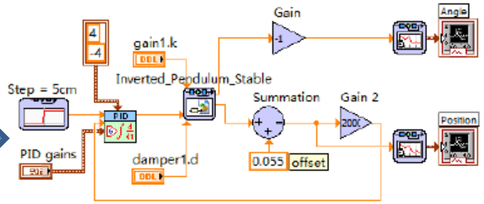
Master of Engineering Capstone Project Info Session 2016

AUTODESK. NATIONAL INSTRUMENTS™ JACOBS INSTITUTE FOR DESIGN INNOVATION COLLEGE OF ENGINEERING, UC BERKELEY Berkeley Engineering

MOTIVATION: To create a tool suite that goes from 3D modeling to physical implementation seamlessly







AUTODESK® INVENTOR®

MODELICA

NATIONAL INSTRUMENTS LabVIEW

"To provide a tool for mechanical systems as the tool Spice provided for electrical systems"

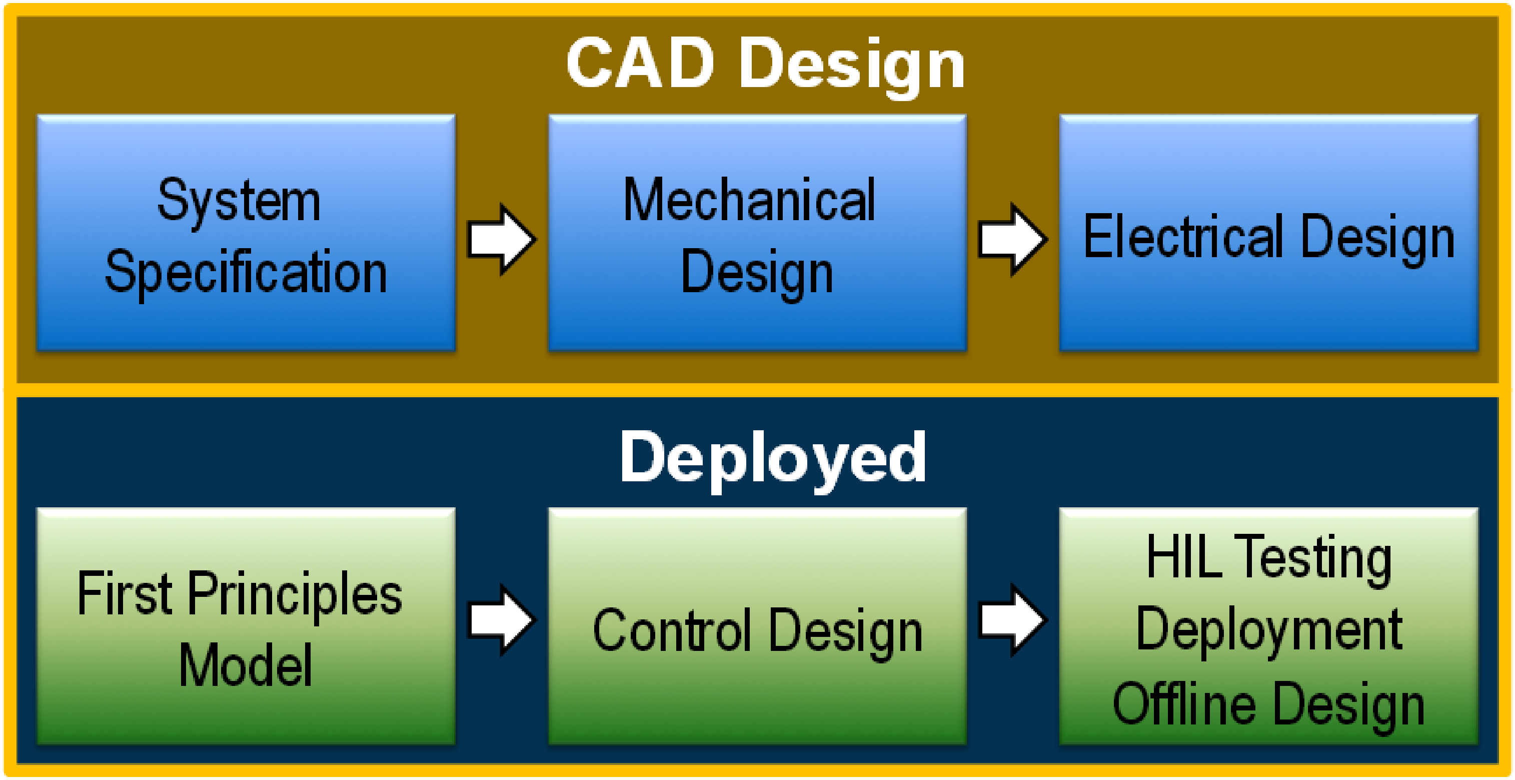
AUTODESK

NATIONAL INSTRUMENTS™

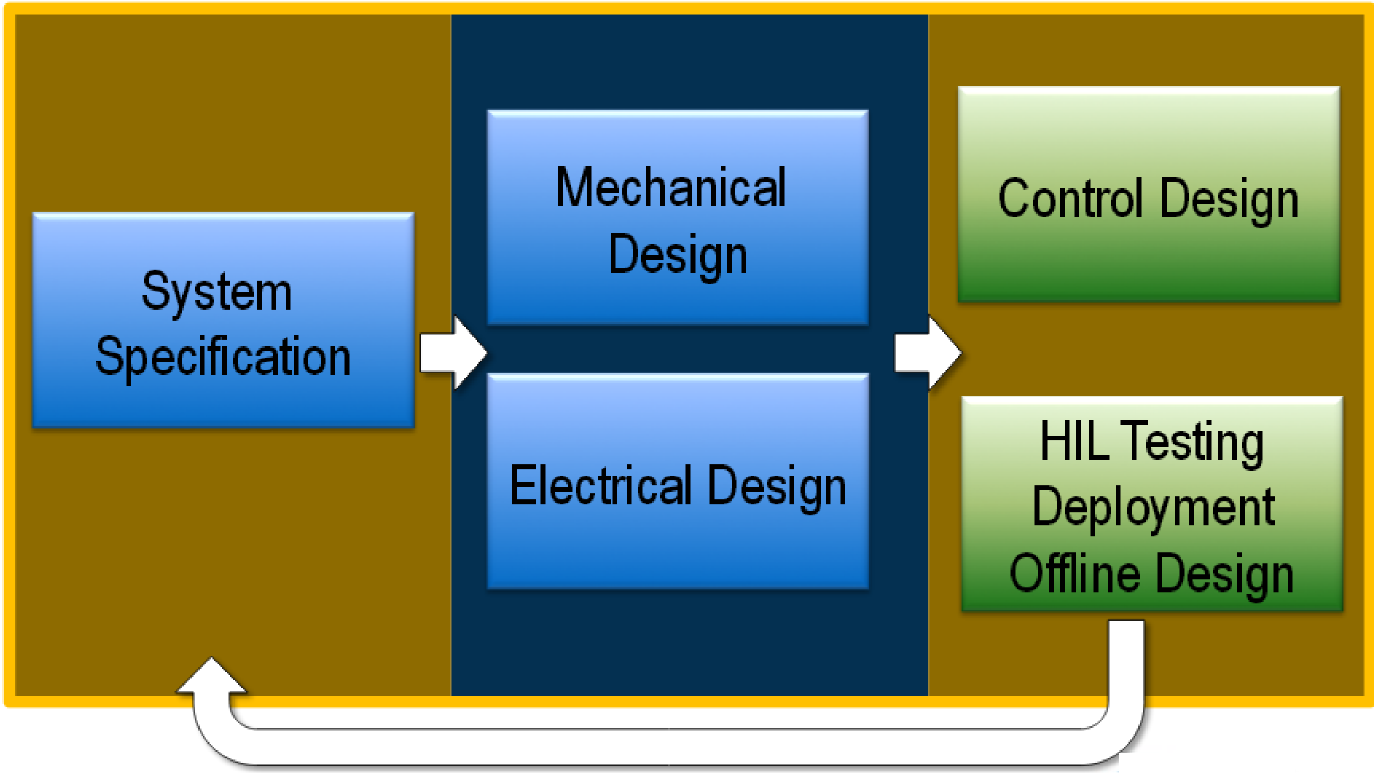
JACOBS INSTITUTE FOR DESIGN INNOVATION
COLLEGE OF ENGINEERING, UC BERKELEY

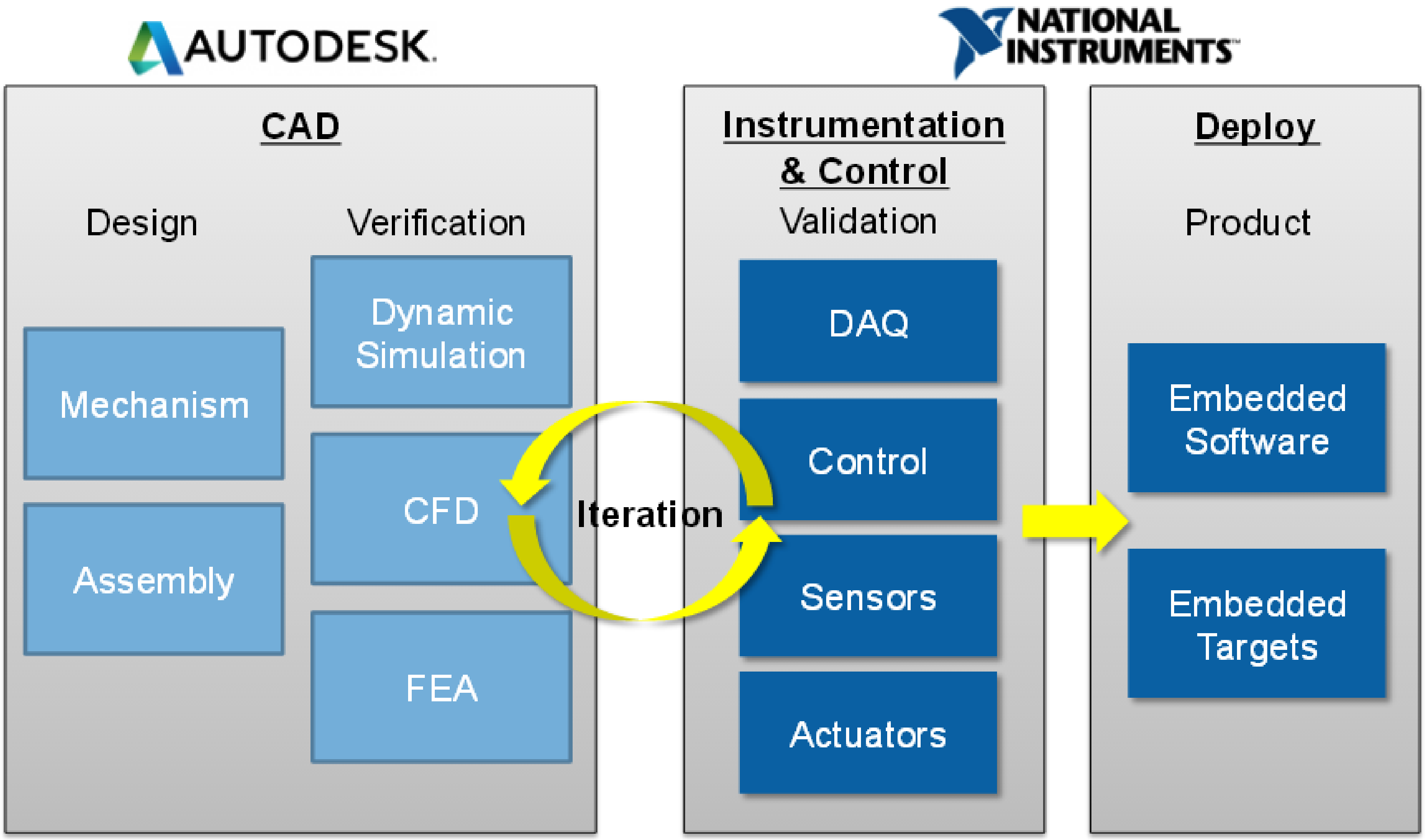
Berkeley Engineering

CURRENT FRAMEWORK:



PROPOSED FRAMEWORK:





AUTODESK

NATIONAL INSTRUMENTS

CAD

Design

- Mechanism
- Assembly

Verification

- Dynamic Simulation
- CFD
- FEA

Iteration

Instrumentation & Control

Validation

- DAQ
- Control
- Sensors
- Actuators

Deploy

Product

- Embedded Software
- Embedded Targets

AUTODESK

NATIONAL INSTRUMENTS

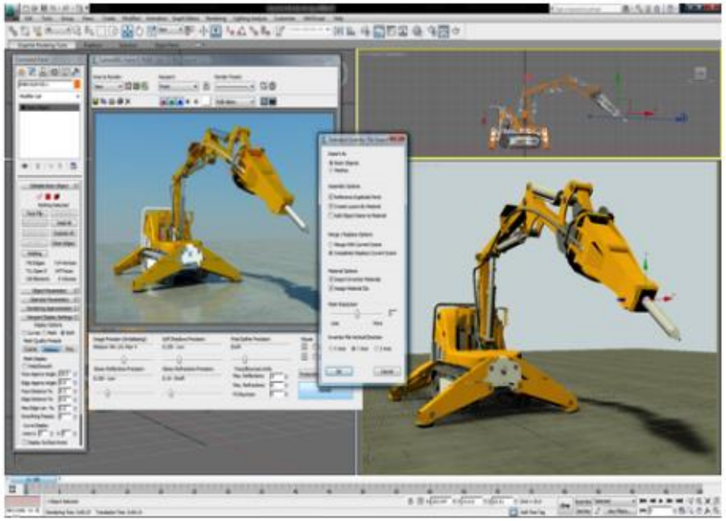
JACOBS INSTITUTE FOR DESIGN INNOVATION
COLLEGE OF ENGINEERING, UC BERKELEY

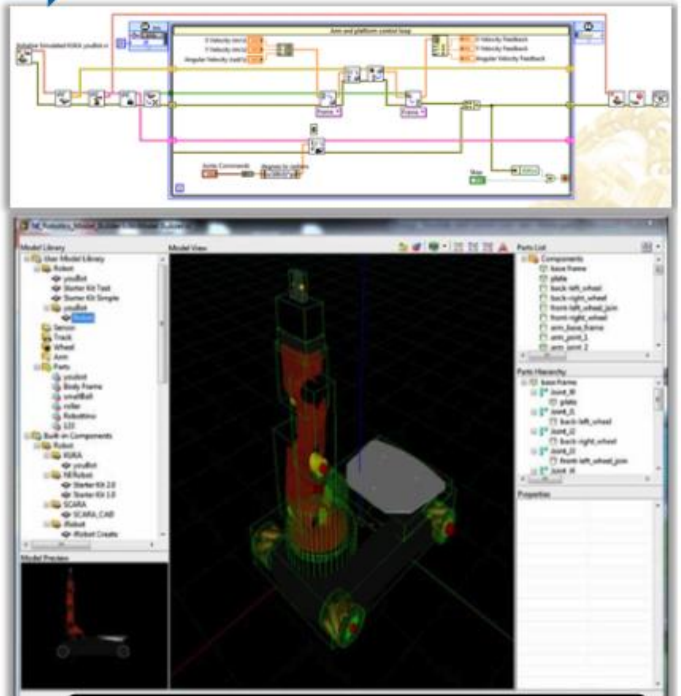
Berkeley Engineering

Dynamic Simulation and Control

Export geometry and mass properties information

AutoDesk Inventor Model → Geometry and Mass Properties → NI LabVIEW Control





Virtual Mechanical design

Control system design and hardware interaction

AUTODESK

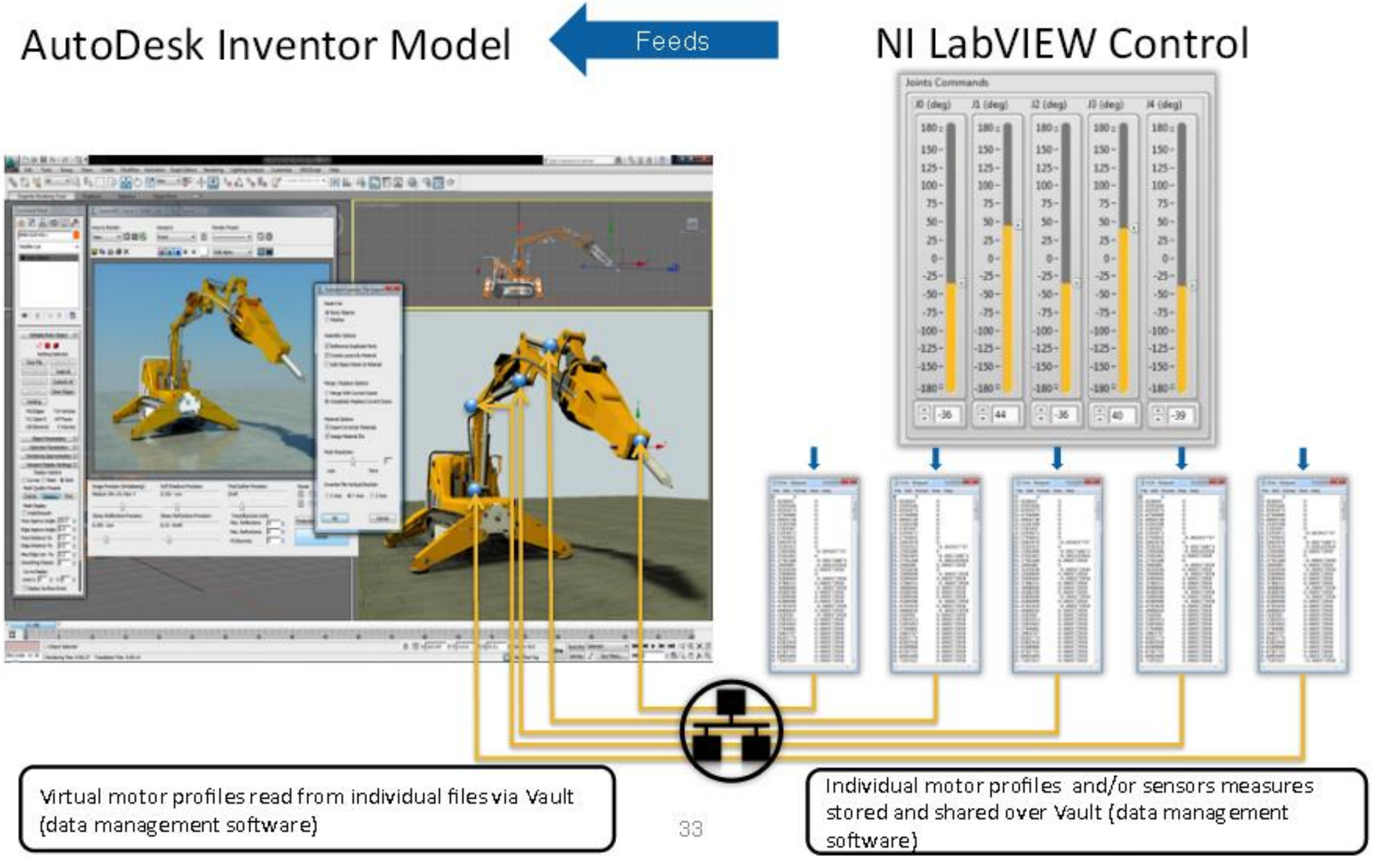
NATIONAL INSTRUMENTS™

JACOBS INSTITUTE FOR DESIGN INNOVATION
COLLEGE OF ENGINEERING, UC BERKELEY

Berkeley Engineering

Dynamic Simulation and Control

Import control inputs



AutoDesk Inventor Model ← Feeds — NI LabVIEW Control

Virtual motor profiles read from individual files via Vault (data management software)

Individual motor profiles and/or sensors measures stored and shared over Vault (data management software)

2016 Master of Engineering Capstone Project:

- Define a minimum information set required to enter at the 3D modeling step to accomplish successful implementation of a physical system

- Develop a test bed to validate the approach from 3D modeling of the mechanical system, simulate the system, build the actual system, validate and test

- End goal is to develop a system worthy of demonstration at NI Week 2017 in Austin.

2015 Master of Engineering Capstone Project:

- Open to all Departments
- Skills preferred but not required:
  - Modelling
  - Dynamics and Controls
  - Mechatronics
  - Simulation
  - Real-time programming
  - Autodesk Inventor, Modelica, LabVIEW
- Project supported by Autodesk and National Instruments

AUTODESK. NATIONAL INSTRUMENTS™ JACOBS INSTITUTE FOR DESIGN INNOVATION COLLEGE OF ENGINEERING, UC BERKELEY Berkeley Engineering

UC Berkeley

Mechanical Engineering Department

Undergraduate Control Courses :

ME132 : Dynamic Systems and Feedback

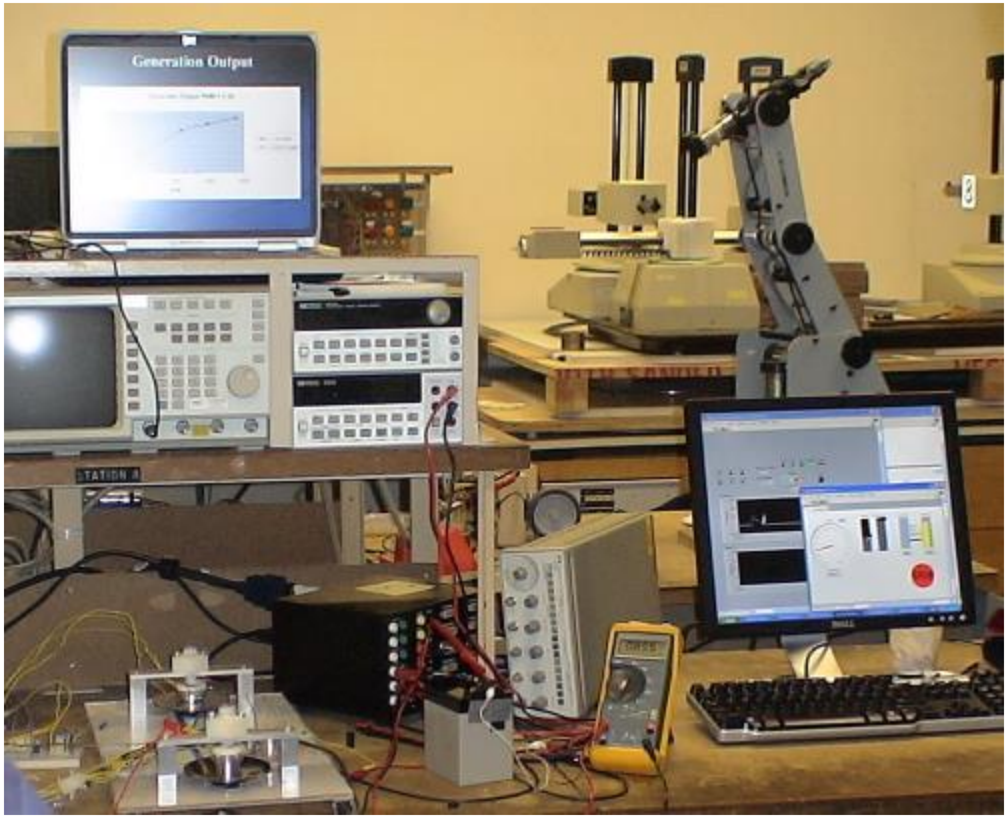
ME134 : Automatic Control Systems

Follow On Project Courses :

ME102B : Mechatronics Design

ME135 : Design of Microprocessor-based Mechanical Systems

ME134 : Automatic Control Systems (4 Units)

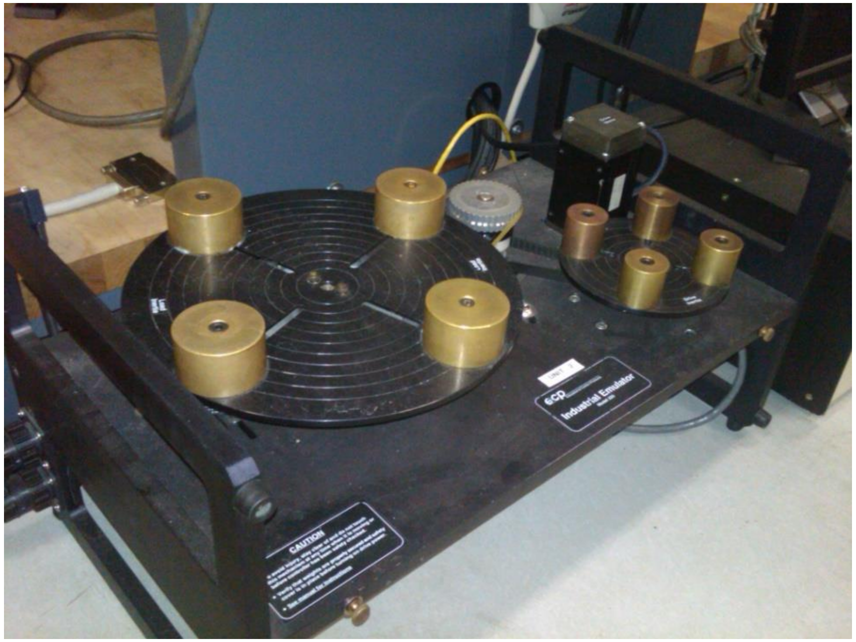


3 hours of lecture
1 hour of discussion per week,
and 3 hours of laboratory every other week.

Prerequisites: 132.

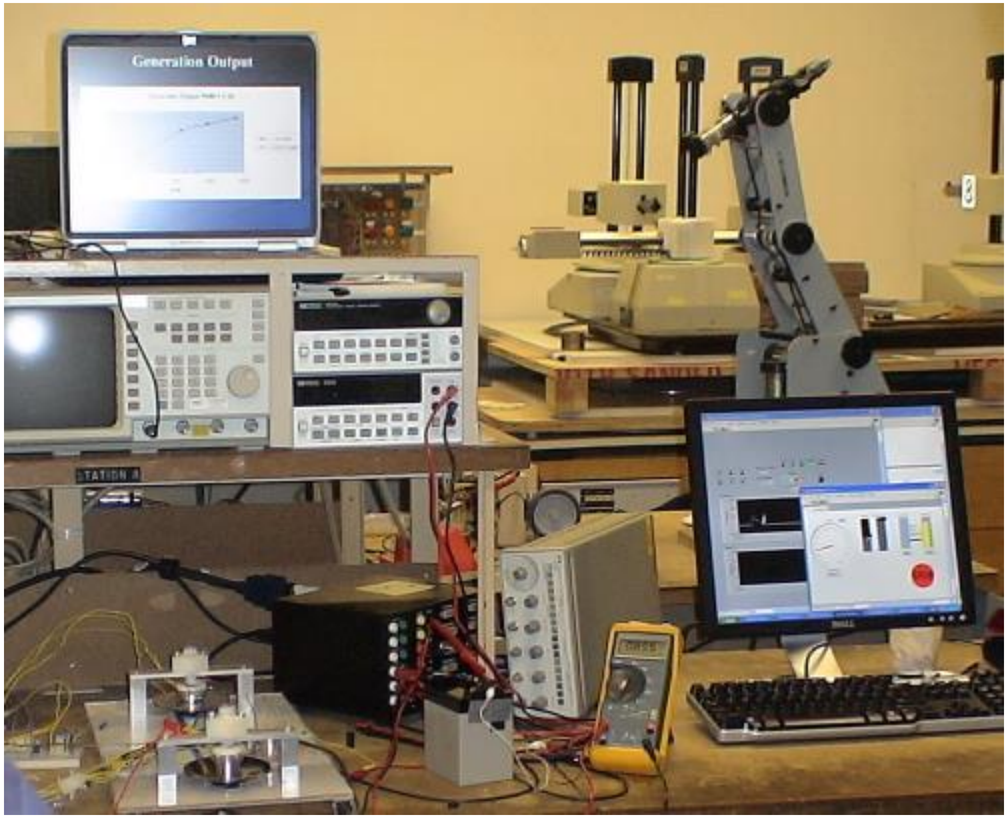
Linear control systems analysis and design in transform domain and time domain. Transfer functions and state equations. Frequency response and Nyquist stability. Loop shaping. State feedback controller and observer design. Applications to mechanical and mechatronics systems. Computer control.

Traditional Laboratory Hardware:



ECP Rotational Inertia and Flexible Drive System

ME134 : Automatic Control Systems Laboratory



Equipment Used:

- Real-time Desktop with NI-PCI7833R
- Luminary Micro LM3S8962
- NI-cRIO
- NI-sbRIO

Software Used:

- LabVIEW 2009
- LabVIEW Embedded for ARM
- LabVIEW Real-time
- LabVIEW FPGA

Statistics for ME134 :

Number of Students: 20-25 per year

70 % Seniors
20 % Juniors
10 % Graduate Students

95 % Mechanical Engineers

Course taught every other year

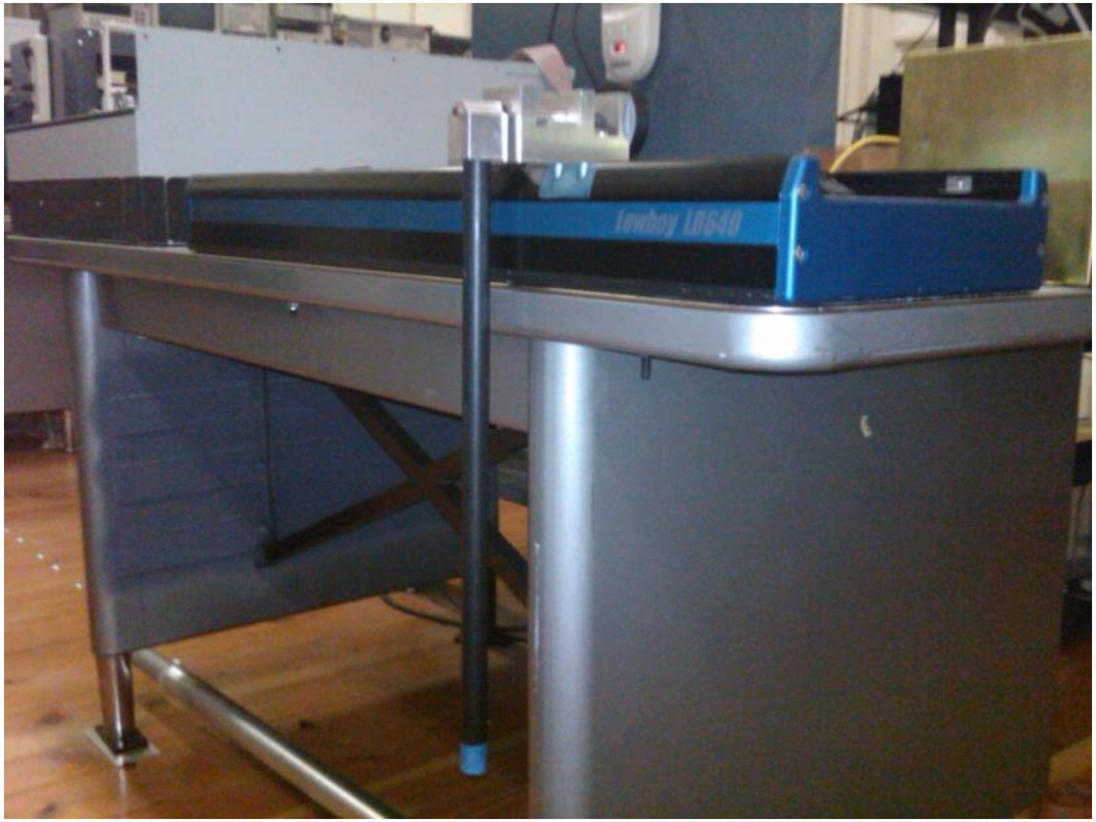
Course objectives

- Introduce and familiarize students with dynamic systems modeling and analysis techniques that can be employed on a large variety of engineering systems.

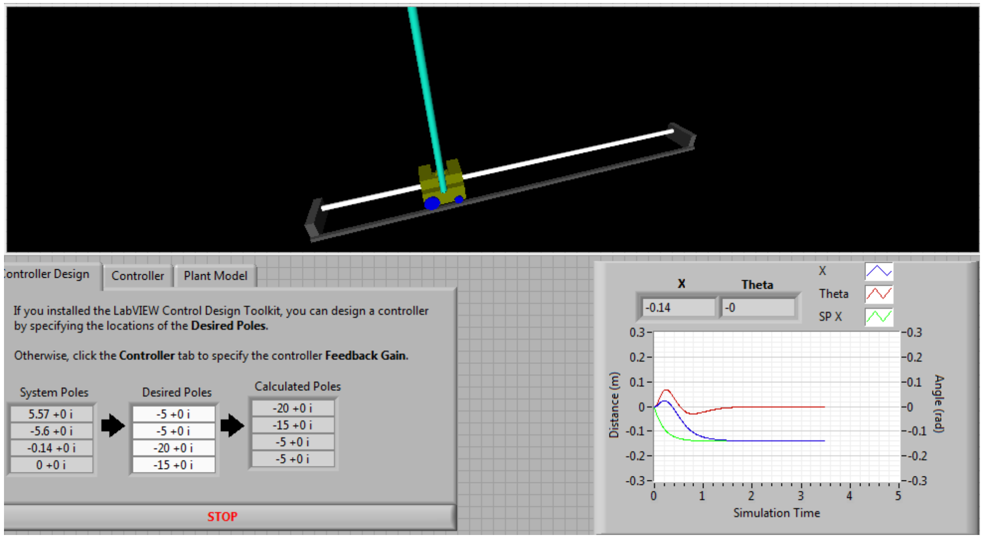
- Introduce and familiarize students with control systems design techniques.

- Provide students with a hands-on laboratory experience on modeling, controller design and implementation of a DC-motor positioning and velocity control system.

The role of LabVIEW and the use of the sbRIO in solving the classical inverted pendulum problem



Inverted Pendulum Hardware System



If you installed the LabVIEW Control Design Toolkit, you can design a controller by specifying the locations of the **Desired Poles**.

Otherwise, click the **Controller** tab to specify the controller **Feedback Gain**.

System Poles	Desired Poles	Calculated Poles
5.57 +0 i	-5 +0 i	-20 +0 i
-5.6 +0 i	-5 +0 i	-15 +0 i
-0.14 +0 i	-20 +0 i	-5 +0 i
0 +0 i	-15 +0 i	-5 +0 i

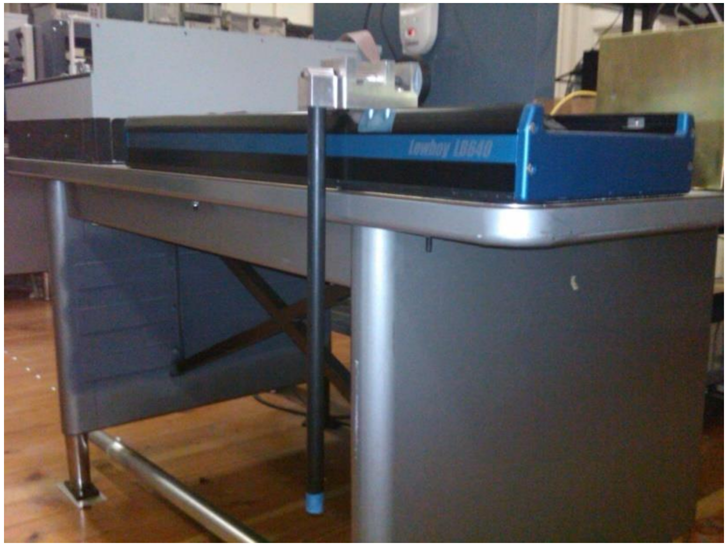
X: -0.14
Theta: -0

Stage: Belleverman LOWBOY 260

Motor: Trilogy Direct Drive
Linear Brushless Motor

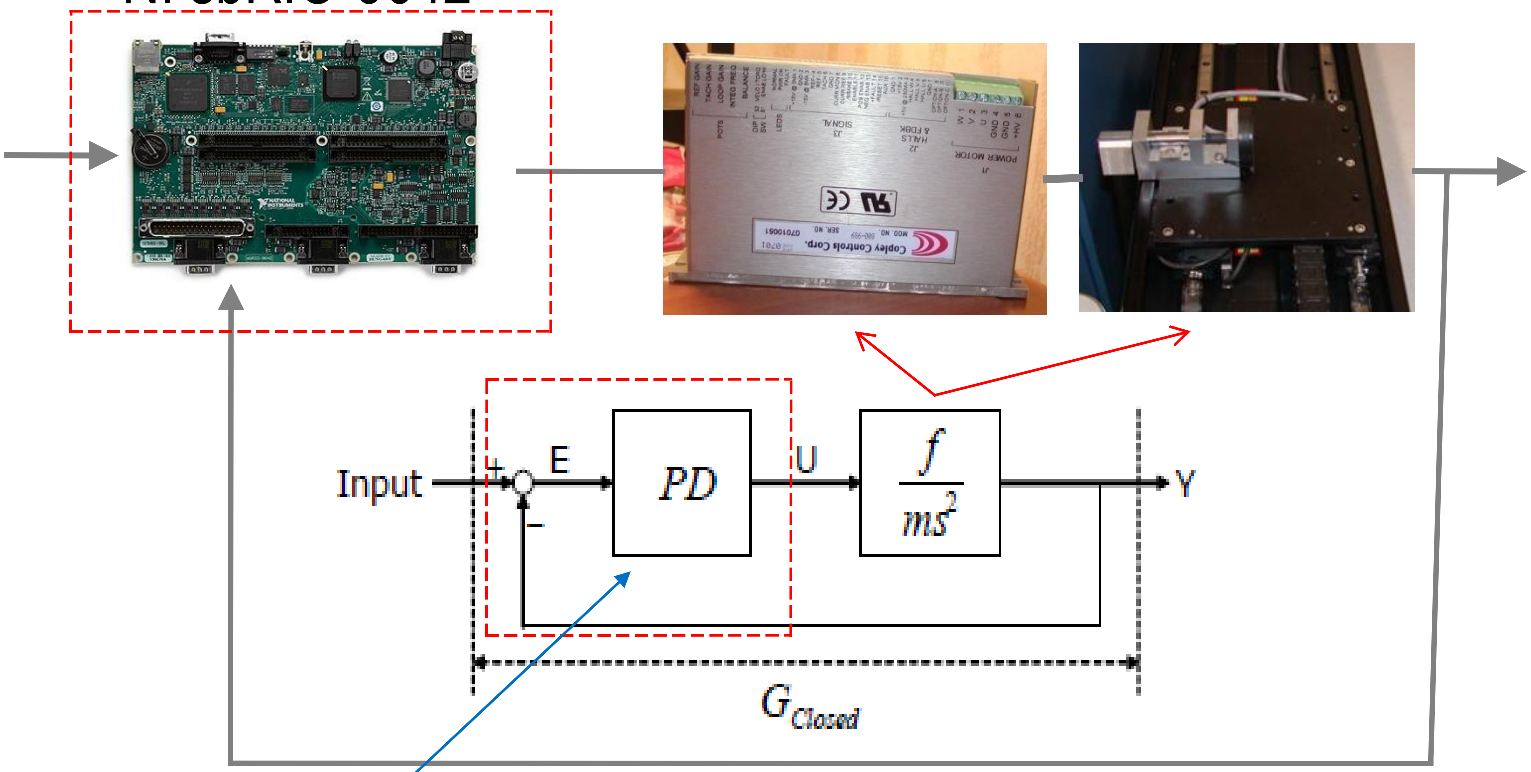
Drive: Copley Controls
Brushless DC Amplifier

Encoder: Renishaw RG22H
1 um resolution



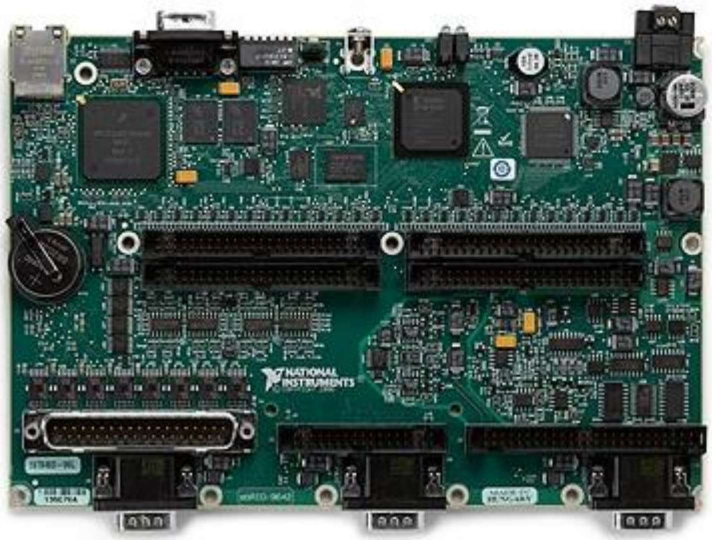
System Set Up

NI sbRIO-9642



Add a simple PD Loop

NI sbRIO-9642



Features:

- 400 MHz Processor
- 2M gate FPGA
- 32 16 bit Analog Input
- 4 16 bit Analog Output
- 32 Digital Output
- 32 Digital Input
- 10/100 Base-T Ethernet port
- RS232 serial port

Controller Configuration Used:

Hardware:

Standard PC Host running LabVIEW

NI sbRIO-9642

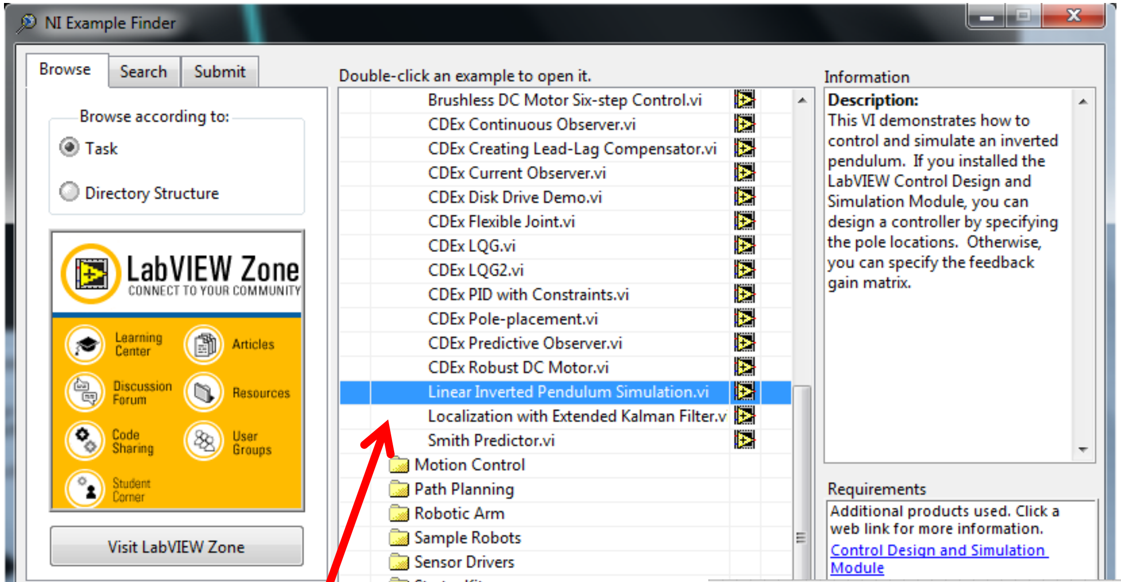
Software:

LabVIEW : HMI
Data Acquisition
Graphical Interface

LabVIEW Realtime:
State Feedback
Mathscript Node

LabVIEW FPGA:
Quadrature Decoding
Analog Output

Inverted Pendulum Simulation



NI Example Finder

Browse | Search | Submit

Browse according to:

● Task

○ Directory Structure

Double-click an example to open it.

- Brushless DC Motor Six-step Control.vi
- CDEx Continuous Observer.vi
- CDEx Creating Lead-Lag Compensator.vi
- CDEx Current Observer.vi
- CDEx Disk Drive Demo.vi
- CDEx Flexible Joint.vi
- CDEx LQG.vi
- CDEx LQG2.vi
- CDEx PID with Constraints.vi
- CDEx Pole-placement.vi
- CDEx Predictive Observer.vi
- CDEx Robust DC Motor.vi
- Linear Inverted Pendulum Simulation.vi
- Localization with Extended Kalman Filter.v
- Smith Predictor.vi
- Motion Control
- Path Planning
- Robotic Arm
- Sample Robots
- Sensor Drivers
- Starter Kit
- Steering
- Vision
- Toolkits and Modules
- Toolkits and Modules Not Installed

Information

Description: This VI demonstrates how to control and simulate an inverted pendulum. If you installed the LabVIEW Control Design and Simulation Module, you can design a controller by specifying the pole locations. Otherwise, you can specify the feedback gain matrix.

Requirements

Additional products used. Click a web link for more information.

Control Design and Simulation Module

□ Include ni.com examples

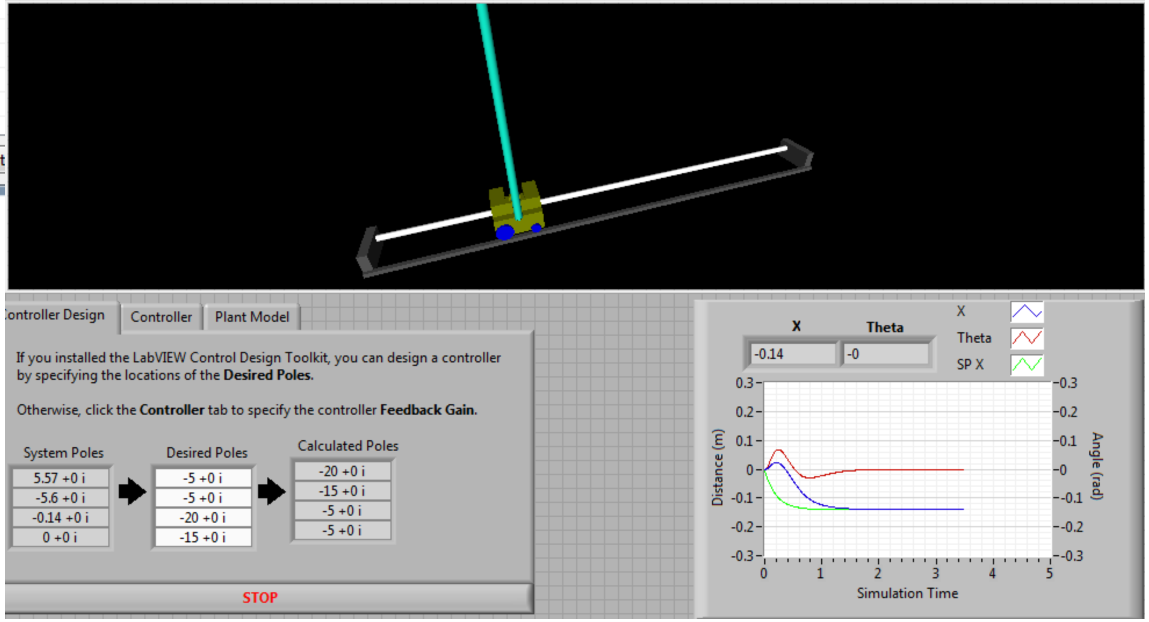
□ ni.com query timeout

Hardware

Find hardware

□ Limit results to hardware

Add t



Controller Design | Controller | Plant Model

If you installed the LabVIEW Control Design Toolkit, you can design a controller by specifying the locations of the **Desired Poles.**

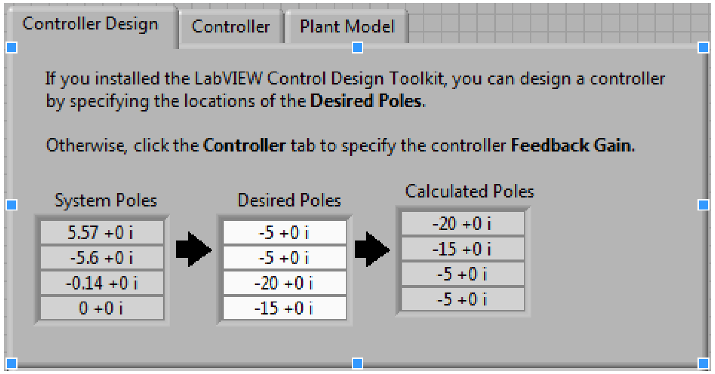
Otherwise, click the **Controller** tab to specify the controller **Feedback Gain.**

System Poles	Desired Poles	Calculated Poles
5.57 +0 i	-5 +0 i	-20 +0 i
-5.6 +0 i	-5 +0 i	-15 +0 i
-0.14 +0 i	-20 +0 i	-5 +0 i
0 +0 i	-15 +0 i	-5 +0 i

STOP

X: -0.14 Theta: -0

Taken right out of LabVIEW Example

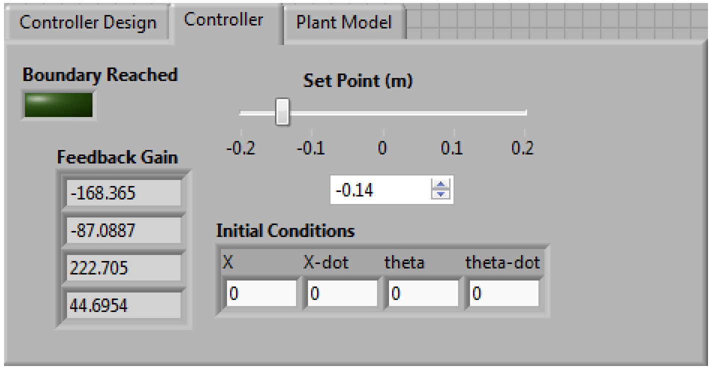


If you installed the LabVIEW Control Design Toolkit, you can design a controller by specifying the locations of the **Desired Poles**.

Otherwise, click the **Controller** tab to specify the controller **Feedback Gain**.

System Poles	Desired Poles	Calculated Poles
5.57 +0 i	-5 +0 i	-20 +0 i
-5.6 +0 i	-5 +0 i	-15 +0 i
-0.14 +0 i	-20 +0 i	-5 +0 i
0 +0 i	-15 +0 i	-5 +0 i

Controller Design – Student Specify Desired Poles



Boundary Reached

Set Point (m) -0.14

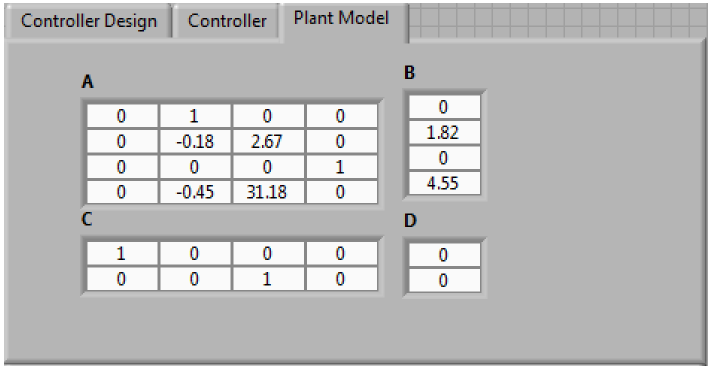
Feedback Gain

-168.365
-87.0887
222.705
44.6954

Initial Conditions

X	X-dot	theta	theta-dot
0	0	0	0

Controller Gains – Student can directly implement gains on actual system



A

0	1	0	0
0	-0.18	2.67	0
0	0	0	1
0	-0.45	31.18	0

B

0
1.82
0
4.55

C

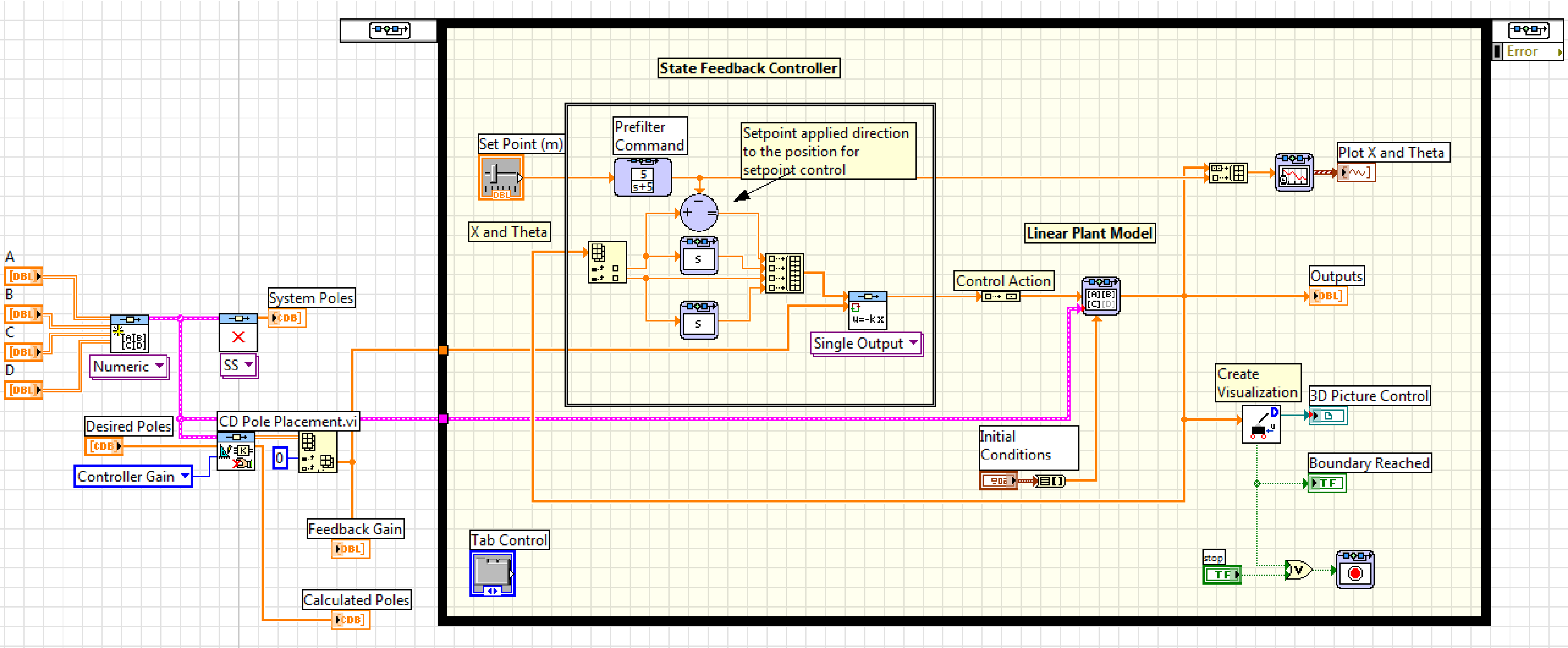
1	0	0	0
0	0	1	0

D

0
0

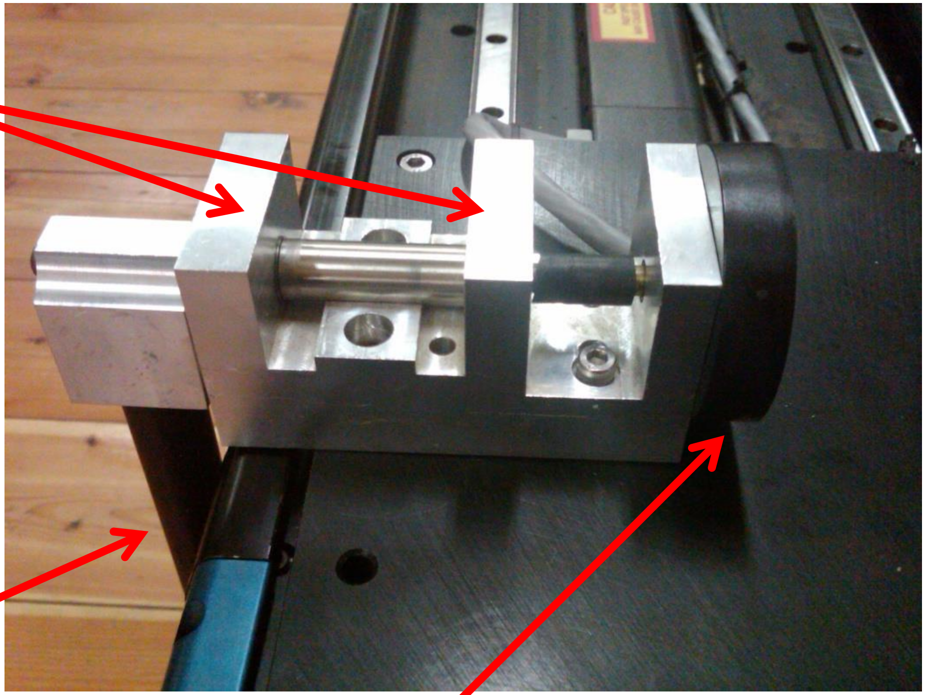
Plant model – Student obtain matrix coefficient through system identification

Block Diagram for Inverted Pendulum Simulation



NATIONAL INSTRUMENTS

Pendulum Subsystem :



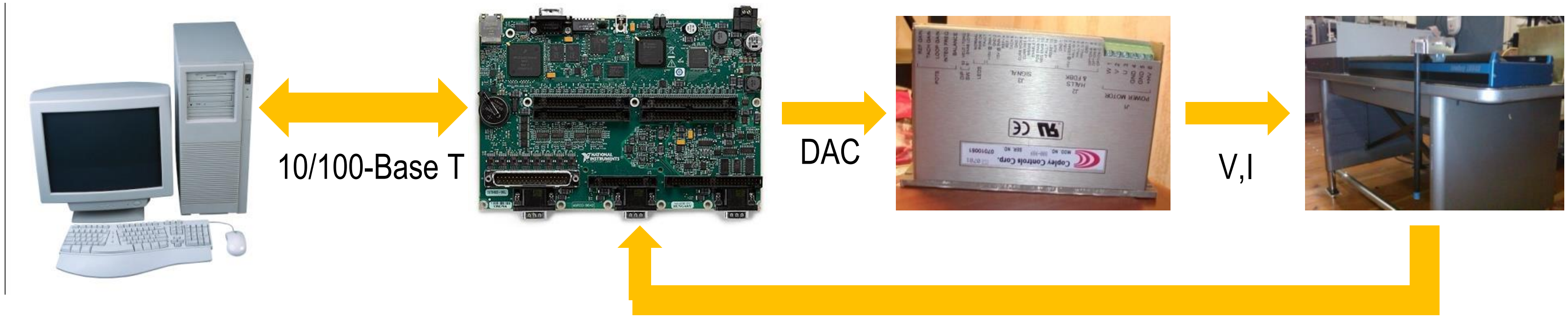
Dual Ball-Bearing Support

Carbon Composite Pendulum Rod

US Digital - 512 lines incremental encoder
2048 counts/rev after quadrature

System Architecture:

NI sbRIO-9642



10/100-Base T

DAC

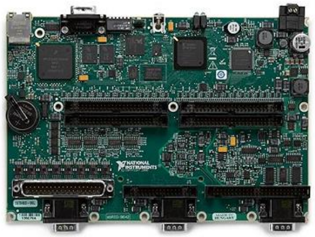
V,I

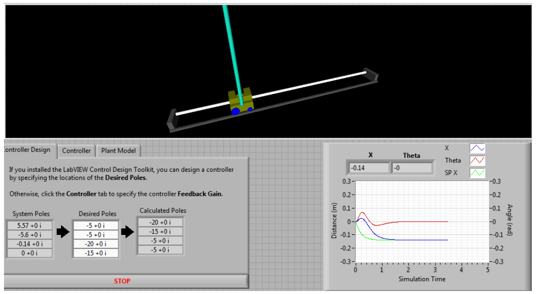
Standard PC Development Platform:
- LabVIEW Development

Stage Position
Pendulum Angle

Software Architecture:

NI sbRIO-9642





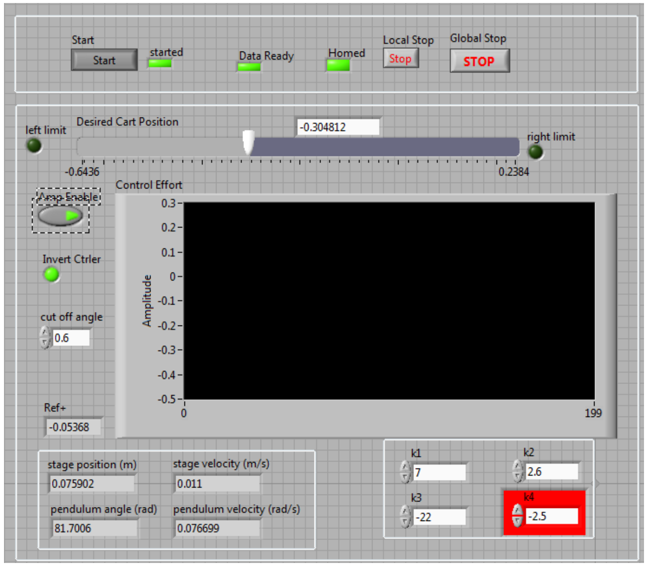
If you installed the LabVIEW Control Design Toolkit, you can design a controller by specifying the locations of the **Desired Poles.**

Otherwise, click the **Controller** tab to specify the controller **Feedback Gain.**

System Poles	Desired Poles	Calculated Poles
5.57 +0 i	-5 +0 i	-20 +0 i
-5.6 +0 i	-5 +0 i	-15 +0 i
-0.14 +0 i	-20 +0 i	-5 +0 i
0 +0 i	-15 +0 i	-5 +0 i

X: -0.14
Theta: -0

LabVIEW 2009
- Simulation
- Data Acquisition



Desired Cart Position: -0.304812 (range -0.6436 to 0.2384)

cut off angle: 0.6

Ref+: -0.05368

stage position (m): 0.075902

stage velocity (m/s): 0.011

pendulum angle (rad): 81.7006

pendulum velocity (rad/s): 0.076699

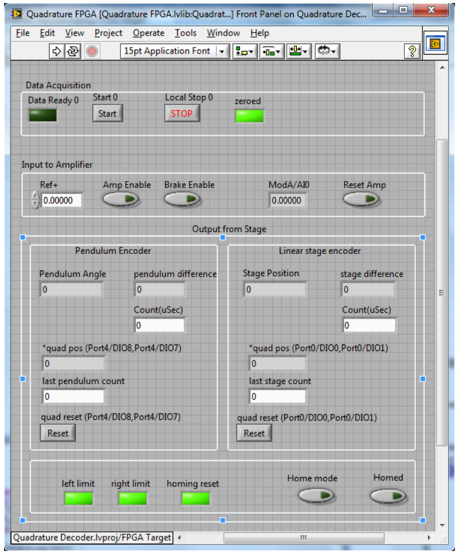
k1: 7

k2: 2.6

k3: -22

k4: -2.5

LabVIEW Real-time
- State Feedback
- Self Erecting Mathscript Node



Ref+: 0.00000

ModA/AI0: 0.00000

Pendulum Angle: 0

pendulum difference: 0

Count(uSec): 0

*quad pos (Port4/DIO8,Port4/DIO7): 0

last pendulum count: 0

quad reset (Port4/DIO8,Port4/DIO7)

Stage Position: 0

stage difference: 0

Count(uSec): 0

*quad pos (Port0/DIO0,Port0/DIO1): 0

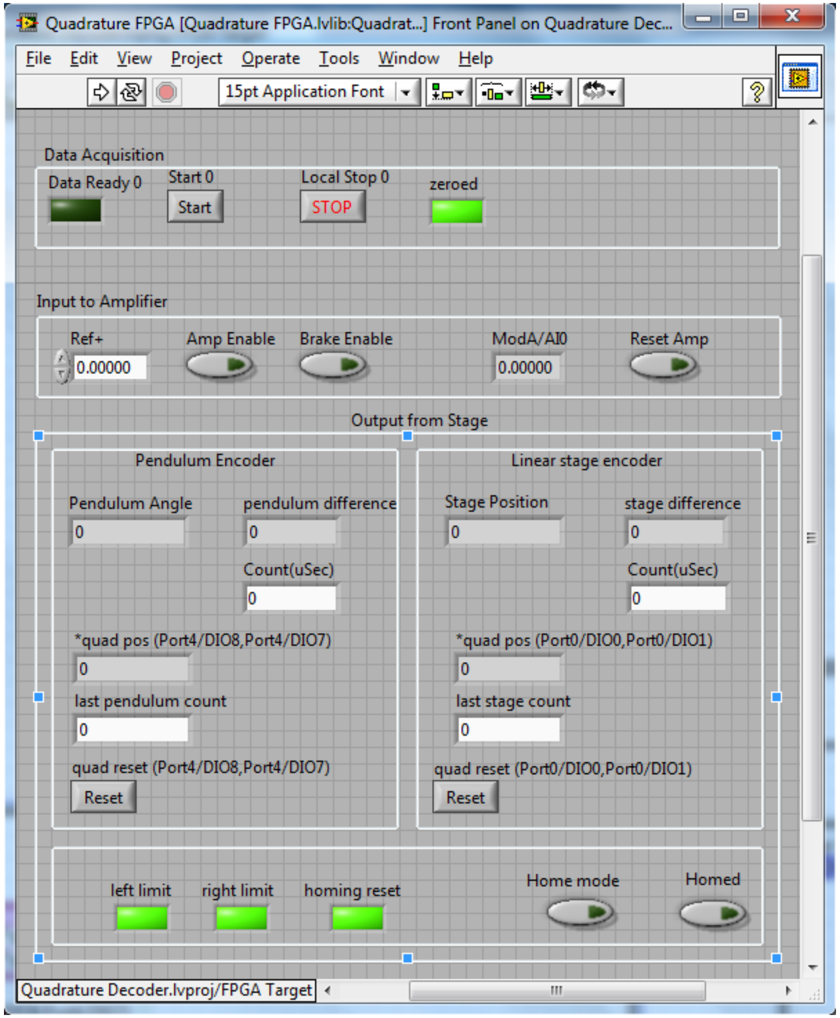
last stage count: 0

quad reset (Port0/DIO0,Port0/DIO1)

LabVIEW FPGA
- Quadrature Decode
- Filtering
- Control Output

LabVIEW FPGA

- •Quadrature Decode
- • Filtering
- •Control Output



Quadrature FPGA [Quadrature FPGA.lvlib:Quadrat...] Front Panel on Quadrature Dec...

Data Acquisition

Data Ready 0 | Start 0 | Local Stop 0 | zeroed

Input to Amplifier

Ref+: 0.00000
Amp Enable
Brake Enable
ModA/AI0: 0.00000
Reset Amp

Output from Stage

Pendulum Encoder

Pendulum Angle: 0
pendulum difference: 0
Count(uSec): 0
*quad pos (Port4/DIO8,Port4/DIO7): 0
last pendulum count: 0
quad reset (Port4/DIO8,Port4/DIO7)

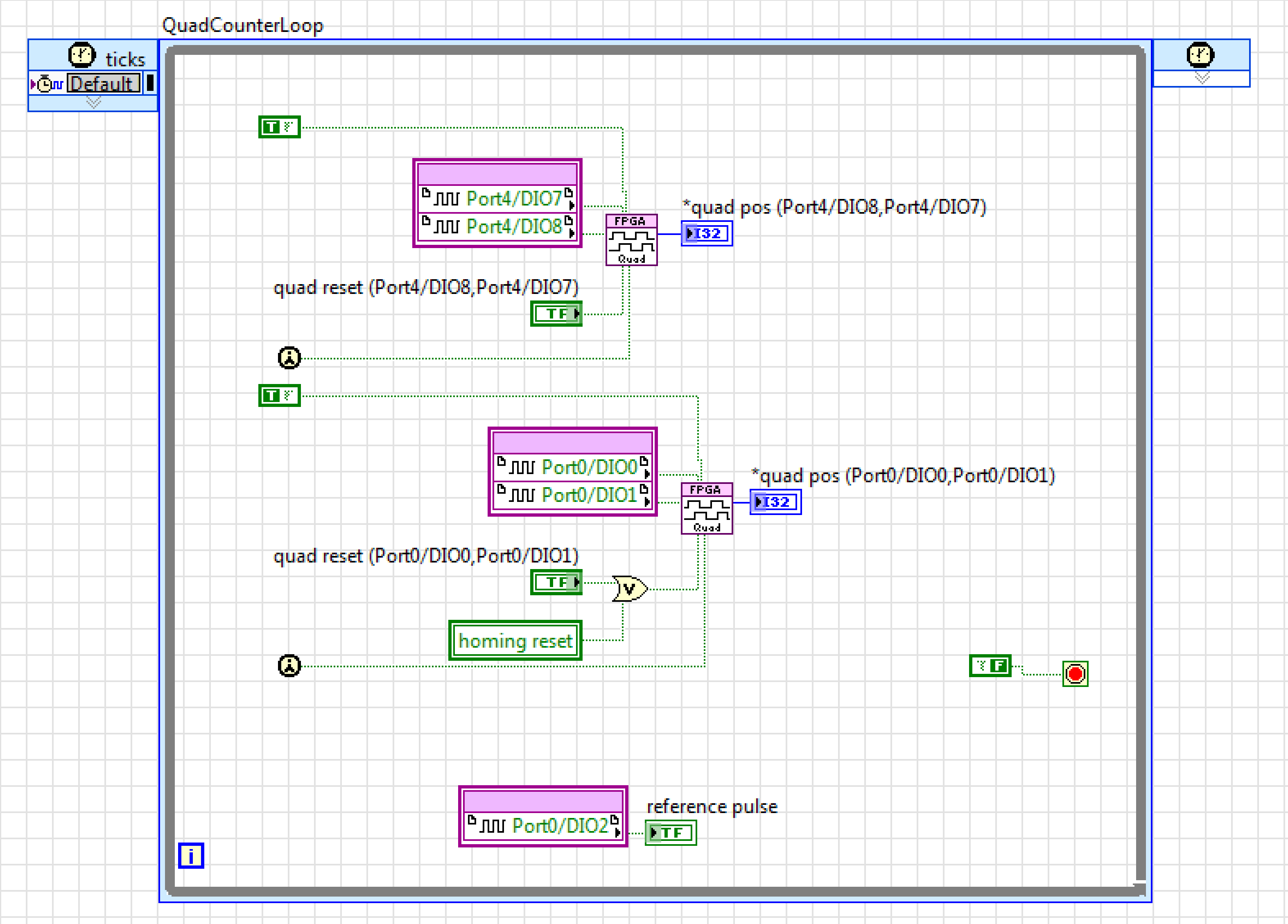
Linear stage encoder

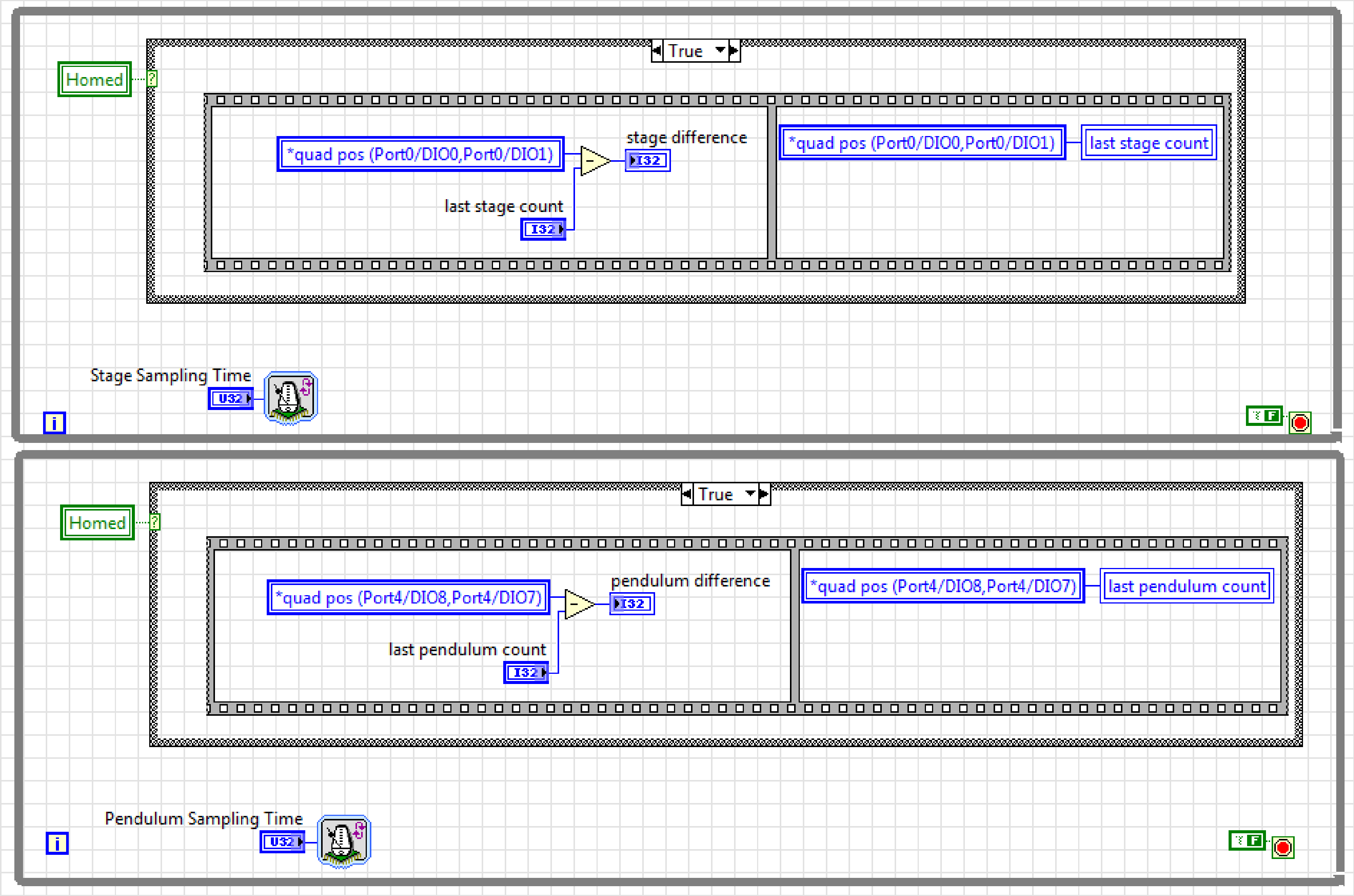
Stage Position: 0
stage difference: 0
Count(uSec): 0
*quad pos (Port0/DIO0,Port0/DIO1): 0
last stage count: 0
quad reset (Port0/DIO0,Port0/DIO1)

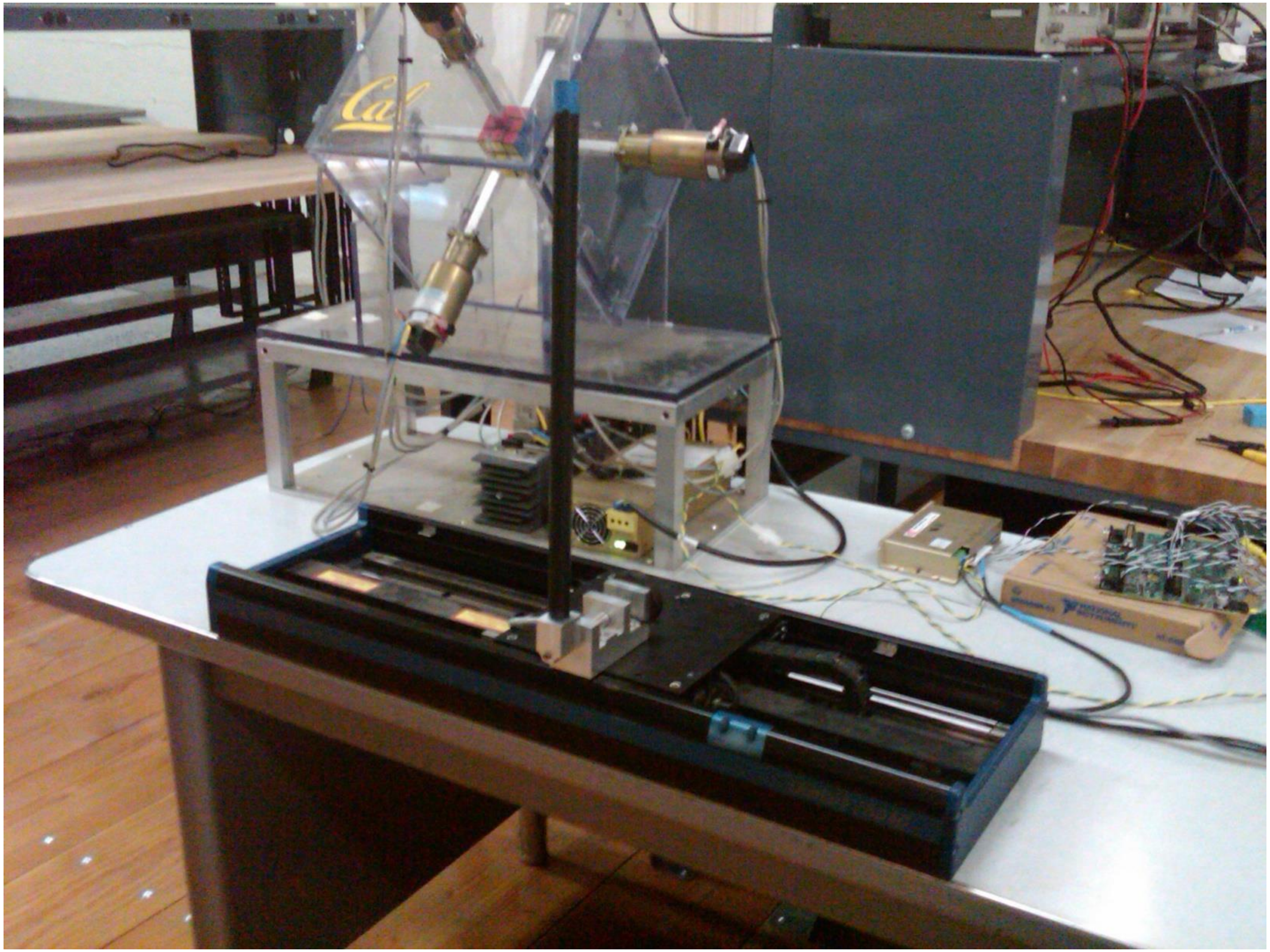
left limit | right limit | homing reset | Home mode | Homed

Quadrature Decoder.lvproj/FPGA Target

Quadrature Decode:


QuadCounterLoop
ticks
Default
Port4/DIO7
Port4/DIO8
FPGA
Quad
*quad pos (Port4/DIO8,Port4/DIO7)
I32
quad reset (Port4/DIO8,Port4/DIO7)
TF
Port0/DIO0
Port0/DIO1
*quad pos (Port0/DIO0,Port0/DIO1)
quad reset (Port0/DIO0,Port0/DIO1)
homing reset
Port0/DIO2
reference pulse
i


True
Homed
*quad pos (Port0/DIO0,Port0/DIO1)
stage difference
I32
last stage count
I32
*quad pos (Port0/DIO0,Port0/DIO1)
last stage count
Stage Sampling Time
U32
True
Homed
*quad pos (Port4/DIO8,Port4/DIO7)
pendulum difference
I32
last pendulum count
I32
*quad pos (Port4/DIO8,Port4/DIO7)
last pendulum count
Pendulum Sampling Time
U32



Cal
NATIONAL INSTRUMENTS

ME 134 Fall 2009: Inverted Pendulum

By:

Brian Phegley

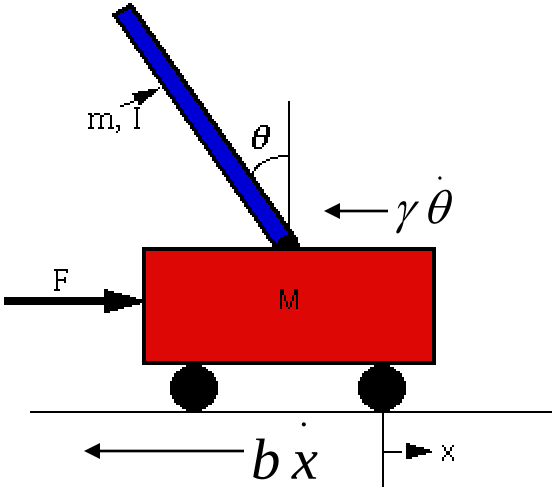
Kevin Ding

$b\dot{x}$

Theory: Physical System

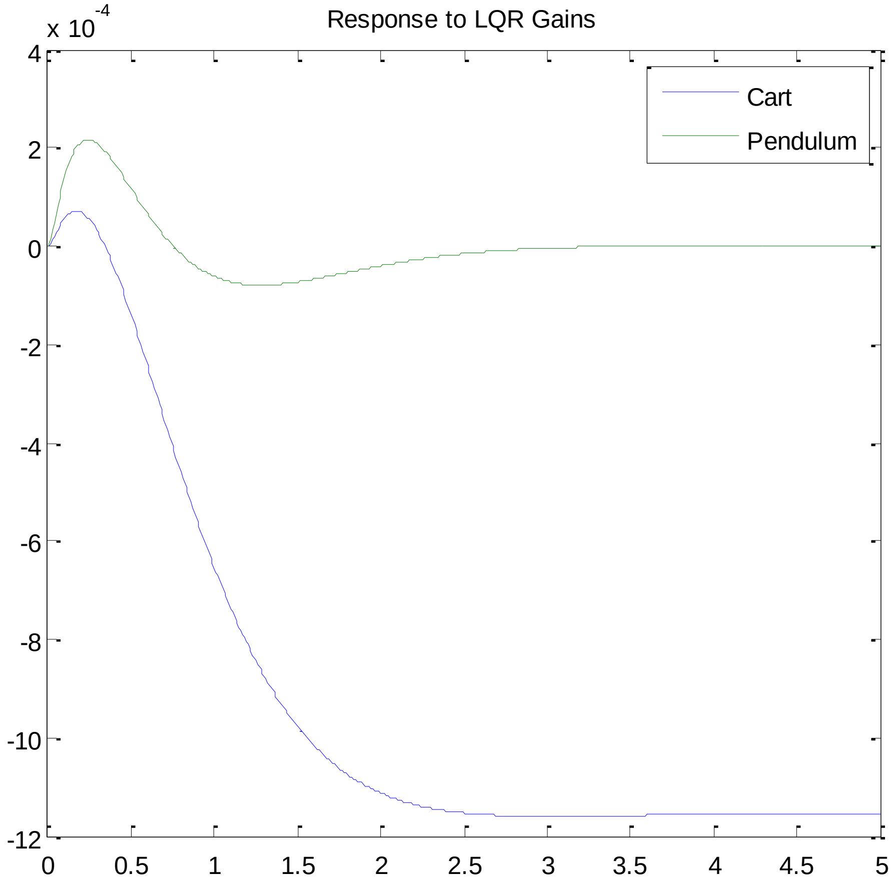
$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-b(I+ml^2)}{I(M+m)+Mml^2} & \frac{gm^2l^2}{I(M+m)+Mml^2} & \frac{\gamma(I+2ml^2)}{I(M+m)+Mml^2} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-bml}{I(M+m)+Mml^2} & \frac{gml(M+m)}{I(M+m)+Mml^2} & \frac{\gamma l(2m+M)}{I(M+m)+Mml^2} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} F$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$$



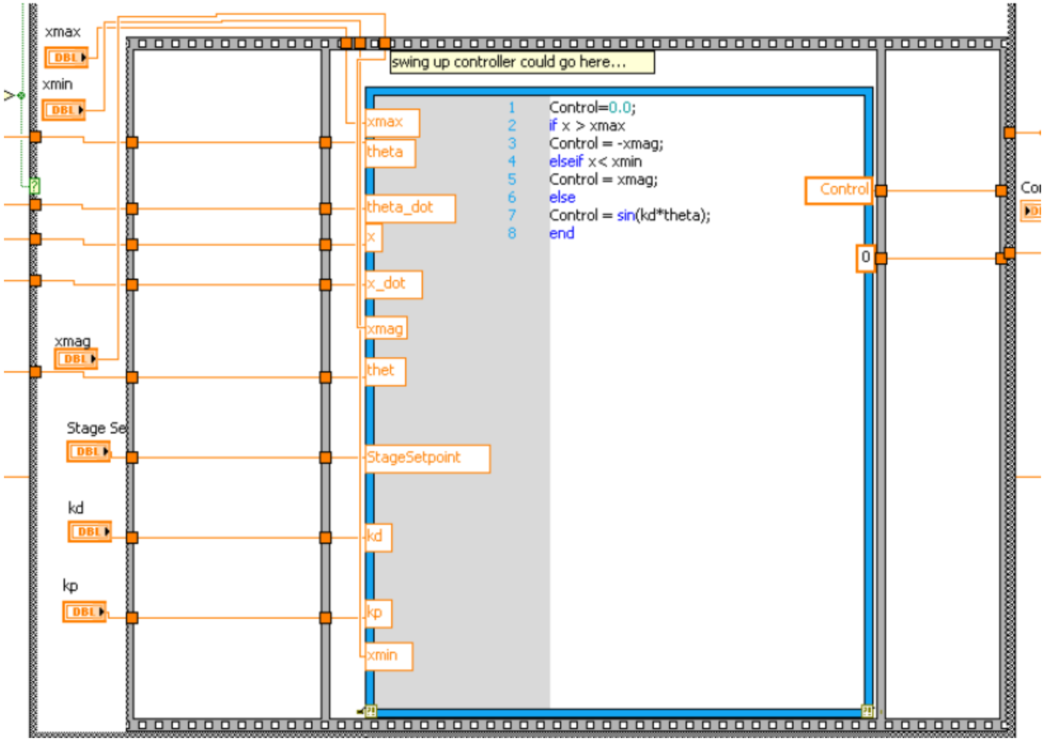
Design

- Initial LQR Implementation
- Performance vs. Stability (High k1 & k2 Gains)



Self-Erecting

```
if ll
      Control=1;
elseif rl
      Control=-1;
elseif mod(theta-pi,2*pi)<2/3*pi
      Control=-1;
elseif mod(theta-pi,2*pi)>4/3*pi
      Control=1;
end
```



swing up controller could go here...

```
Control=0.0;
if x > xmax
Control = -xmag;
elseif x< xmin
Control = xmag;
else
Control = sin(kd*theta);
end
```

Self-Erecting Single Inverted Pendulum (SESIP)

By

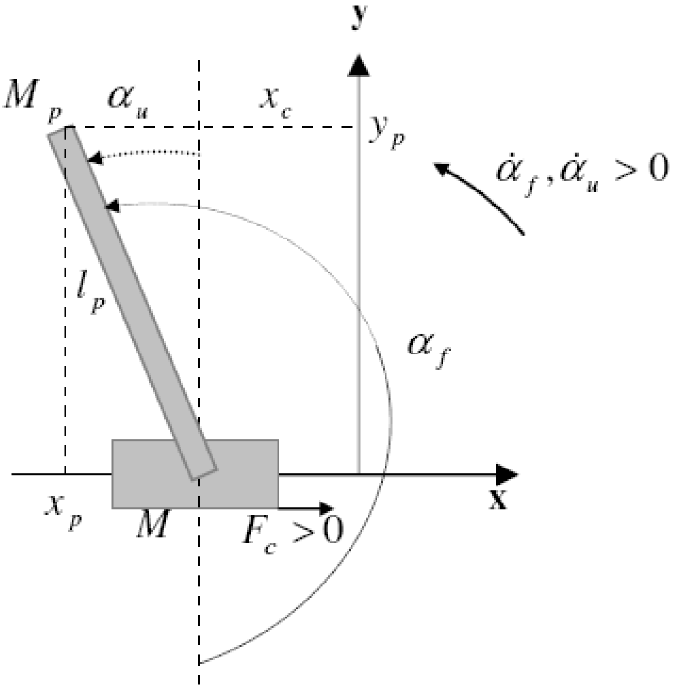
Jonathan Brown

Ben Dokko

Zhen Wang

Shaomin Xiong

Inverted Pendulum



Inverted pendulum schematic

Model Parameters:

Mc = 4.2	Cart Mass (kg)
Mp = 0.106	Pendulum Mass (kg)
Beq = 5	Viscous Damping Coefficient of Cart (N*m*s/rad)
Bp = 0.02	Viscous Damping Coefficient of Pendulum (N*m*s/rad)
L = 0.122	Length to Center of Mass of Pendulum (m)
Ip = 3.85e-3 - Mp*L^2	Moment of Inertia of Pendulum about center of mass (kg*m^2)
g = 9.81	Acceleration due to gravity (m/s^2)

Inverted Pendulum

- State space of the system

$$\dot{\mathbf{x}} = A\,x + B\,u$$
$$y = C\,x$$

where, $x = [x_c \;\; \dot{x}_c \;\; \alpha \;\; \dot{\alpha}]^T$, $u = F_c$ and

$y = [x_c \;\; \alpha]^{\mathrm{T}}$ and matrix A, B and C are given as:

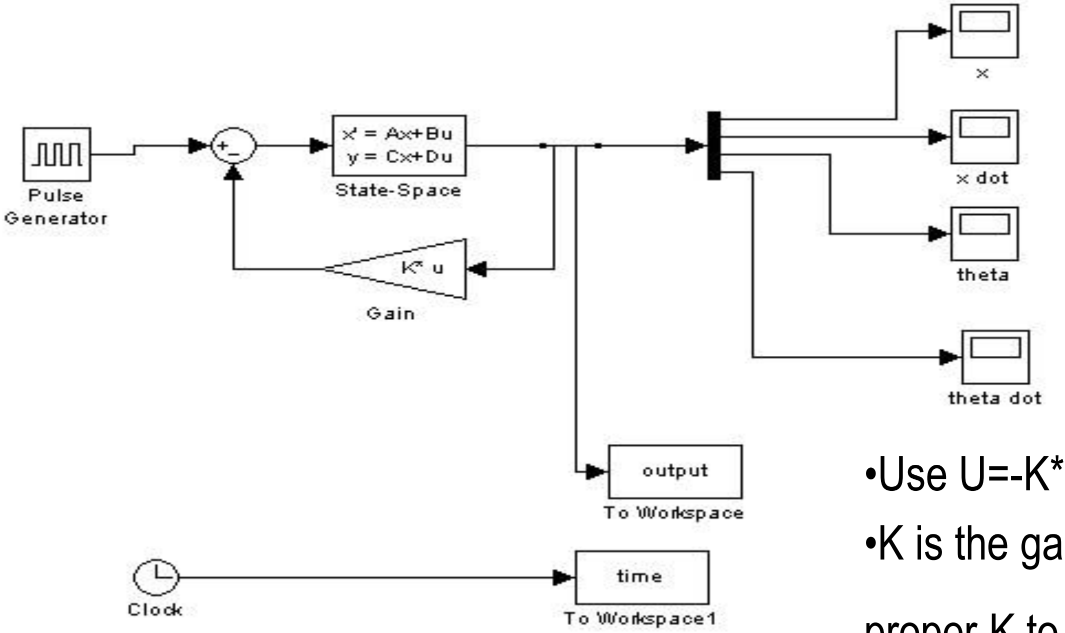
$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -\frac{B_{eq}(M_p l_p{}^2 + I)}{Z} & \frac{(M_p l_p)^2 g}{Z} & -\frac{M_p l_p B_p}{Z} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{M_p l_p B_{eq}}{Z} & \frac{(M_p + M_c) M_p g l_p}{Z} & -\frac{(M_p + M_c) B_p}{Z} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \frac{(I + M_p l_p{}^2)}{Z} \\ 0 \\ \frac{M_p l_p}{Z} \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

where $Z = (M_c + M_p) I_p + M_c M_p l_p{}^2$

Inverted Pendulum

- Simulink scheme:



•Use U=-K*X feedback controller

•K is the gain of feedback and choose

proper K to make the system stable,

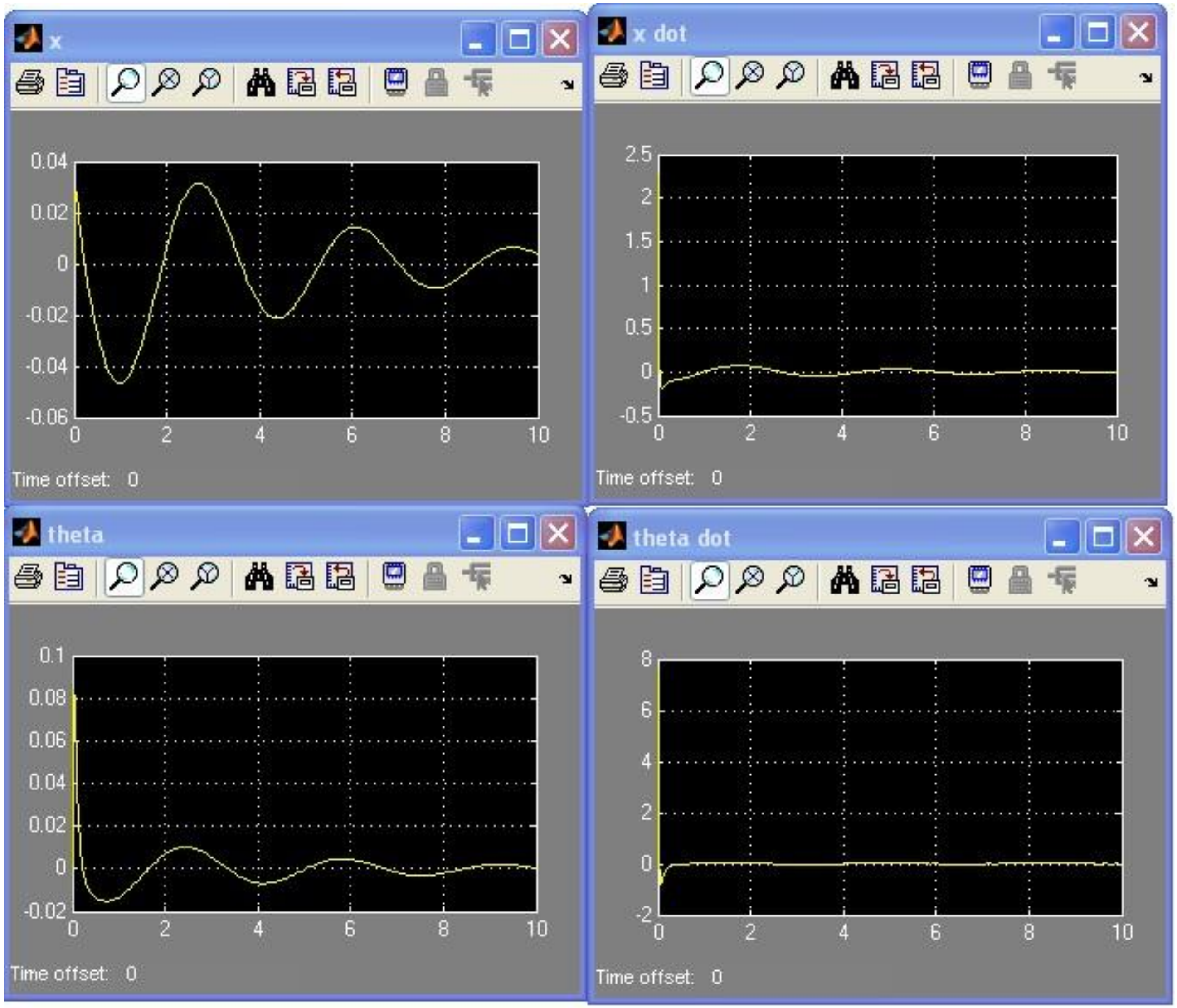
Example:

- Set the desired poles [-59.7501, -13.8837, -0.2338 + 1.8538i, -0.2338 - 1.8538i]

K = 1000 *[-0.3746   -0.1475   1.4750   0.1298]

K_act = [6.35, 2.5, -25, -2.2] ← Actual Implemented Gains (K/59.1)

Impulse response



Swing-up Controller

- Energy pumping method:
  - Pumps energy into the system until pendulum reaches desired energy level. Spring term on the right is to make sure cart doesn’t hit the side walls. Up to 0.25 rad from vertical.
  - Energy of Pendulum:

$$E = M_p g L\left[\frac{2I_p\dot{\theta}^2}{M_p g L} + \cos(\theta) - 1\right]$$

- Controller Output:

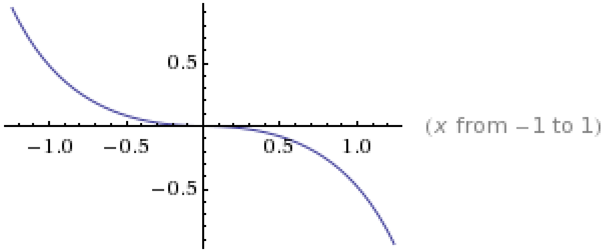
$$V = 3(k_a(E - E_o))sign\left(\dot{\theta}\cos(\theta)\right) - k_c sign(x)\left[\ln^2\left(1 - \frac{|x|}{0.5L_{Track}}\right)\right]$$

Where:

Eo = Desired Energy Level

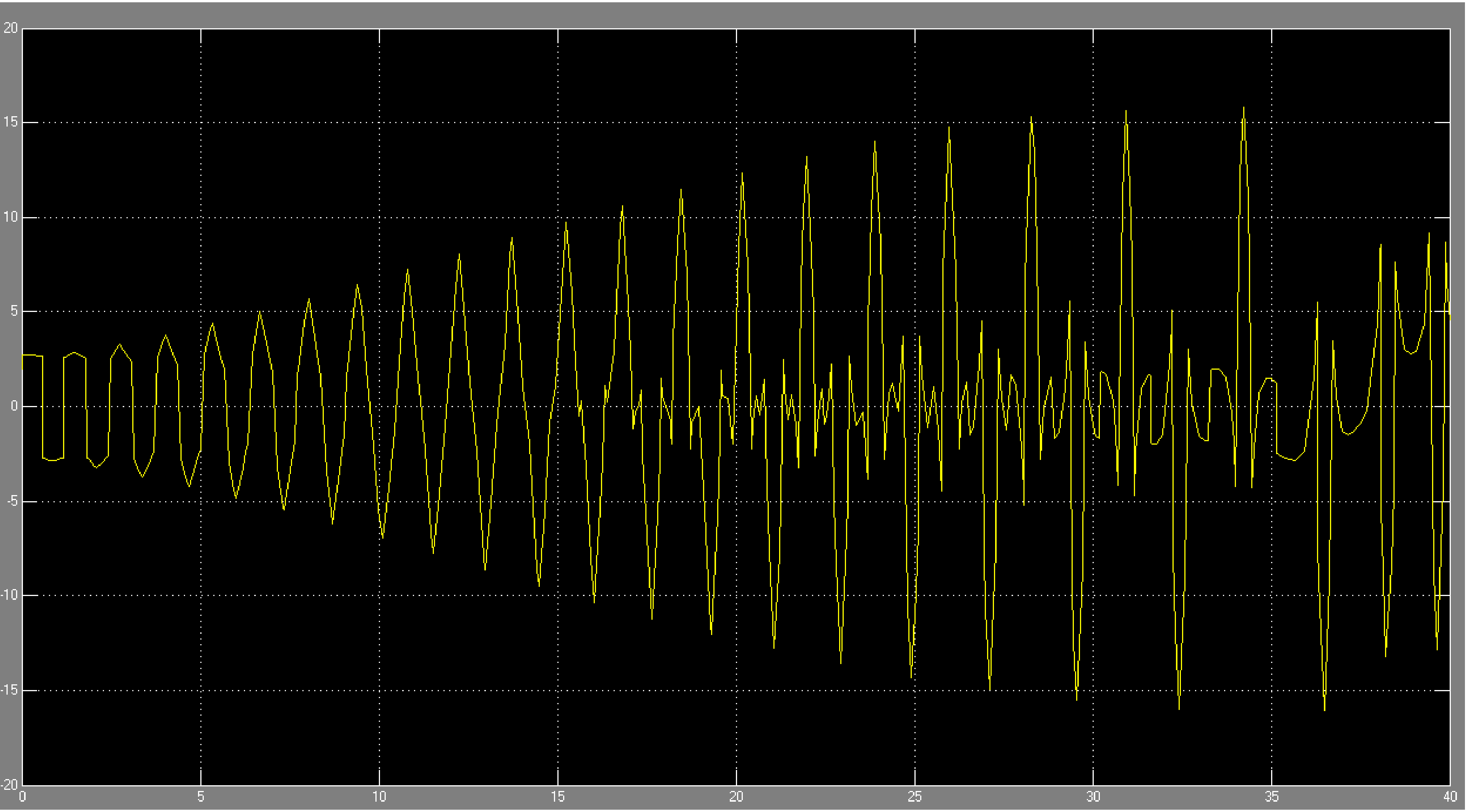
Ka = Pumping Control Gain

Kc = Self-centering/spring gain

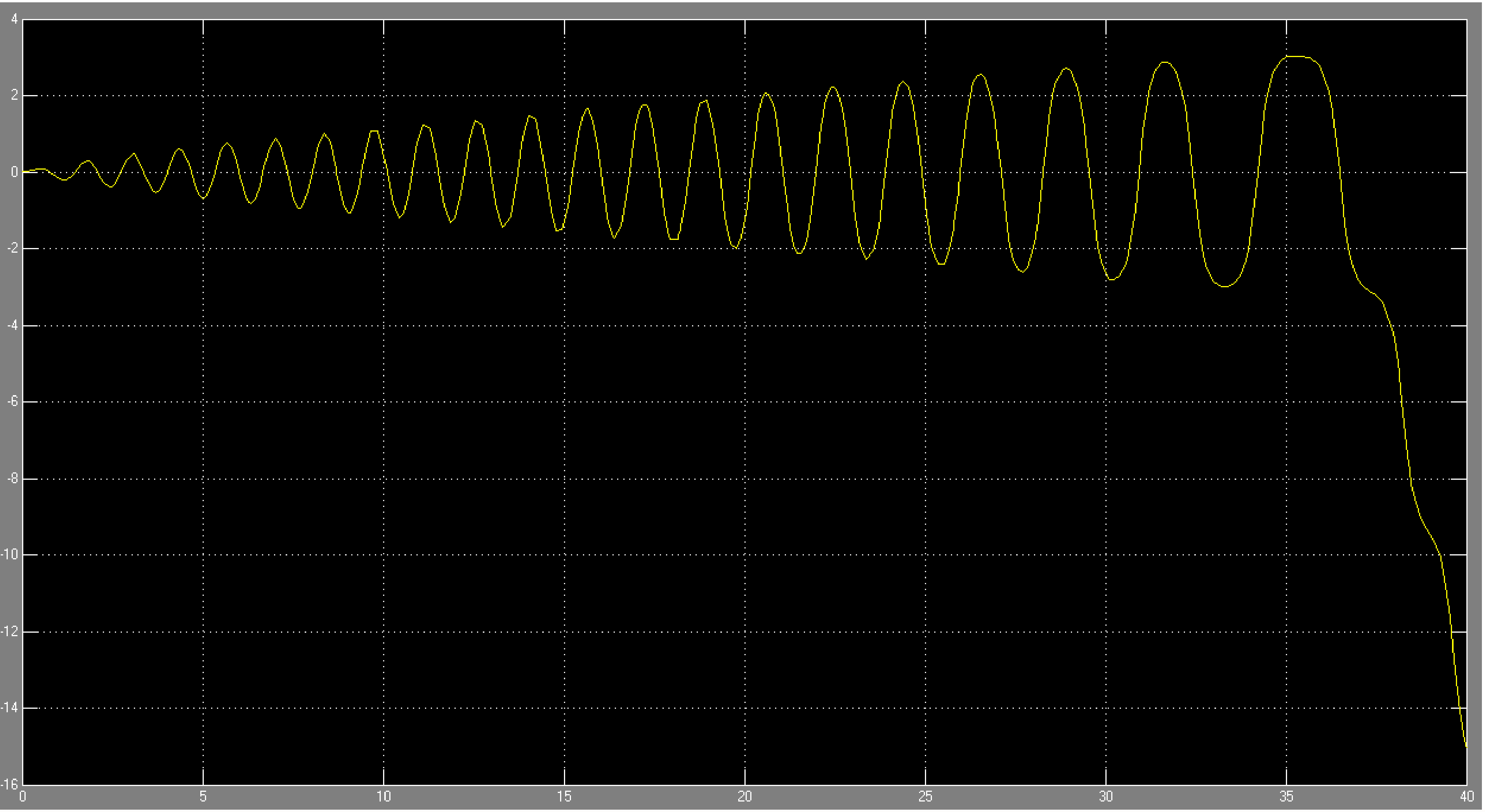


Sources: Lam, J., Chatterjee, D.

Swing-up Controller: Output

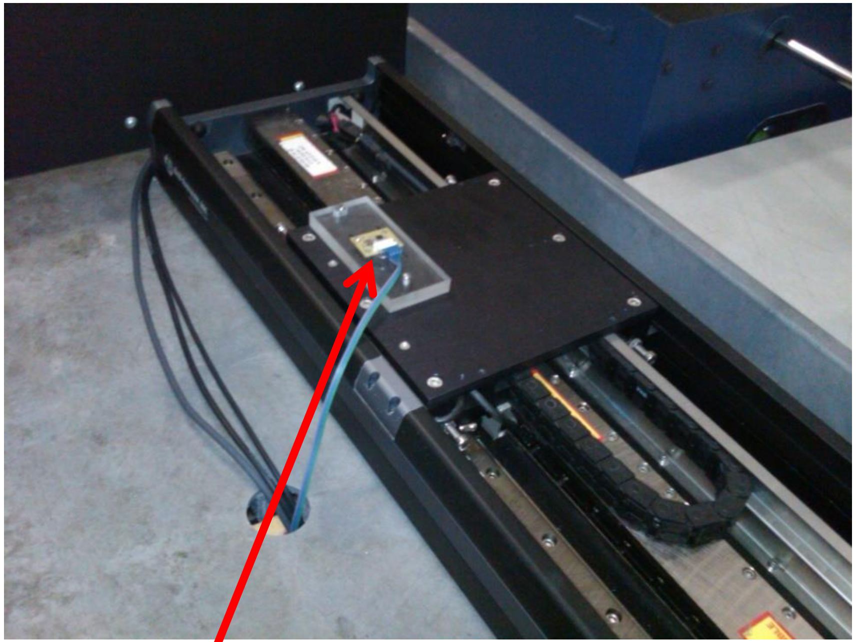


Swing-up Controller: Theta

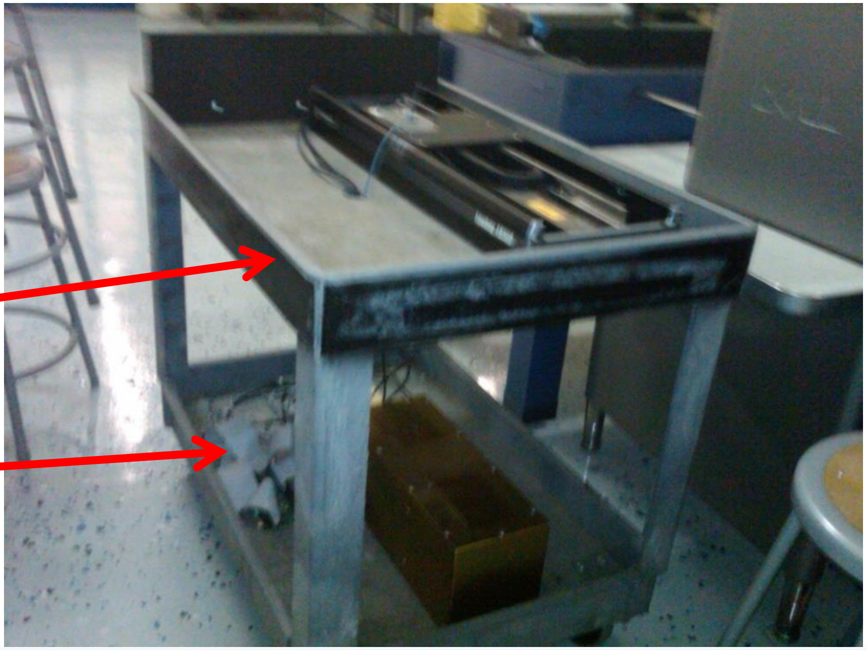


Other Possibilities with the NI sbRIO-9642

Portability with the NI sbRIO-9642

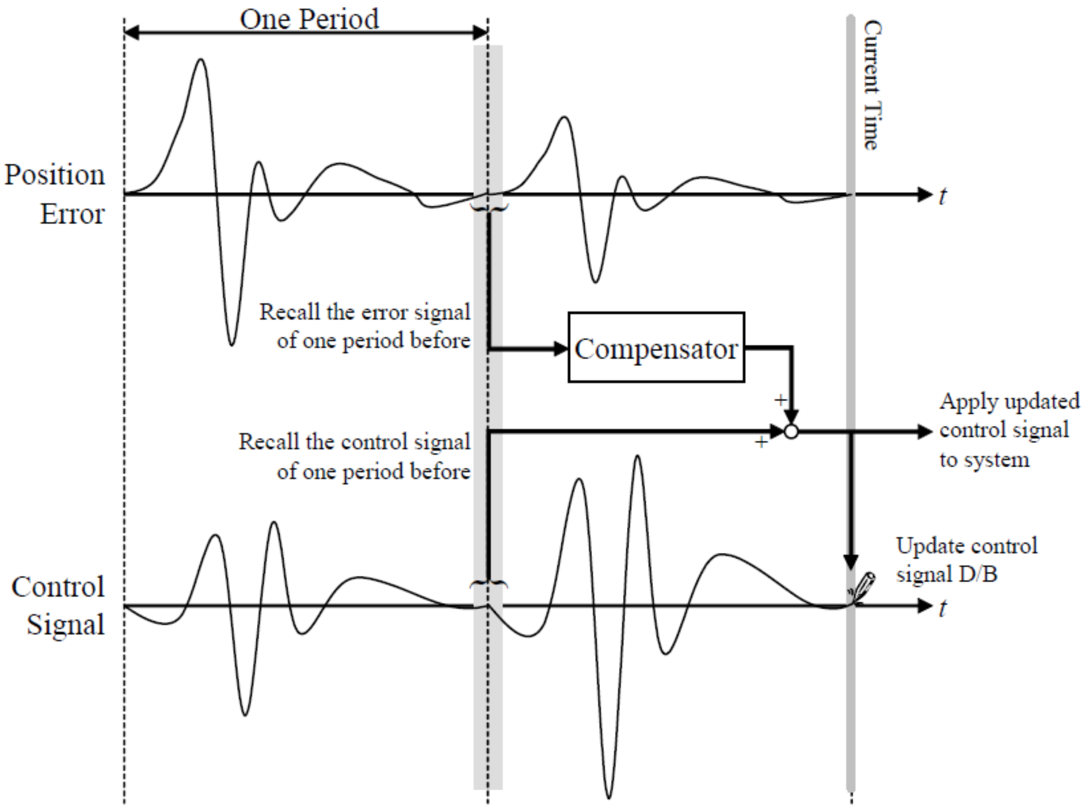


Accelerometer



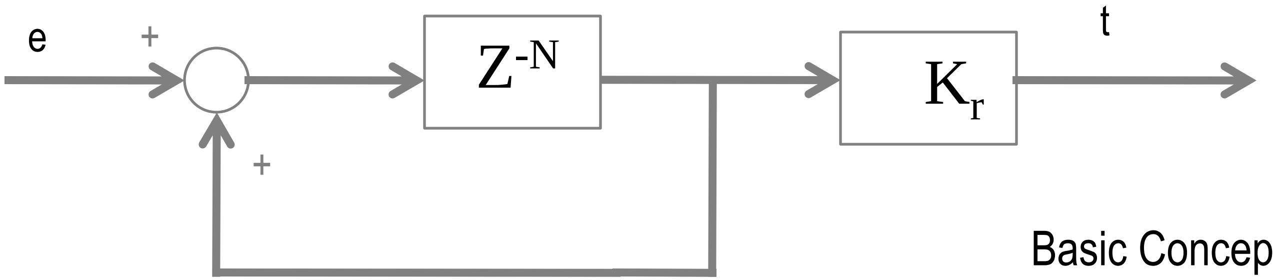
Lowboy on a Cart

NI sbRIO-9642



- Application is repetitive

•Cycle is known



K_r = Repetitive Gain

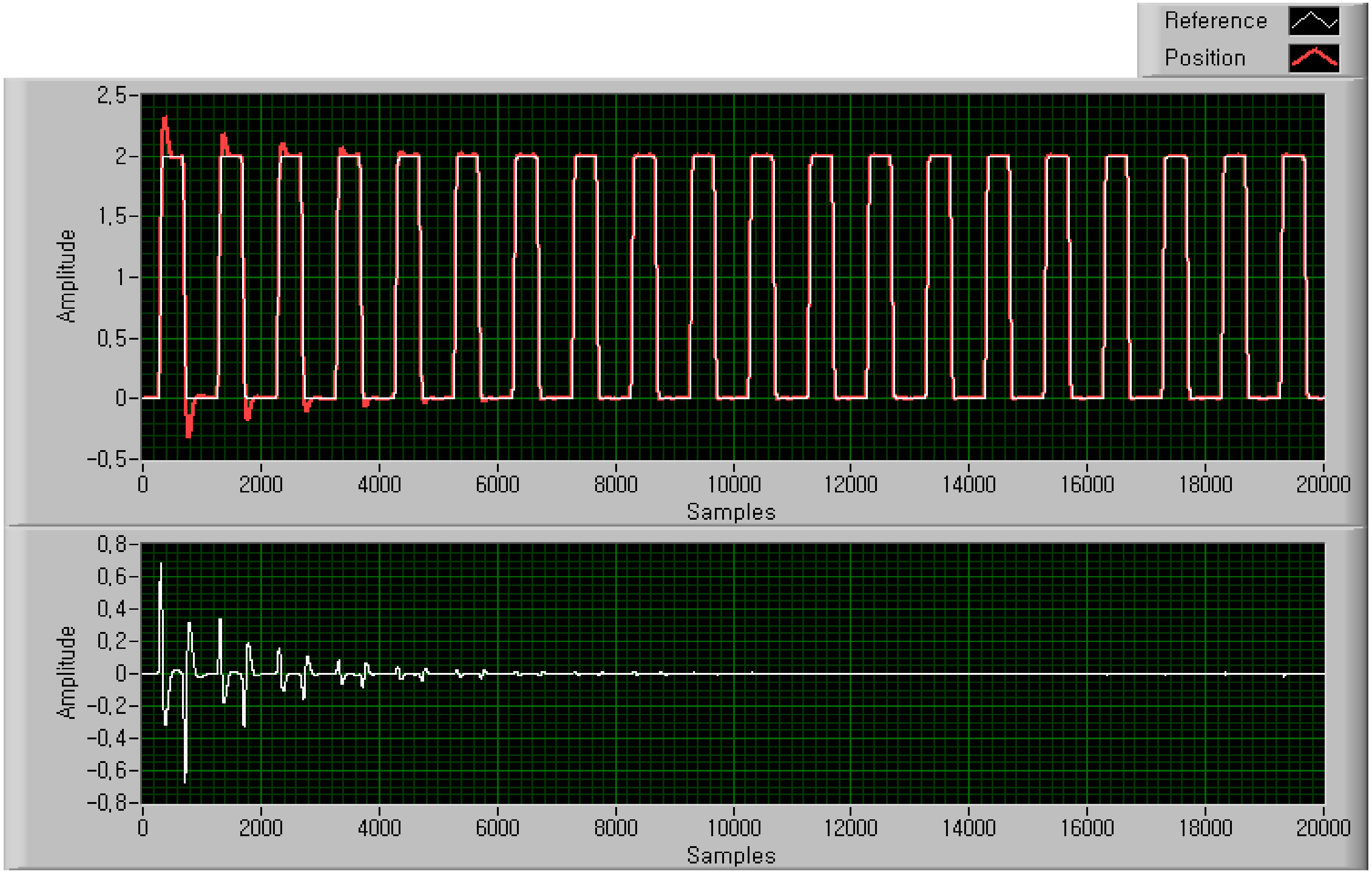
N = Number of sampling rate cycle to complete cycle

Basic Concept

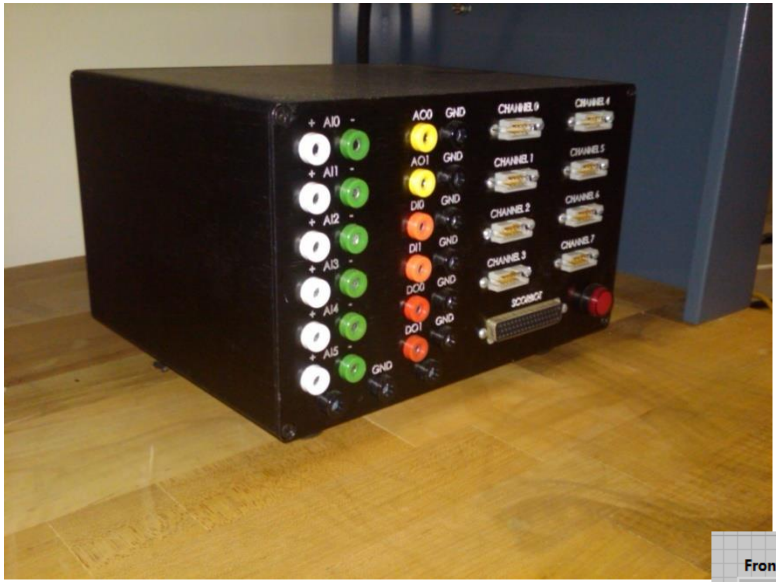
1. Memorize first cycle
2. Generate cyclic error
3. Generate correction
4. Update error database
5. Apply correction

UC Berkeley

Mechanical Engineering Department

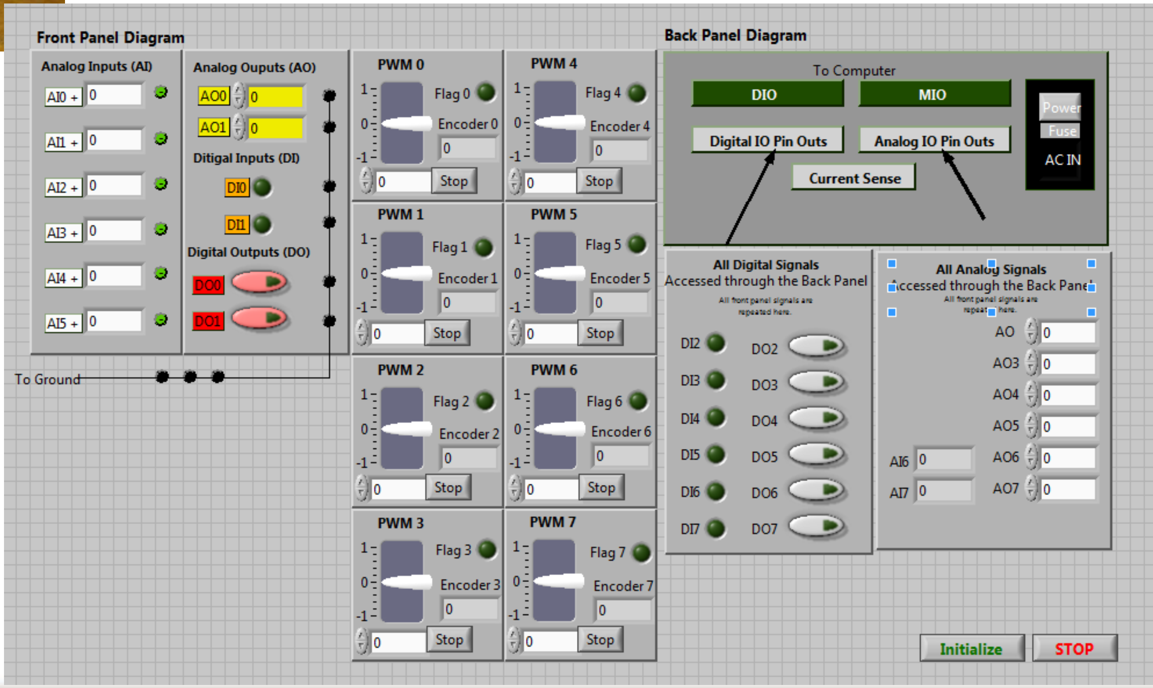


Future of sbRIO system for the laboratory



Integrated sbRIO into the general purpose I/O interface

Student threat interface as a network appliance



UC Berkeley

Mechanical Engineering Department

ME135/ME235
Design of Microprocessor-Based Mechanical Systems

Lecturer:	George Anwar	
Classroom:	105 Northgate Hall TuTh 12:30 – 2:00 PM	
Office:	120 Hesse Hall or 5106 Etcheverry Hall	
Office Hrs:	TuTh 10:00 AM – 12:00 PM, and by appointment	
Lab Space:	120 Hesse Hall	
GSI:	Harshil Goel	harshilgoel@berkeley.edu
	Jianlan Luo	jianlauluo@berkeley.edu
	Daniel Hsieh	daniel_hsieh@berkeley.edu

Overview

- Introduction to Real-time Programming
- Task and State design methodology
- Introduction to LabVIEW 2015
- Real-time implementation issues
- Feedback control basics
- Operator interface

Course Objectives

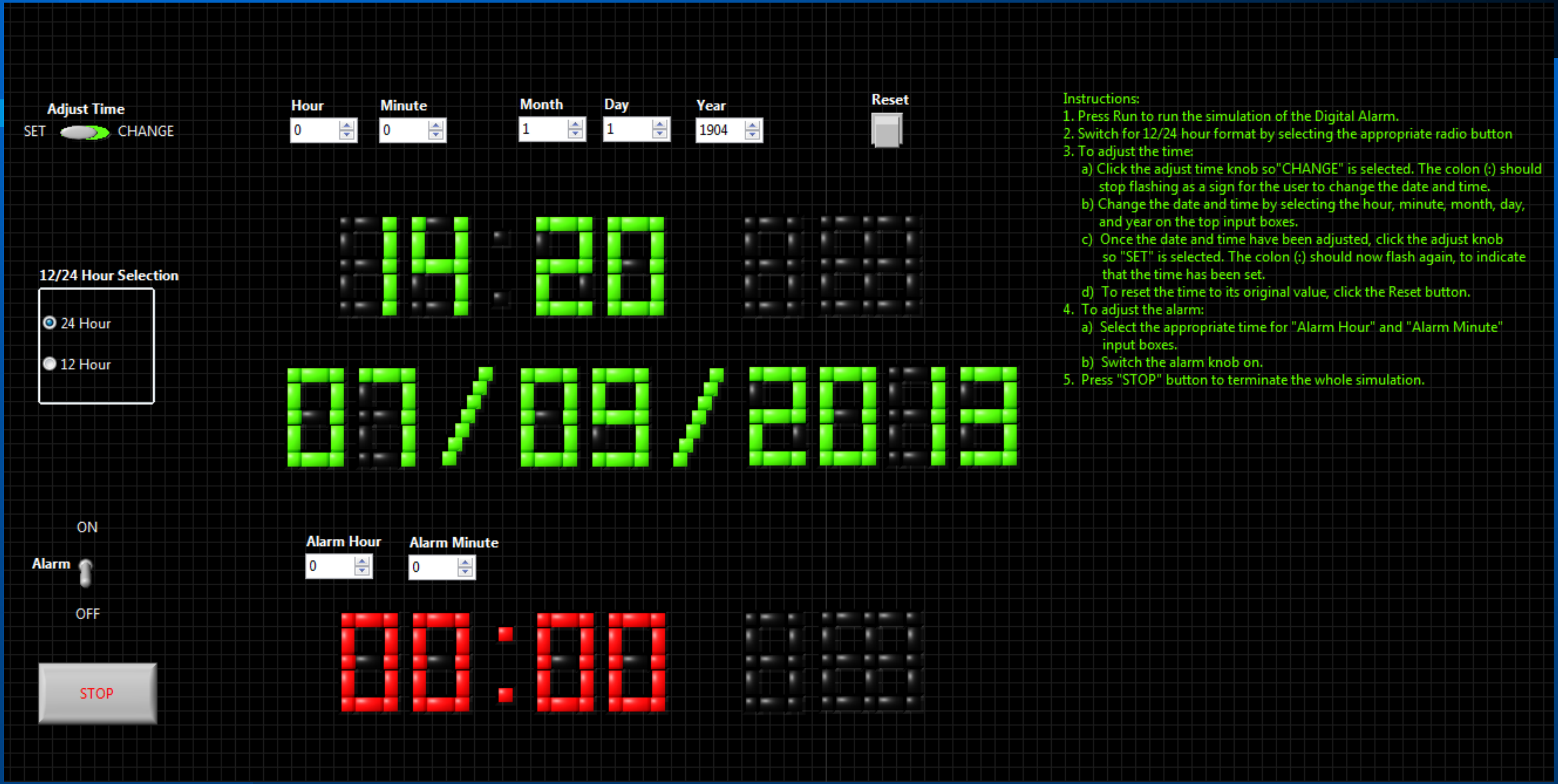
- Assess the relative difficulty of a problem
- Outline a solution to it
- Estimate the resources to solve the problem
- Develop and document a design
- Implement a prototype solution
- Test and evaluate the solution
- Work as part of a team
- Time management

Basis for Grading

- **5-6 Lab Exercises**
- **Final project proposal**
- **Weekly Progress reports**
- **Midterm milestone presentation**
- **CLAD exam (passing will help with grading)**
- **Final project presentation (RRR week)**

Main Software Emphasis

LabVIEW	PC	GUI	1 ms
LabVIEW real-time	Dual Core Cortex-A9	Embedded	1 µs
LabVIEW FPGA	FPGA	I/O	25 ns



Hour: 0
Minute: 0
Month: 1
Day: 1
Year: 1904

14:20

07/09/2013

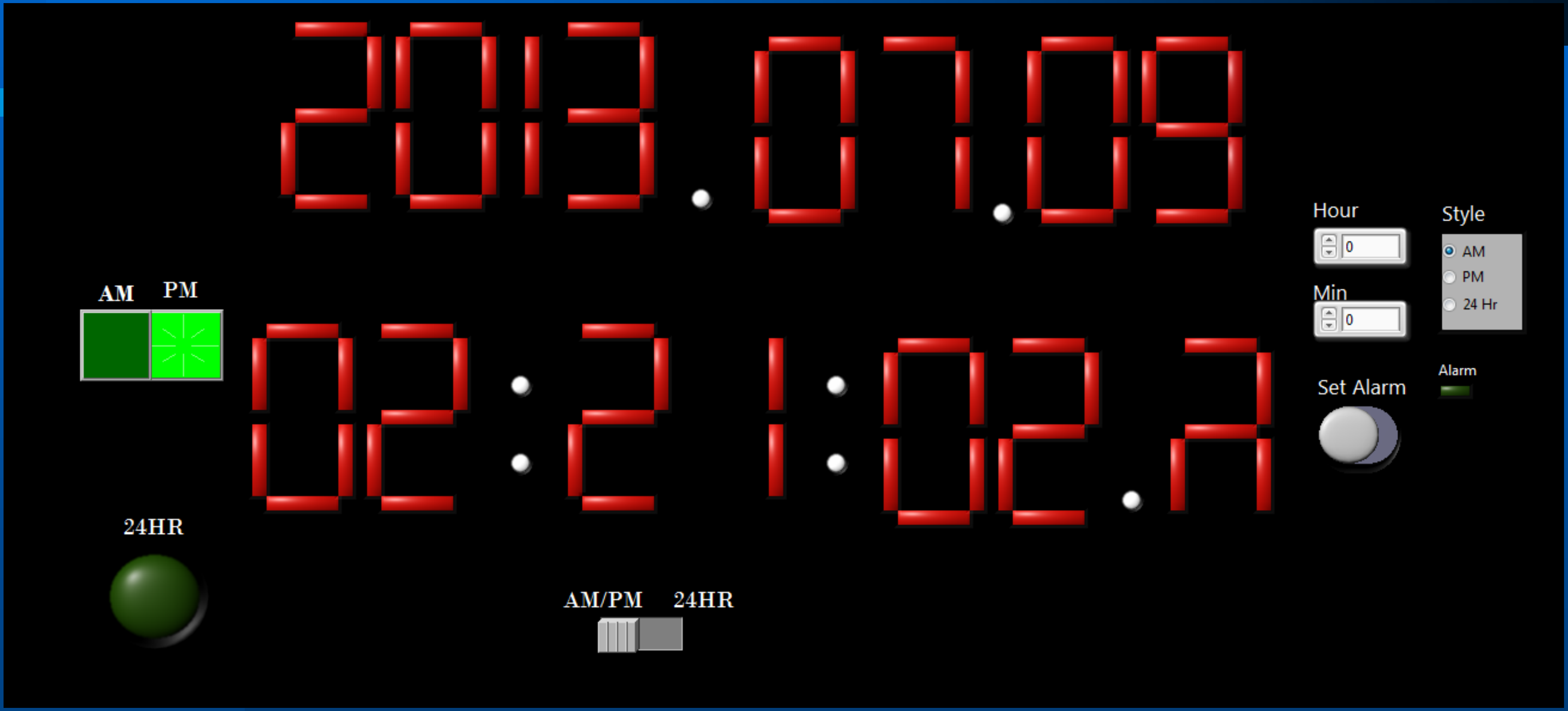
● 24 Hour
○ 12 Hour

Alarm Hour: 0
Alarm Minute: 0

00:00

Instructions:

1. Press Run to run the simulation of the Digital Alarm.
2. Switch for 12/24 hour format by selecting the appropriate radio button
3. To adjust the time:
   a) Click the adjust time knob so"CHANGE" is selected. The colon (:) should stop flashing as a sign for the user to change the date and time.
   b) Change the date and time by selecting the hour, minute, month, day, and year on the top input boxes.
   c) Once the date and time have been adjusted, click the adjust knob so "SET" is selected. The colon (:) should now flash again, to indicate that the time has been set.
   d) To reset the time to its original value, click the Reset button.
4. To adjust the alarm:
   a) Select the appropriate time for "Alarm Hour" and "Alarm Minute" input boxes.
   b) Switch the alarm knob on.
5. Press "STOP" button to terminate the whole simulation.



2013.07.09

02:21:02.7

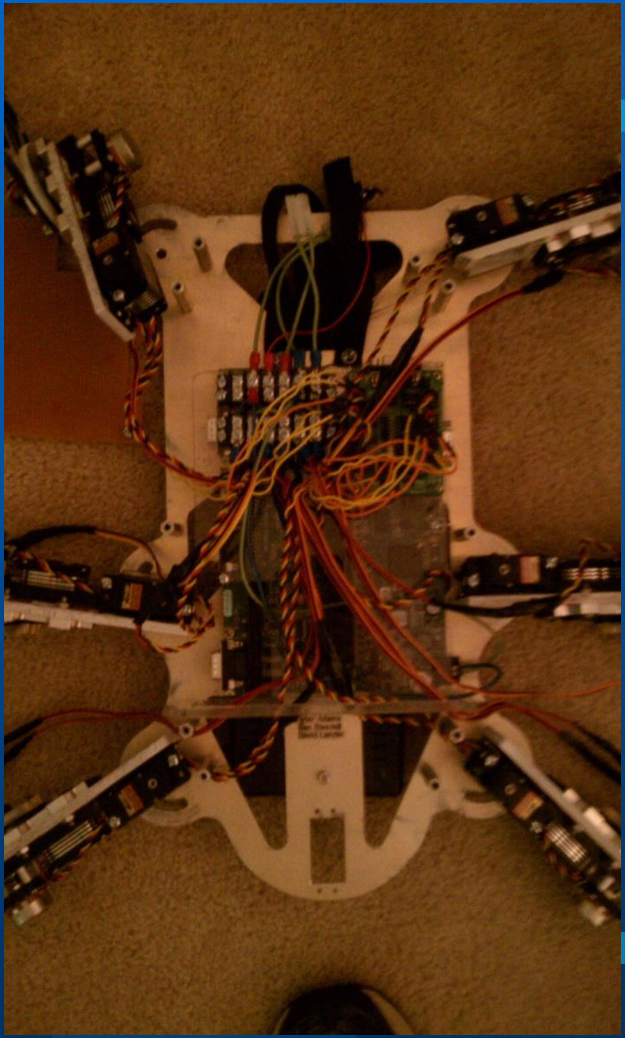
Hour 0

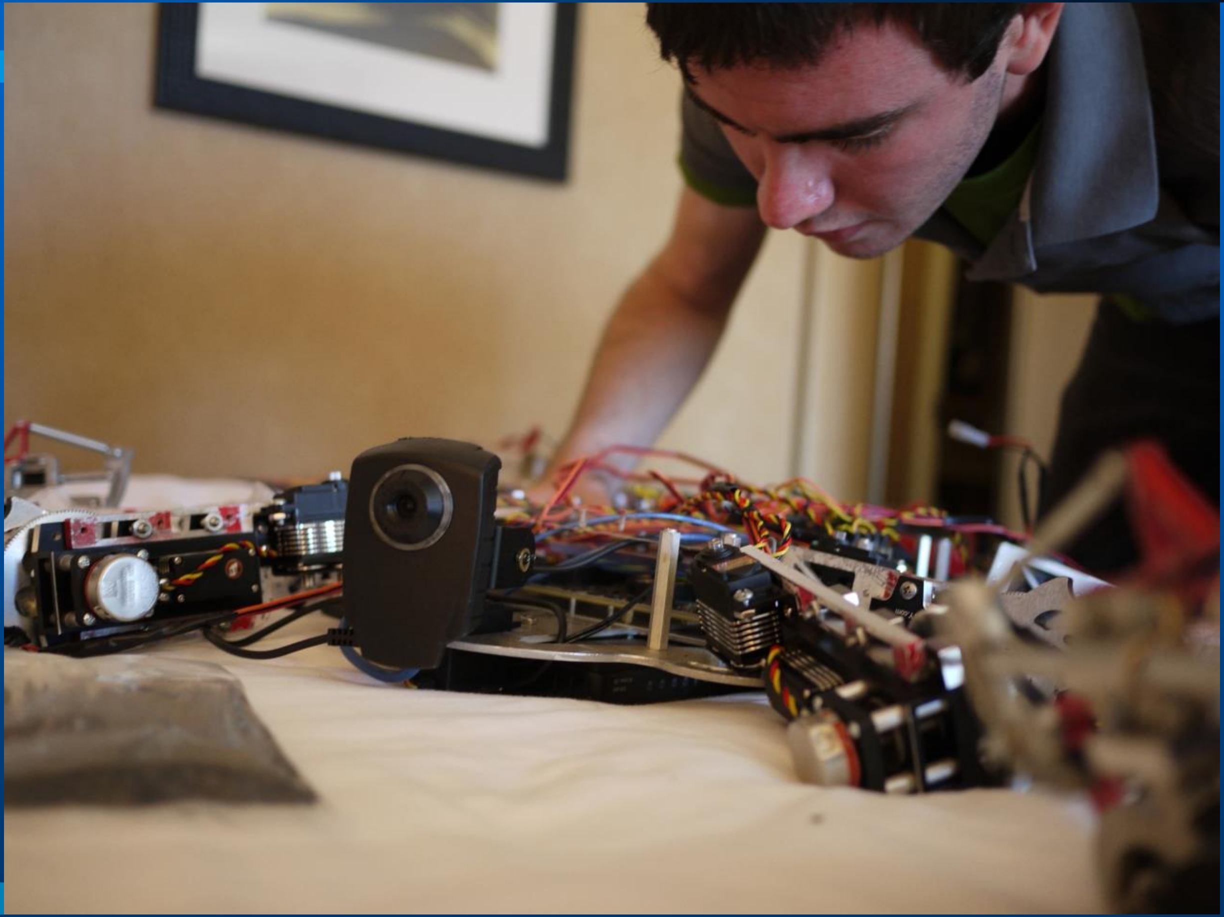
Min 0

Style
● AM
○ PM
○ 24 Hr

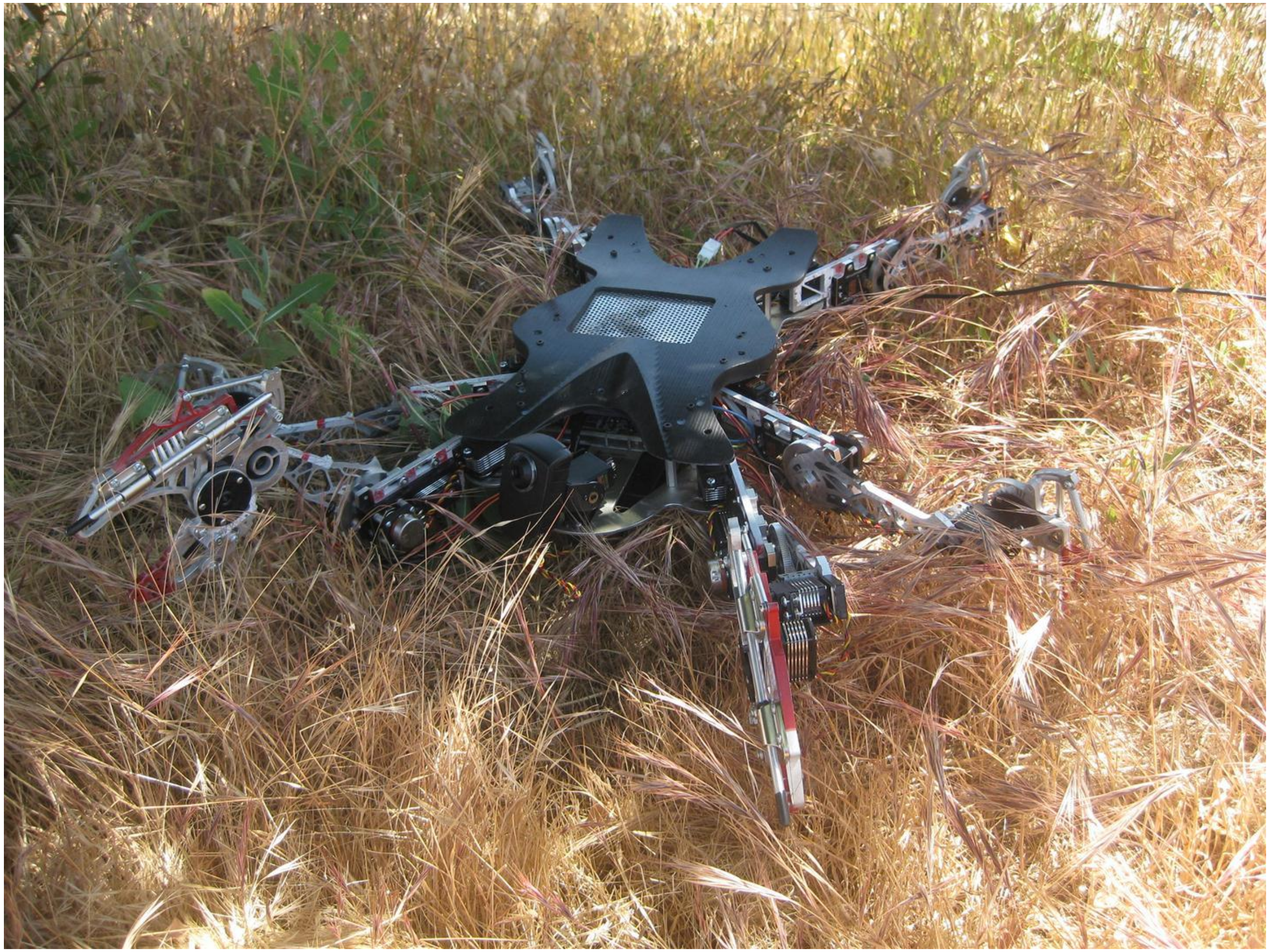
Final Project

- Group Effort (3-4 members optimal)
- Demonstrate the use of real time software
- Design and development of Host GUI software
- Components running on multiple CPU’s or Cores
- Interaction with the external world through sensors, actuators, or other computing units
- Must be multitasking and real time





The HexaQuad Project

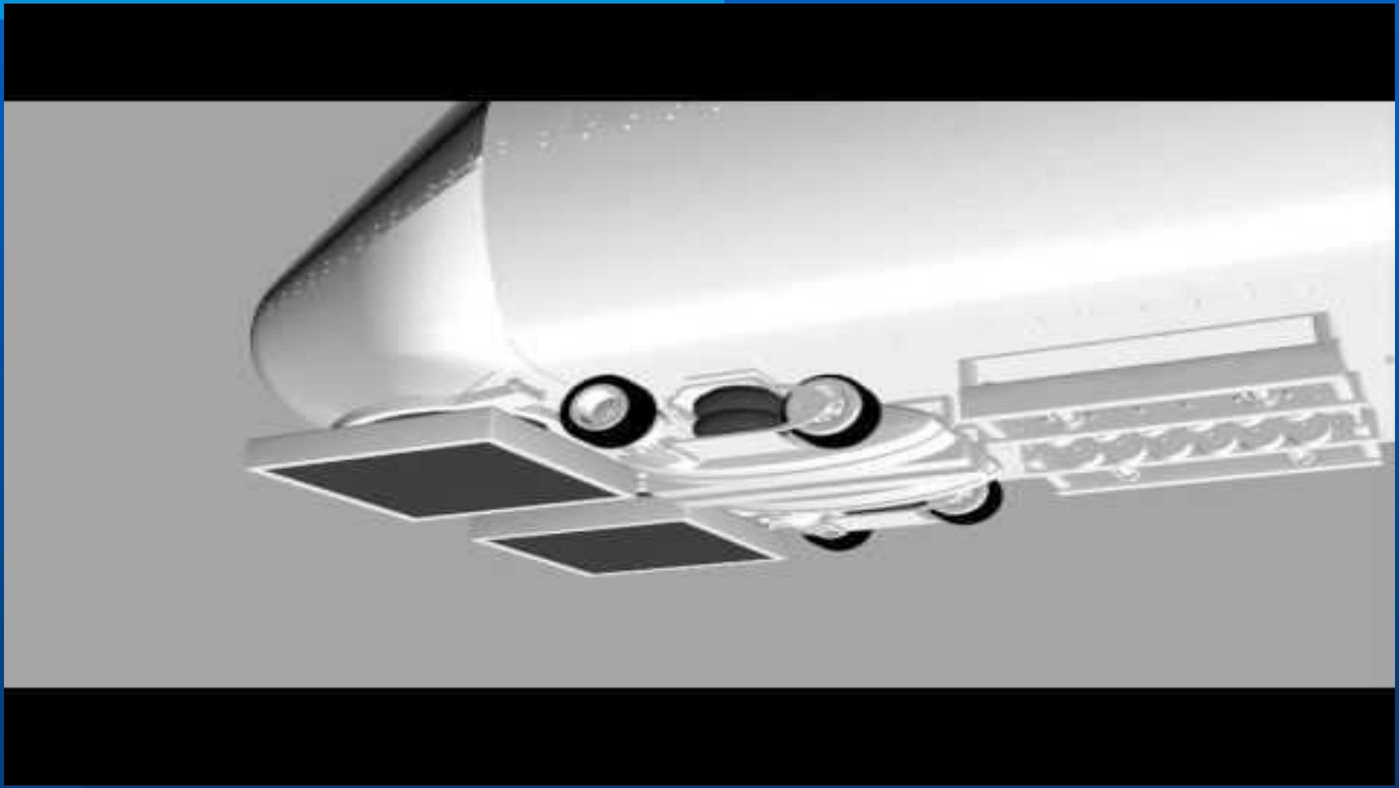


Special Thanks to:

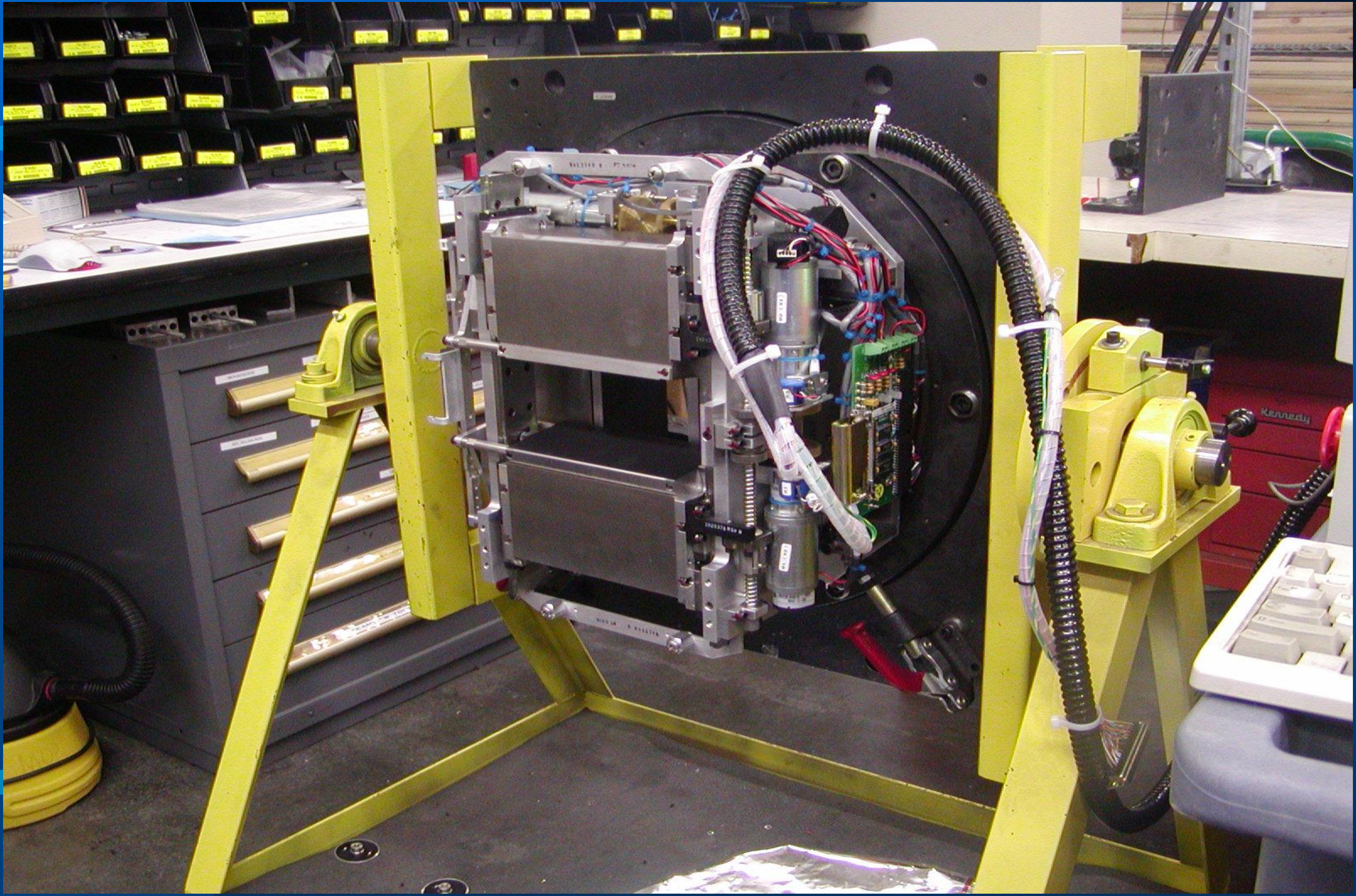
National Instruments
Zach Nelson
Dr. Jeannie Falcon

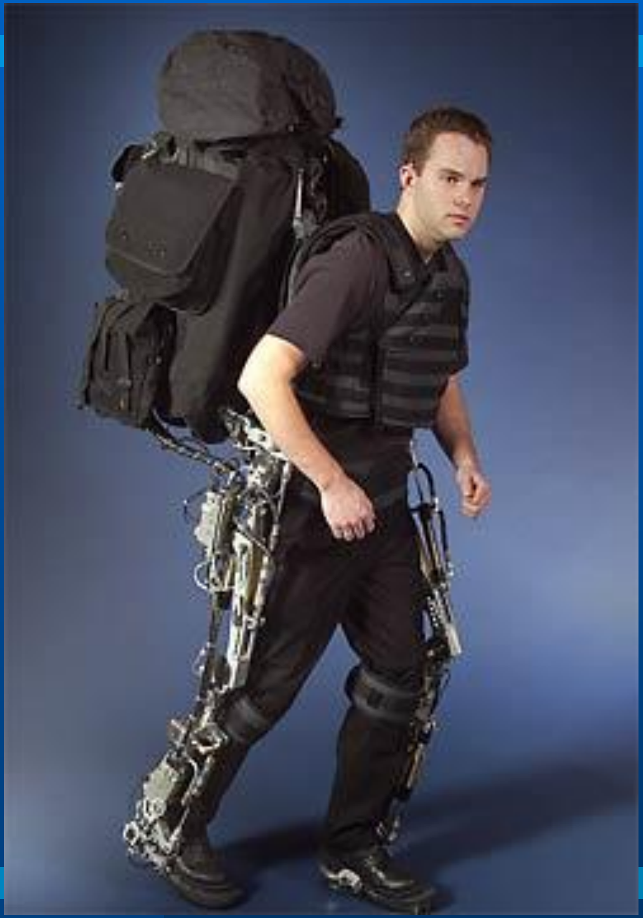
Questions?

Autonomous Guided Vehicles or Drones



CIMLINE
CIMLINE





Berkeley
Lower
Extremity
EXoskeleton

CONTACT INFORMATION:

George Anwar, PhD
Lecturer
Mechanical Engineering Department
5136 Etcheverry Hall
UC Berkeley
Berkeley, CA 94720

ganwar@integratedmotions.com

(510) 205-4839

Thank you & Go Bears!

CLAD Exam:

- **Certified LabVIEW Associate Developer**
- **Exam Format: Multiple choice**
  Exam Duration: One-hour duration

Starting this week

- Form your group
- Each group member
  - Come up with 3 ideas
  - At your first group meeting, select 1.
- Project Presentations: 2/2 and 2/4
  - 2-3 minutes (3-4 slides)

Final Project

- Group Effort (3-4 members optimal, will allow up to 6)
- Demonstrate the use of real time software
- Design and development of Host GUI software
- Components running on multiple CPU’s or Cores
- Interaction with the external world through sensors, actuators, or other computing units
- Must be multitasking and real time

Multitasking:

- Human Multitasking – The ability for someone to perform more than one task at a time

- Computer Multitasking – The apparent simultaneous performance of one or more task by a CPU

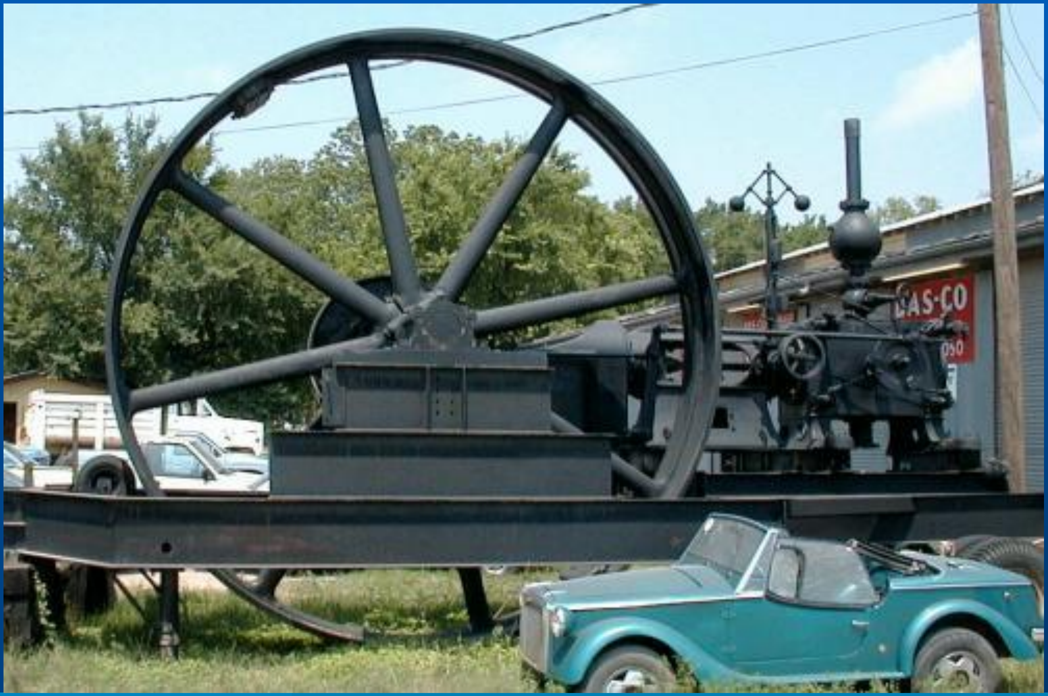
Real-time programming:

In computer science, **real-time computing** (**RTC**), or **reactive computing** describes hardware and software systems subject to a "real-time constraint", for example operational deadlines from event to system response. Real-time programs must guarantee response within specified time constraints, often referred to as "deadlines".[1] Real-time responses are often understood to be in the order of milliseconds, and sometimes microseconds. A system not specified as operating in real time cannot usually guarantee a response within any timeframe, although actual or expected response times may be given.

Source: Wikipedia

Controls
Settings
SUSPENSION
Driving
DOORS & LOCKS
LIGHTS
20.0°
20.0°
271
0

WELCOME TO
O.CO
COLISEUM
BART



- For cotton mill
- 1856
- 100 HP, 30 RPM
- Note flyball (Watt) governor
- Smithville, TX

Some Interesting Stats:

Microcontrollers – tiny computer chips running things From microwave ovens (1) to cars (< 10) to jets (< 1000).

High-end cars (>50)

According to Embedded Systems Programming:

Ten times more microcontrollers than microprocessors are sold.

To Obtain LabVIEW 2015 :

http://software.berkeley.edu/labview

What Is An Embedded System ?

- A type of computer system.
- Some of the Most Common Traditional Definitions :
  - Embedded systems are more limited in hardware and/or software functionality then the PC.
  - An embedded system is designed to perform a dedicated function
  - Does not require human intervention to operate

Computer Program :

Simply a collection of instructions for a computer

Computer :

A machine that manipulates data according to a list of instructions

Source : Wikipedia

Components of a Programming language

- Means for expressing and manipulation of data
  - Data types (int, char, double, float)
  - Operators   (+,-,*,/,%)
  - Expressions (a+b, a+b*c)
- Methods of Controlling Program Flow
  - Statements and Blocks: functions and subroutines, VI's
  - If-Else ; Else-If
  - Switch
  - Loops – Whiles and For, Do-while
- Input and Output
- Advanced Features
  - Pointers and Arrays
  - Structures

Sequential Flow Languages:

Almost all traditional text based languages, C, C++ Java…..

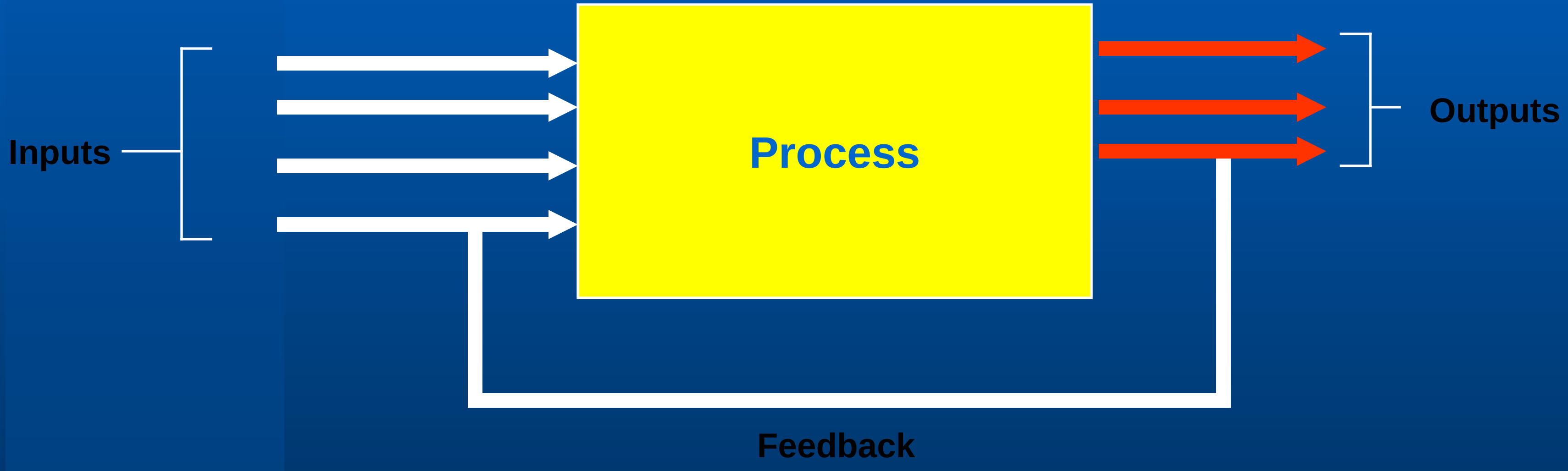
Program flow dictated by the order in which instructions are listed.

Data Flow Languages :

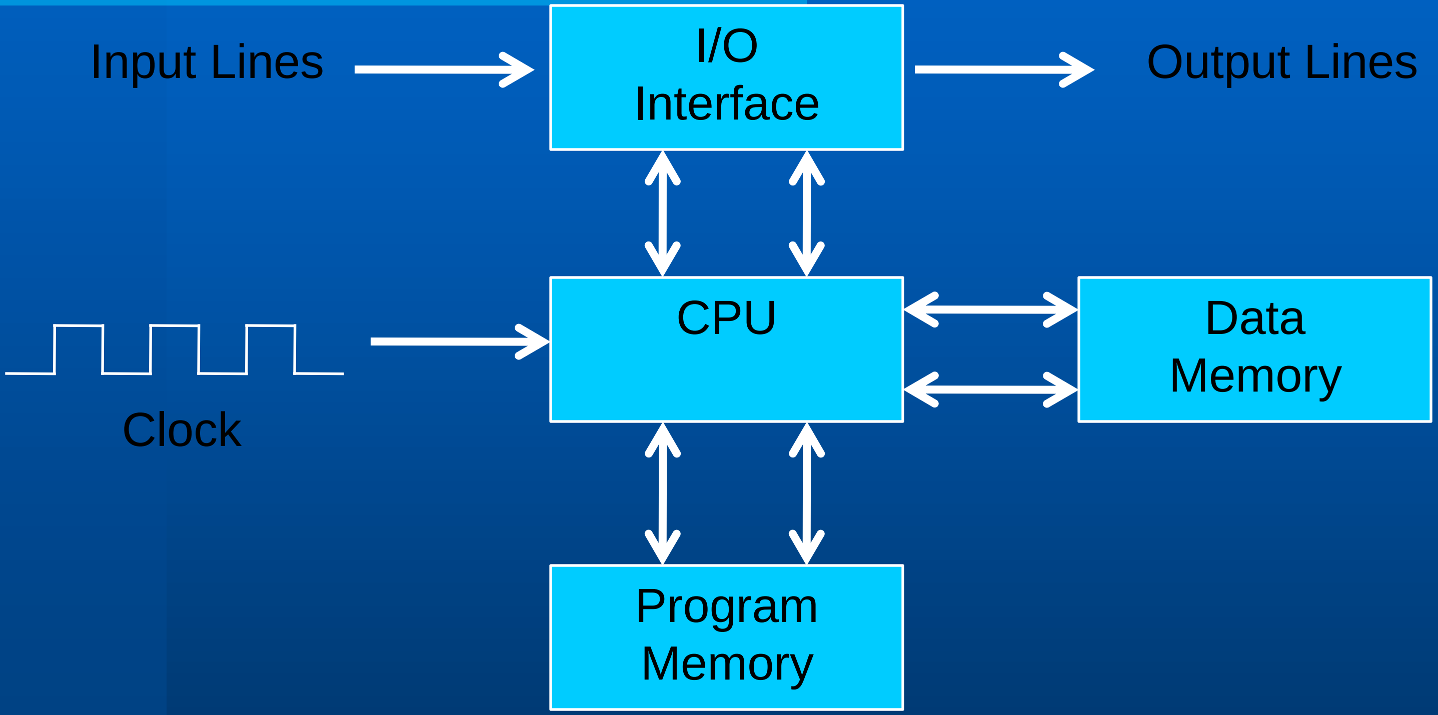
Execution dictated by the readiness of inputs to a set of instructions.

LabVIEW is a data flow language.

Generalized Architecture :



Simplified Model of an Early Processor

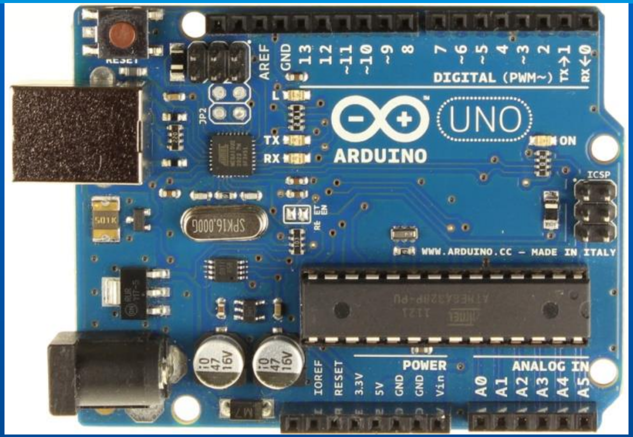


CPU : Central Processing Unit

MyRIO :

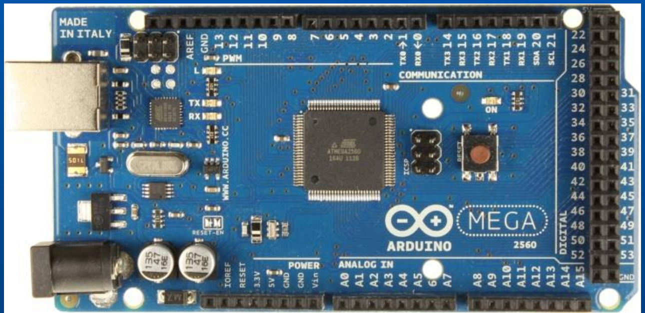
NATIONAL INSTRUMENTS
NI myRIO
POWER
STATUS
LED0
LED1
LED2
LED3

Arduino :



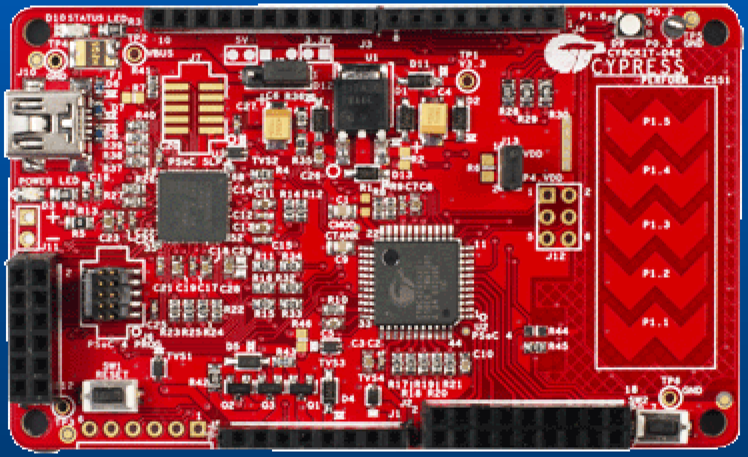
UNO

MEGA



MICRO

Cypress PSOC 4:



PDP-8 -- DEC in 1960

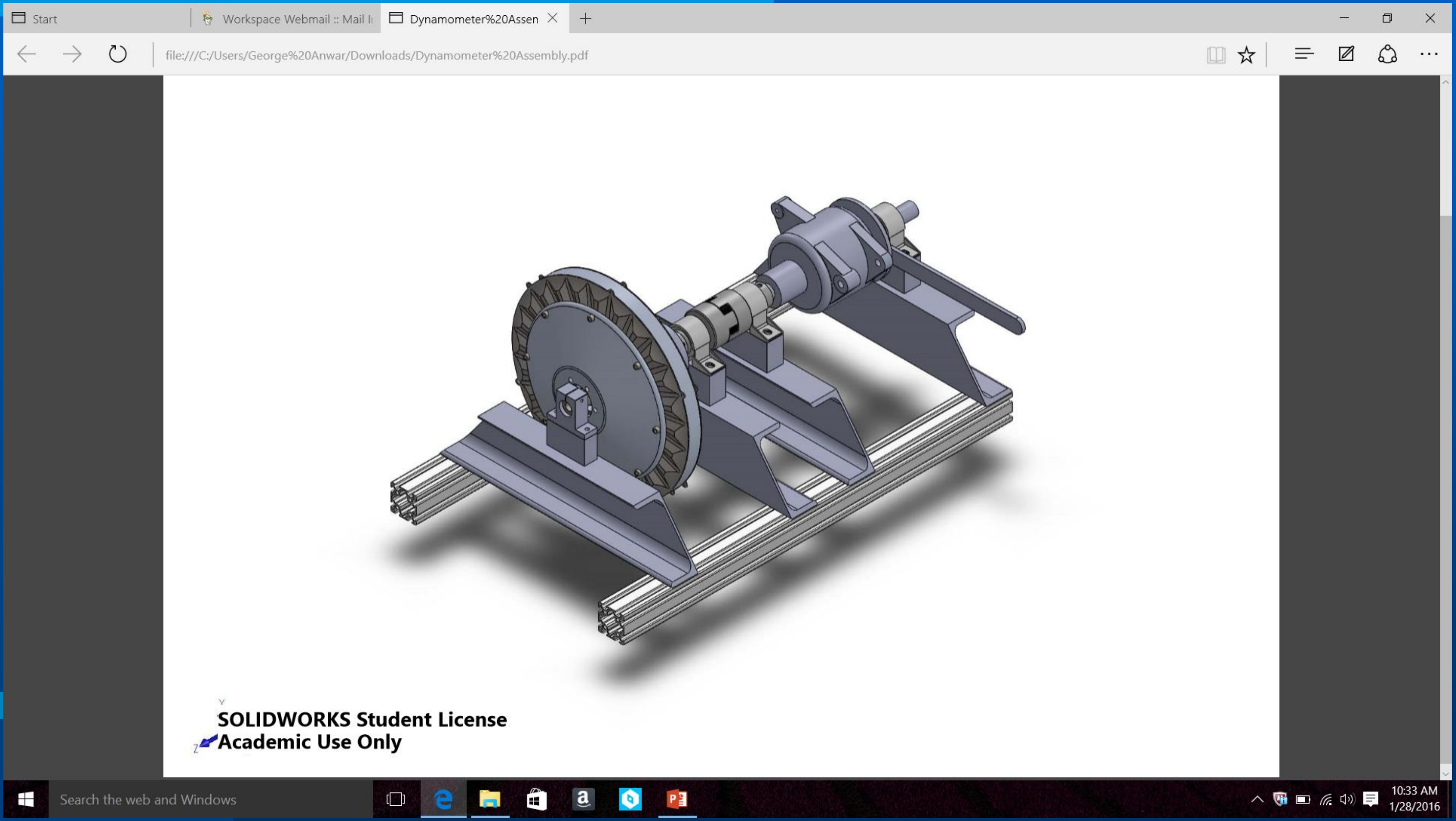
- •12 bit computer
- •Magnetic Core Memory
- •4K Word RAM expandable to 32K
- •Multiply/Divide is an option

Multitasking:

- Human Multitasking – The ability for someone to perform more than one task at a time

- Computer Multitasking – The apparent simultaneous performance of one or more task by a CPU

Dynanometer for CalSol:



Start
Workspace Webmail :: Mail I
Dynamometer%20Assen
file:///C:/Users/George%20Anwar/Downloads/Dynamometer%20Assembly.pdf
Search the web and Windows
10:33 AM
1/28/2016

SOLIDWORKS Student License
Academic Use Only

Real-time Programming:

Real-time Programming:

All aspect of traditional programming with the added complexity of time

Un-Real-time Programming:

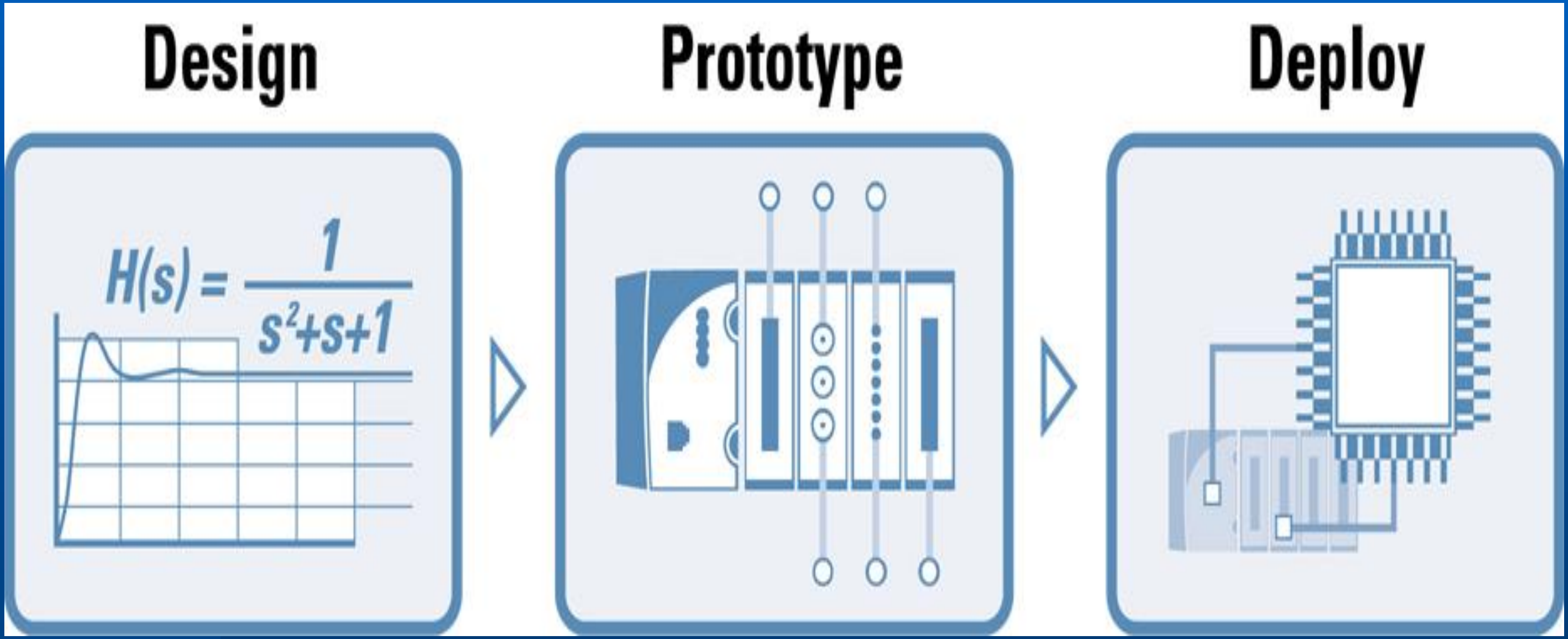
The time assigned to complete projects in this class

Typical Real-time Task:

- Controls
  - Stability and predictable behavior
- Trajectory Generation
  - Following a profile
- Data Acquisition
  - Signal reproduction
- Sound Reproduction
- Video Streaming

NEXT STEPS:

- DEFINE YOUR PROJECT
- DOES IT FULFILL THE COURSE REQUIREMENTS
- SOLIDIFY PROJECT SPECIFICATION
- BEWARE OF MOVING TARGET
- DIVIDE AND CONQUER
- DIVIDE BY TASK
- DIVIDE BY DISCIPLINE
- APPOINT LEADS


Design
Prototype
Deploy
$H(s) = \frac{1}{s^2+s+1}$

Last year’s 29 Projects:

• 16	Tracking Problems
• 6	Balancing Problems
• 5	Others
• 2	Flying

Most Common Actuator:

Motors

Sensors:

1. Encoders
2. Vision
3. Accelerometer/Rate Gyros

Project Proposal Presentations:

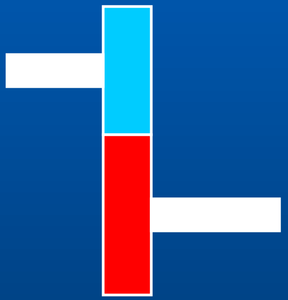
- 3 PPT Slides (limit to 2 minutes)
- Present Group Members
- Final Project Proposal
- Motivation (To get an A in the class is implied)
- Real time component
- Multitasking element
- Challenges

🟨 Title Slide
🟩 Slide 2
🟧 Slide 3

Presentations : Feb 9th and Feb 11th in class
Have slides in by midnight Feb 7th

Mechanical Example :

Input Velocity = Vin



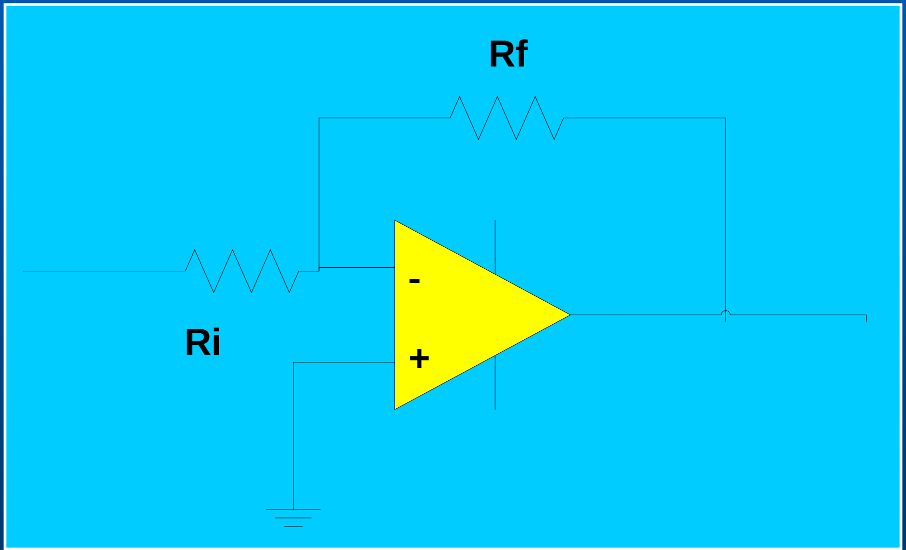
Output Velocity = Vout

Vout = Vin/100

100:1 Gear

Electrical Example :

Input Voltage = Vin



Output Voltage = Vout

Vout = -Rf/Ri *Vin

Operational Amplifier

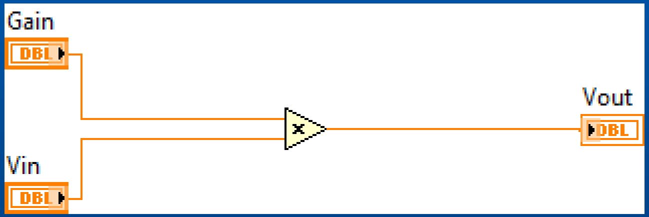
C

```
#include <stdio.h>
#include <math.h>

float gain = 2.0;
float Vout;
float Vin = 15.0;

main()
{
            Vout = gain*Vin;
            printf(“Vout = %f  \n”);
}
```

LabVIEW



Components of a Programming language

- Means for expressing and manipulation of data
  - Data types (int, char, double, float)
  - Operators (+,-,*,/,%)
  - Expressions (a+b, a+b*c)
- Methods of Controlling Program Flow
  - Statements and Blocks: functions and subroutines, VI's
  - If-Else ; Else-If
  - Switch
  - Loops – Whiles and For, Do-while
- Input and Output
- Advanced Features
  - Pointers and Arrays
  - Structures

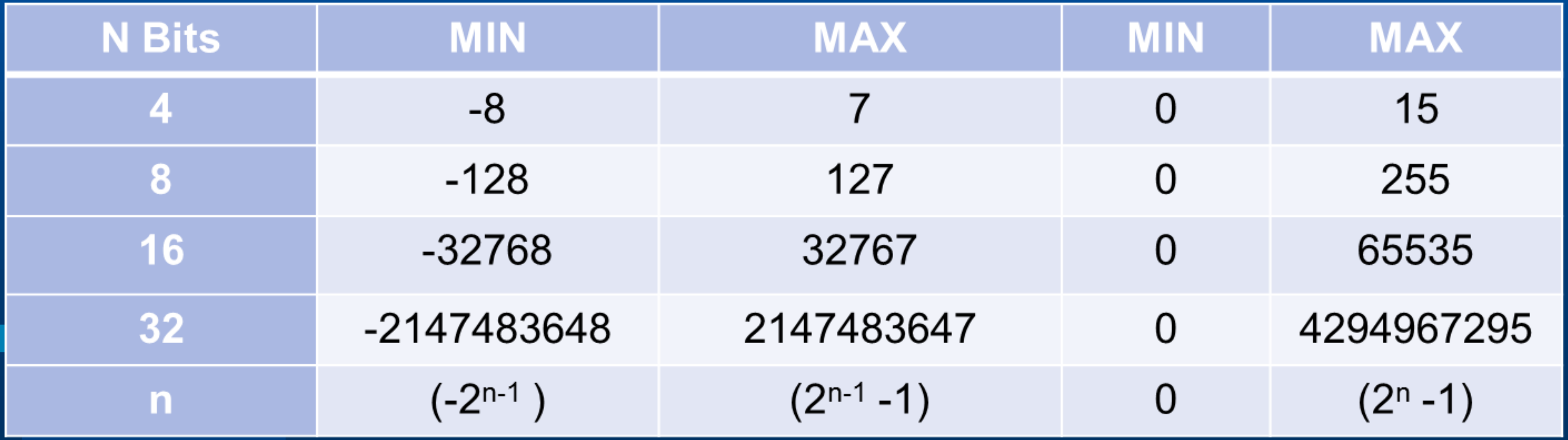
Data types (int, char, double, single……)

- Allocate space for storage
  - Number of bytes
  - How to interprete the data
- Integers
  - Represent only whole numbers
  - Size typically 8 bits, 16 bits, 32 bits……
  - Signed, unsigned
- Floating point
  - Represent fractional numbers
  - Extended, Double, Single   Precision
  - Computationally expensive
- String
  - Represent Alpanumeric characters
  - Each character is a byte
  - String is an array of bytes

Integer

- Signed
  - I8, I16, I32, I64 ---- represent both positive and negative numbers
- Unsigned
  - U8, U16, U32, U64 --- represent only positive numbers
  - Signed, unsigned

SIGNED **UNSIGNED**



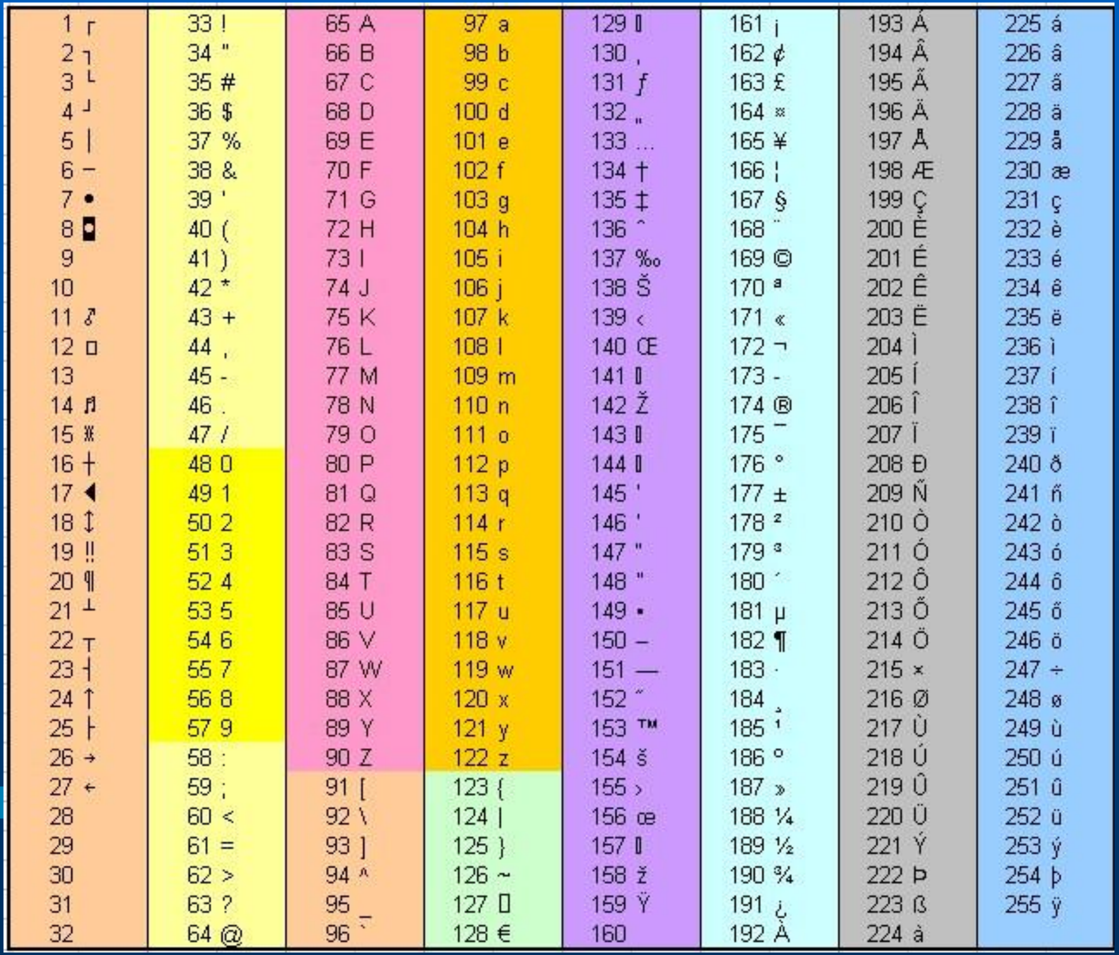
N Bits	MIN	MAX	MIN	MAX
4	-8	7	0	15
8	-128	127	0	255
16	-32768	32767	0	65535
32	-2147483648	2147483647	0	4294967295
n	(-2^{n-1})	$(2^{n-1} -1)$	0	$(2^{n} -1)$

Floating Point

- Extended Precision
  - 10 bytes
  - 15-20 digit precision (dependent on computer)
- Double Precision
  - 8 bytes (64 bits)
  - 15 digits of precision
- Single Precision
  - 4 bytes (32 bits)
  - 6 digits of precision

String

- **Array of Characters**
  - **Each character is a byte**
  - **Represents ASCII table**



1 ┌	33 !	65 A	97 a	129 ▯	161 ¡	193 Á	225 á
2 ┐	34 "	66 B	98 b	130 ‚	162 ¢	194 Â	226 â
3 └	35 #	67 C	99 c	131 ƒ	163 £	195 Ã	227 ã
4 ┘	36 $	68 D	100 d	132 „	164 ¤	196 Ä	228 ä
5 │	37 %	69 E	101 e	133 …	165 ¥	197 Å	229 å
6 ─	38 &	70 F	102 f	134 †	166 ¦	198 Æ	230 æ
7 •	39 '	71 G	103 g	135 ‡	167 §	199 Ç	231 ç
8 ◘	40 (	72 H	104 h	136 ˆ	168 ¨	200 È	232 è
9	41)	73 I	105 i	137 ‰	169 ©	201 É	233 é
10	42 *	74 J	106 j	138 Š	170 ª	202 Ê	234 ê
11 ♂	43 +	75 K	107 k	139 ‹	171 «	203 Ë	235 ë
12 □	44 ,	76 L	108 l	140 Œ	172 ¬	204 Ì	236 ì
13	45 -	77 M	109 m	141 ▯	173 -	205 Í	237 í
14 ♫	46 .	78 N	110 n	142 Ž	174 ®	206 Î	238 î
15 ☼	47 /	79 O	111 o	143 ▯	175 ¯	207 Ï	239 ï
16 ┼	48 0	80 P	112 p	144 ▯	176 °	208 Ð	240 ð
17 ◄	49 1	81 Q	113 q	145 '	177 ±	209 Ñ	241 ñ
18 ↕	50 2	82 R	114 r	146 '	178 ²	210 Ò	242 ò
19 ‼	51 3	83 S	115 s	147 "	179 ³	211 Ó	243 ó
20 ¶	52 4	84 T	116 t	148 "	180 ´	212 Ô	244 ô
21 ┴	53 5	85 U	117 u	149 •	181 µ	213 Õ	245 õ
22 ┬	54 6	86 V	118 v	150 –	182 ¶	214 Ö	246 ö
23 ┤	55 7	87 W	119 w	151 —	183 ·	215 ×	247 ÷
24 ↑	56 8	88 X	120 x	152 ˜	184 ¸	216 Ø	248 ø
25 ├	57 9	89 Y	121 y	153 ™	185 ¹	217 Ù	249 ù
26 →	58 :	90 Z	122 z	154 š	186 º	218 Ú	250 ú
27 ←	59 ;	91 [	123 {	155 ›	187 »	219 Û	251 û
28	60 <	92 \	124 |	156 œ	188 ¼	220 Ü	252 ü
29	61 =	93]	125 }	157 ▯	189 ½	221 Ý	253 ý
30	62 >	94 ^	126 ~	158 ž	190 ¾	222 Þ	254 þ
31	63 ?	95 _	127 ▯	159 Ÿ	191 ¿	223 ß	255 ÿ
32	64 @	96 `	128 €	160	192 À	224 à	

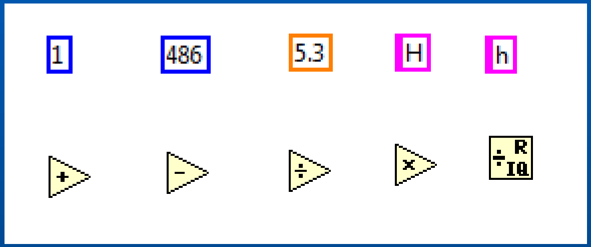
Primitive Expressions :

C :

- Data -- 1 486 5.3 H h

- Instructions -- + - / * %

LabVIEW:



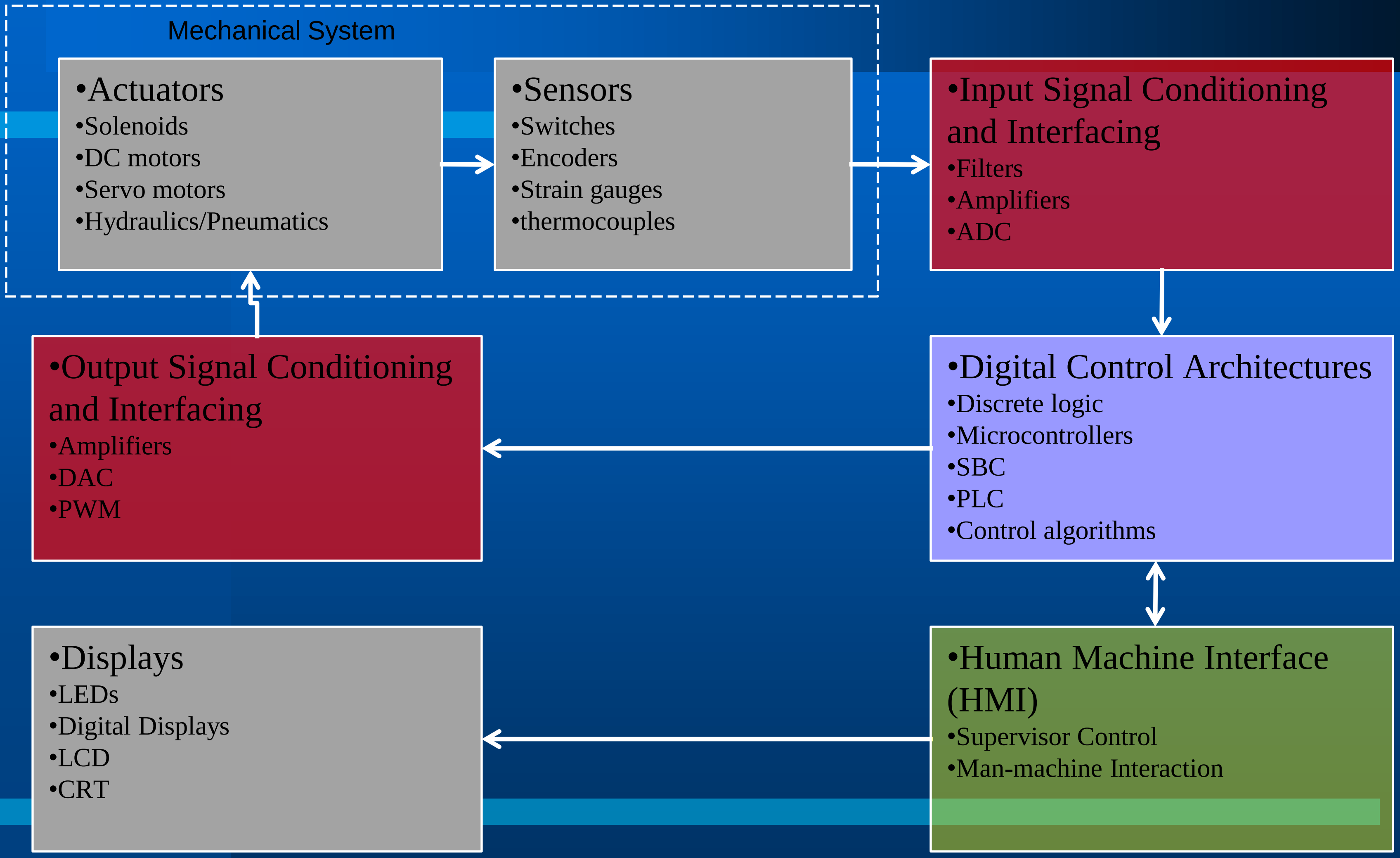
Next Steps:

- Decide Simulate or Build
- Divide Main Task
- Assign Lead
- Allocate time using Gantt Chart Resource Manager
- Work backward from Demo Day 05/04/2016 (10 weeks and a day including Spring Break)
- Complete Gantt Chart
- Revisit Real-time and multitasking
- Start thinking about use model and GUI functionality
- Start inventory of resources

Start thinking about:

- Motors types
  - Steppers
  - Servos
  - DC
  - DC Brushless
- Sensors
  - What Sensors
  - Analog or Digital
  - Resolution
  - Bandwidth

Mechatronic system components:



Mechanical System

- Actuators
  - Solenoids
  - DC motors
  - Servo motors
  - Hydraulics/Pneumatics

- Sensors
  - Switches
  - Encoders
  - Strain gauges
  - thermocouples

- Input Signal Conditioning and Interfacing
  - Filters
  - Amplifiers
  - ADC

- Digital Control Architectures
  - Discrete logic
  - Microcontrollers
  - SBC
  - PLC
  - Control algorithms

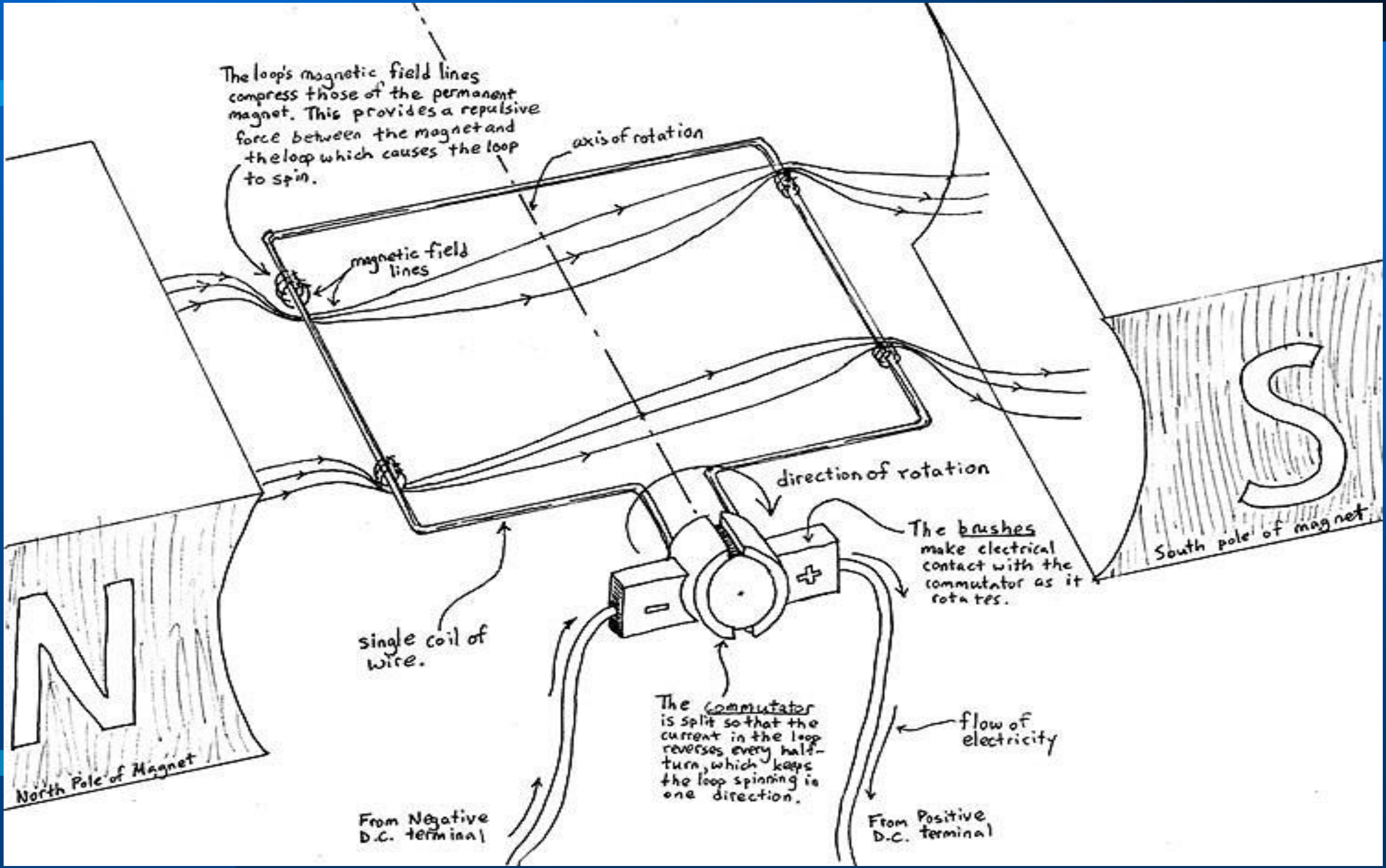
- Output Signal Conditioning and Interfacing
  - Amplifiers
  - DAC
  - PWM

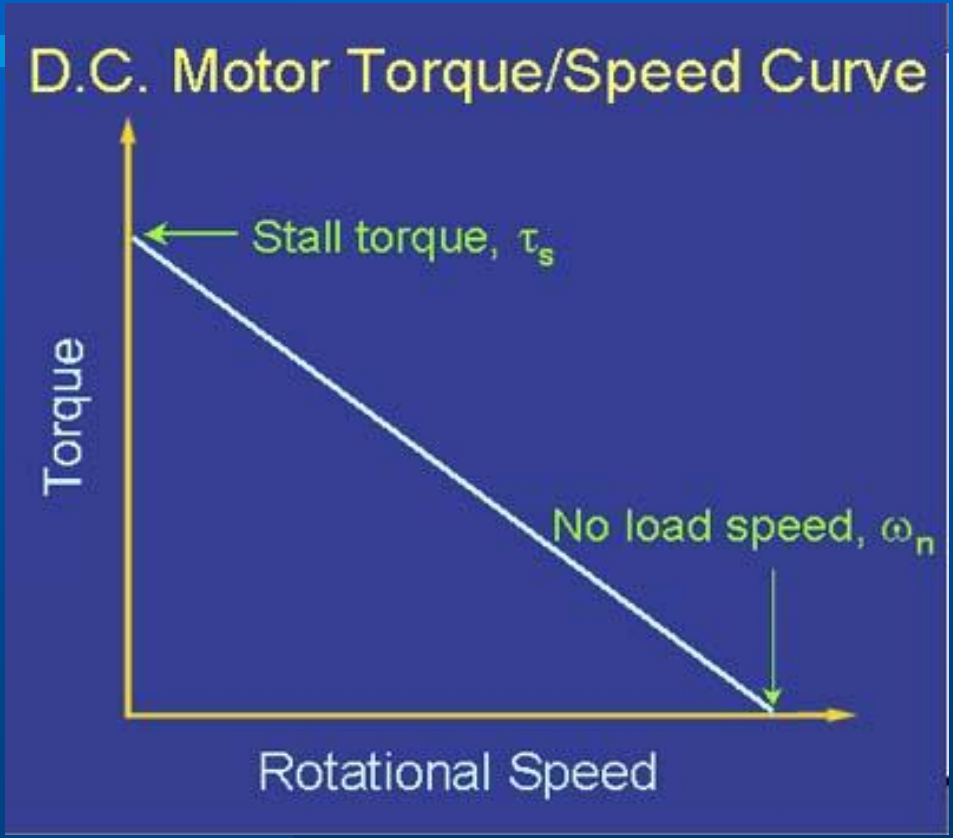
- Human Machine Interface (HMI)
  - Supervisor Control
  - Man-machine Interaction

- Displays
  - LEDs
  - Digital Displays
  - LCD
  - CRT

Next Steps:

- Decide on Actuator
- Derive your Torque – Speed requirement
- Find the ideal actuator
- 9 weeks and a day including Spring Break
- Begin thinking about your sensing architecture


The loop's magnetic field lines compress those of the permanent magnet. This provides a repulsive force between the magnet and the loop which causes the loop to spin.
axis of rotation
magnetic field lines
direction of rotation
The brushes make electrical contact with the commutator as it rotates.
South pole of magnet
S
N
North Pole of Magnet
single coil of wire.
−
+
The commutator is split so that the current in the loop reverses every half-turn, which keeps the loop spinning in one direction.
flow of electricity
From Negative D.C. terminal
From Positive D.C. terminal

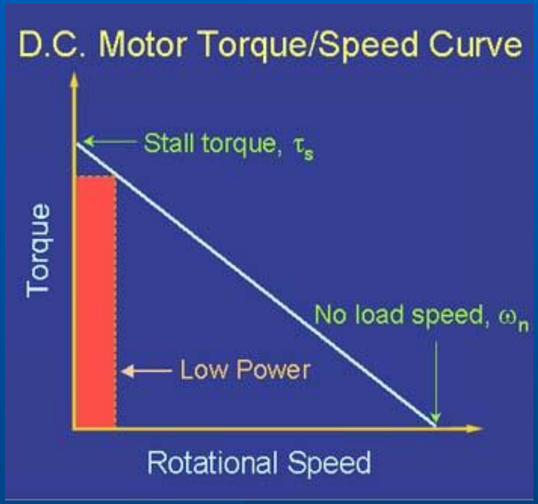


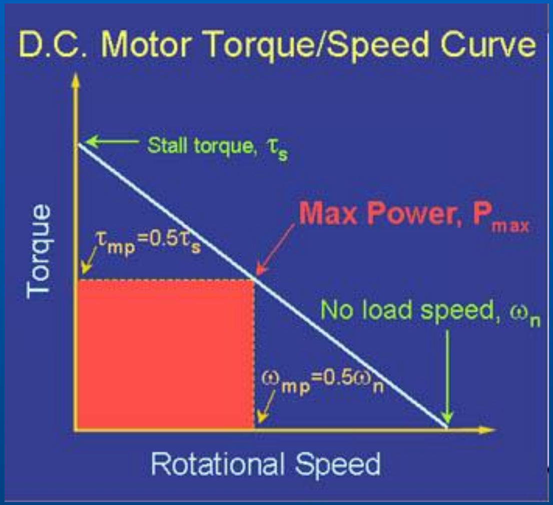
$T = T_s * (1 - \omega/\omega_n)$

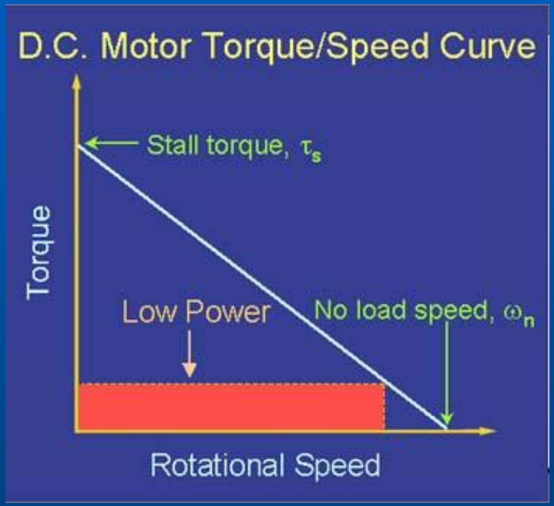
$\omega = \omega_n * (1 – T/T_s)$

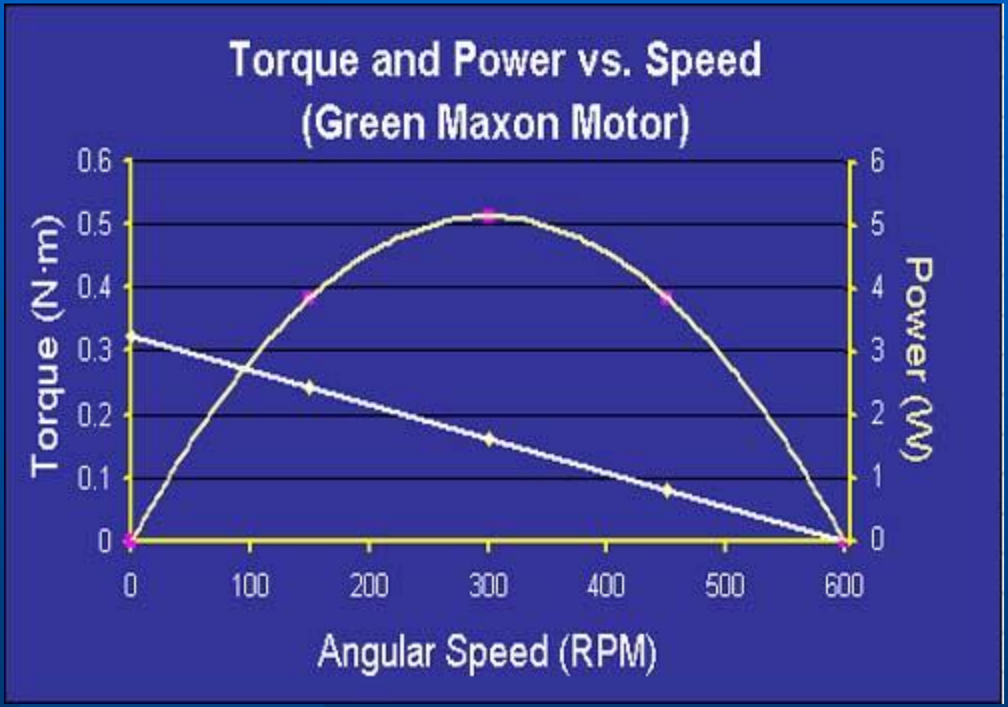
T = Motor Torque
T_s = Stall Torque

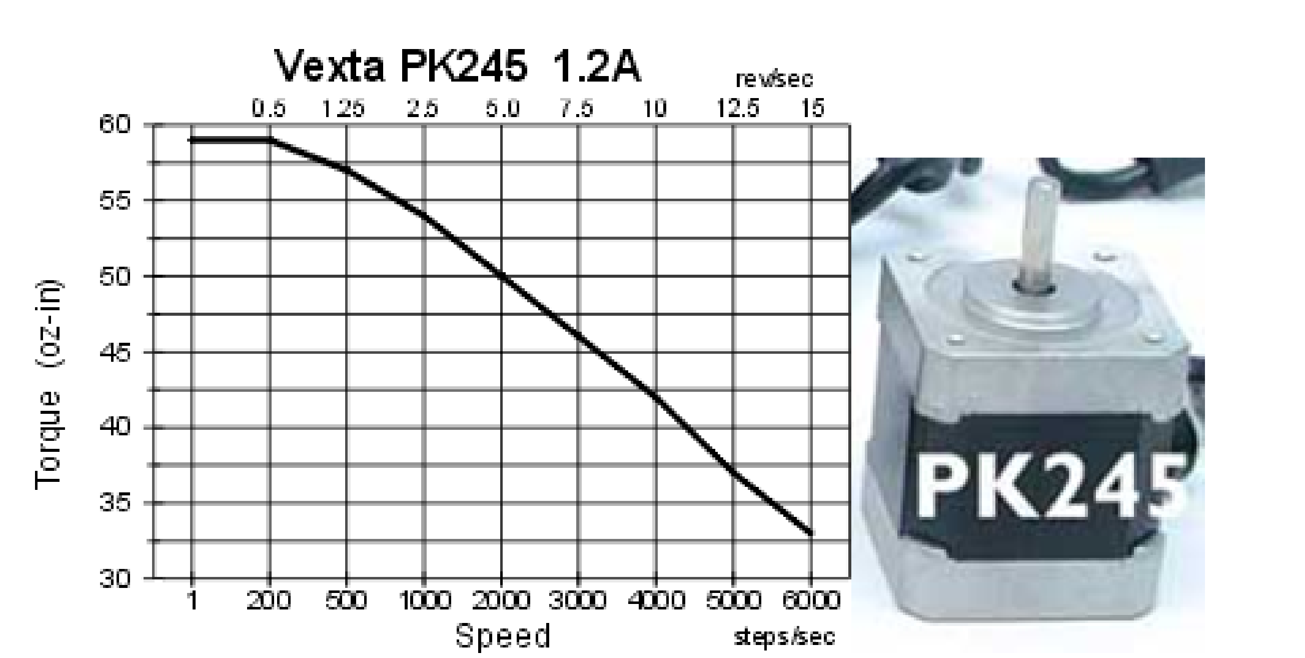
ω = Motor Speed
ω_n = No Load Speed


D.C. Motor Torque/Speed Curve
Stall torque, τ_s
Torque
No load speed, ω_n
Low Power
Rotational Speed


D.C. Motor Torque/Speed Curve
Stall torque, τ_s
Max Power, P_{max}
Torque
τ_{mp}=0.5τ_s
No load speed, ω_n
ω_{mp}=0.5ω_n
Rotational Speed


D.C. Motor Torque/Speed Curve
Stall torque, τ_s
Torque
Low Power
No load speed, ω_n
Rotational Speed


Torque and Power vs. Speed
(Green Maxon Motor)
Torque (N·m)
0.6
0.5
0.4
0.3
0.2
0.1
0
Power (W)
6
5
4
3
2
1
0
0
100
200
300
400
500
600
Angular Speed (RPM)


Vexta PK245 1.2A
rev/sec
0.5
1.25
2.5
5.0
7.5
10
12.5
15
60
55
50
45
40
35
30
Torque (oz-in)
1
200
500
1000
2000
3000
4000
5000
6000
Speed
steps/sec
PK245

Next Steps:

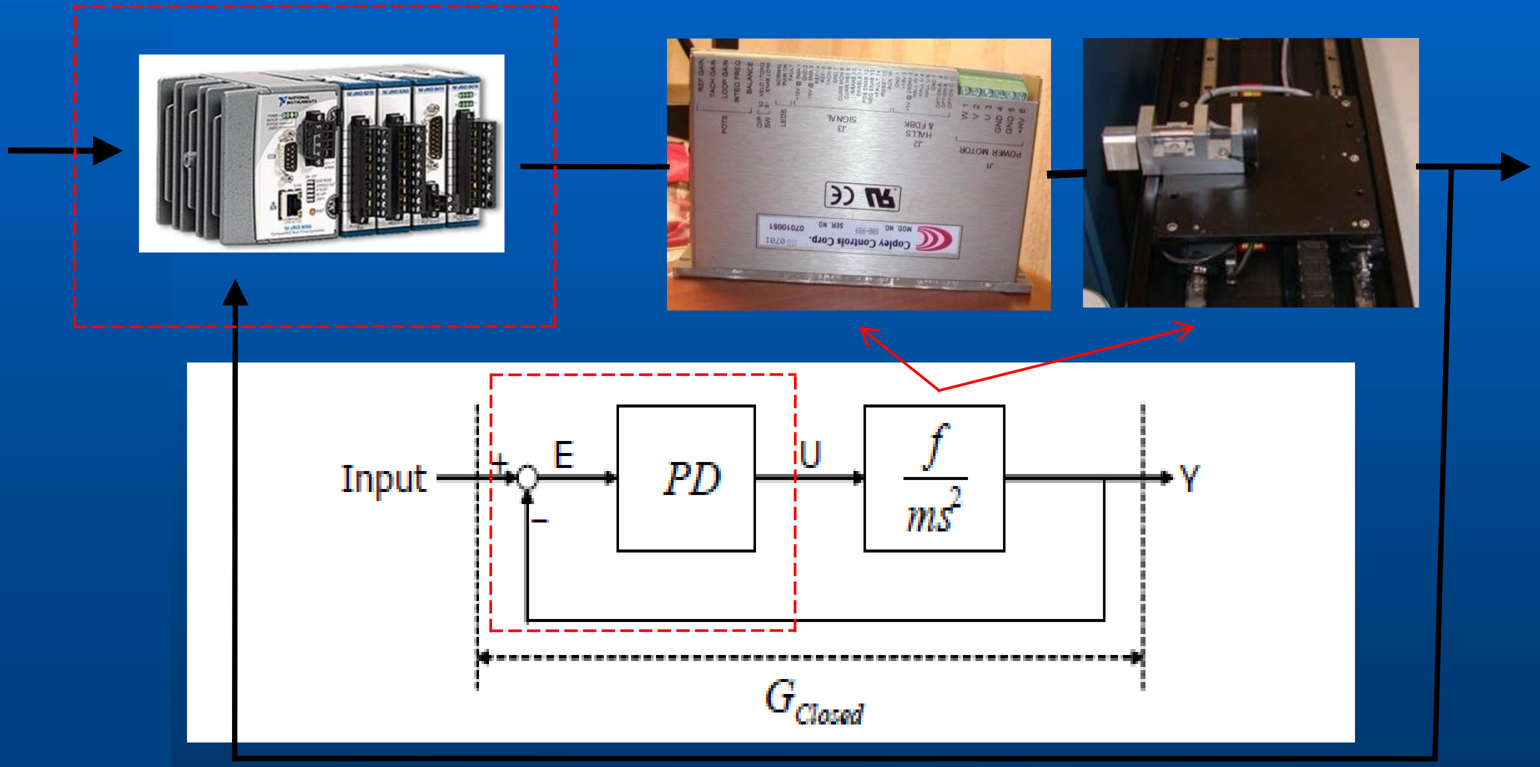
- Decide on software architecture
- Decide on Pinouts to use
- Develop simulated or simple test VI's
- 7 weeks and a day including Spring Break
- Spring Break in 2 weeks from now

Stage: Belleverman LOWBOY 260

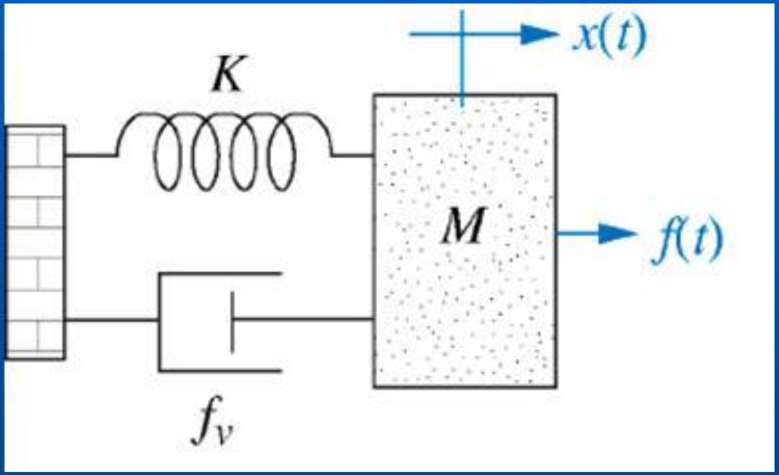
Motor: Trilogy Direct Drive
Linear Brushless Motor
96 Volts
5 Amps Peak
3 Amps Continuous

Drive: Copley Controls
Brushless DC Amplifier

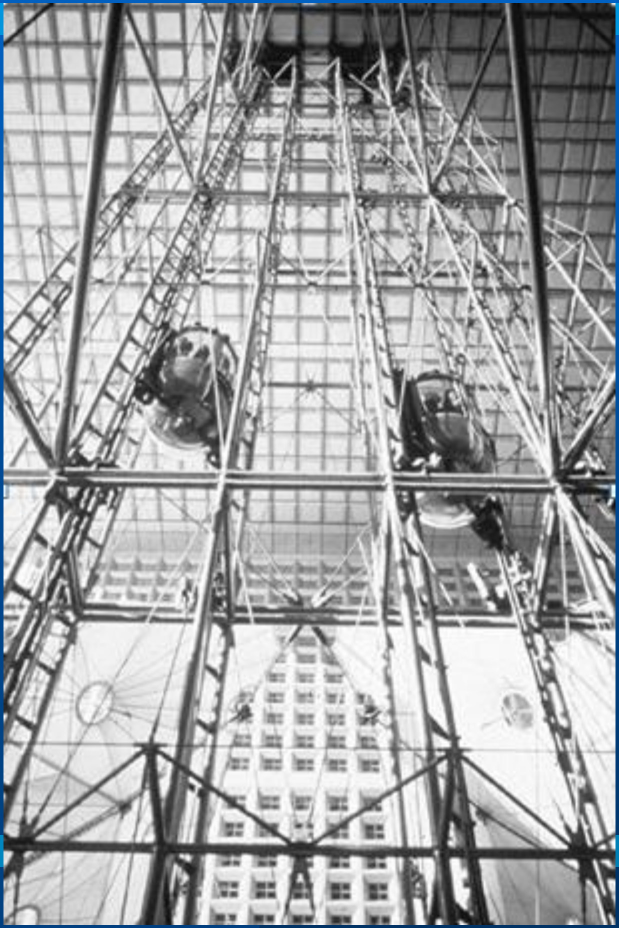
Encoder: Renishaw RG22H
1 um resolution

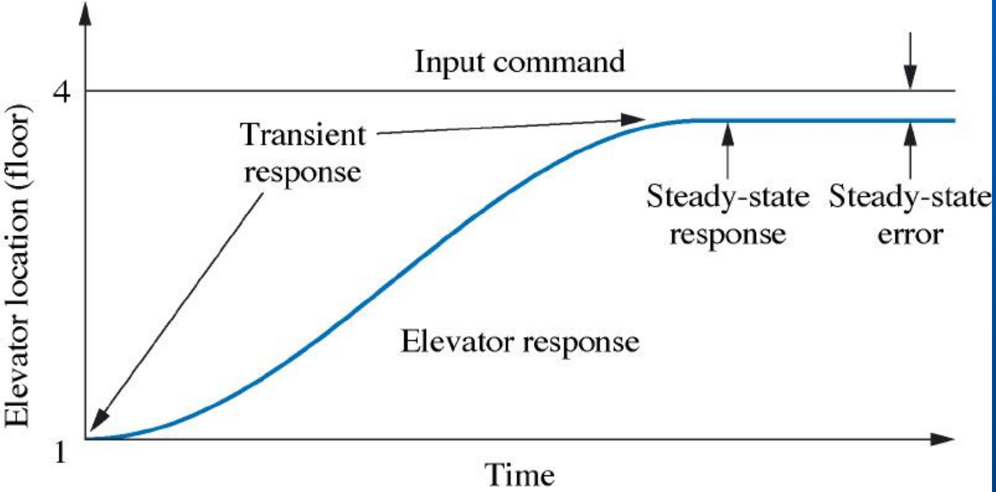

Input
+
E
PD
U
$\frac{f}{ms^2}$
Y
−
G_{Closed}
Copley Controls Corp.

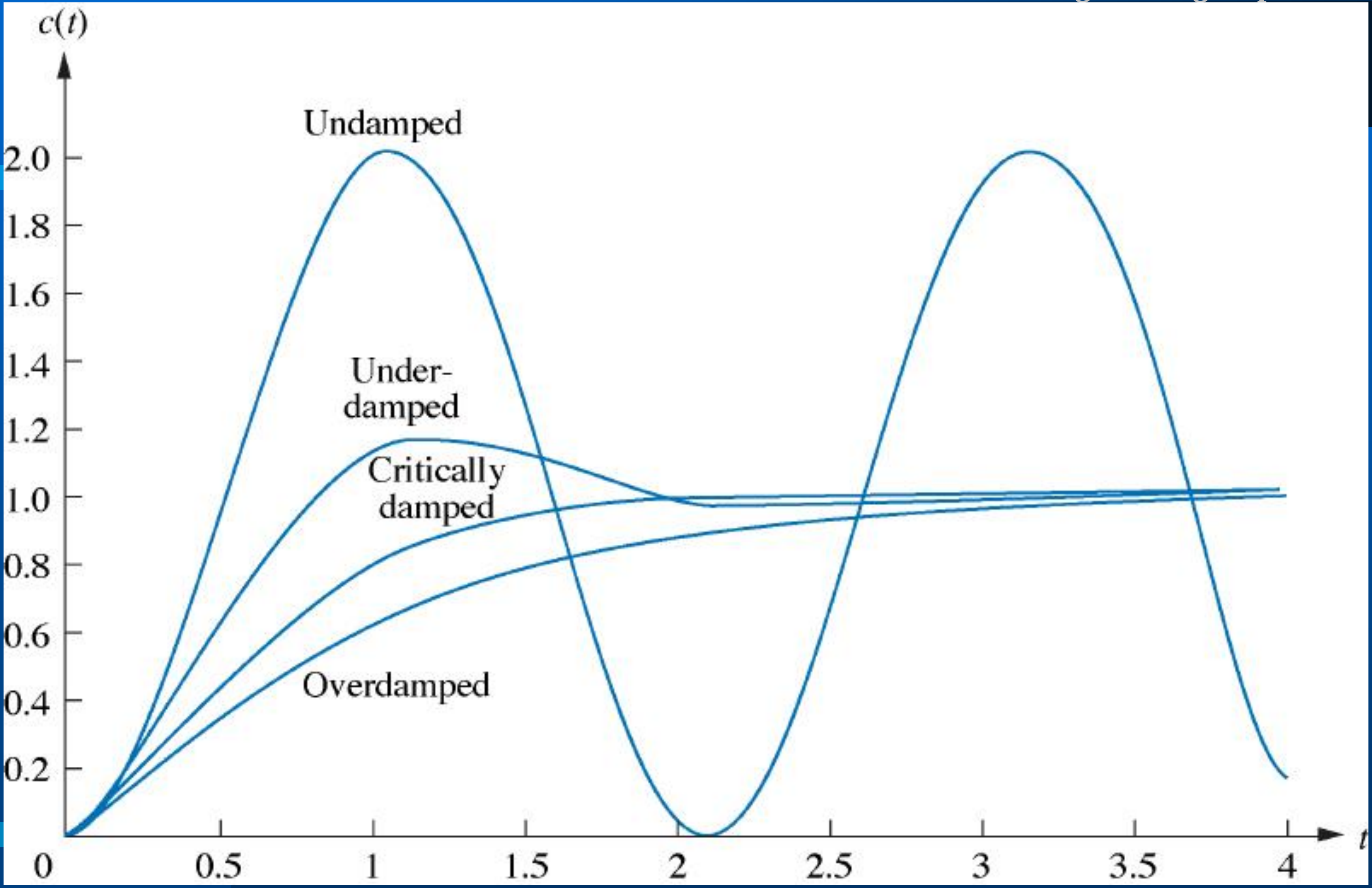
Example: Block Diagram Development

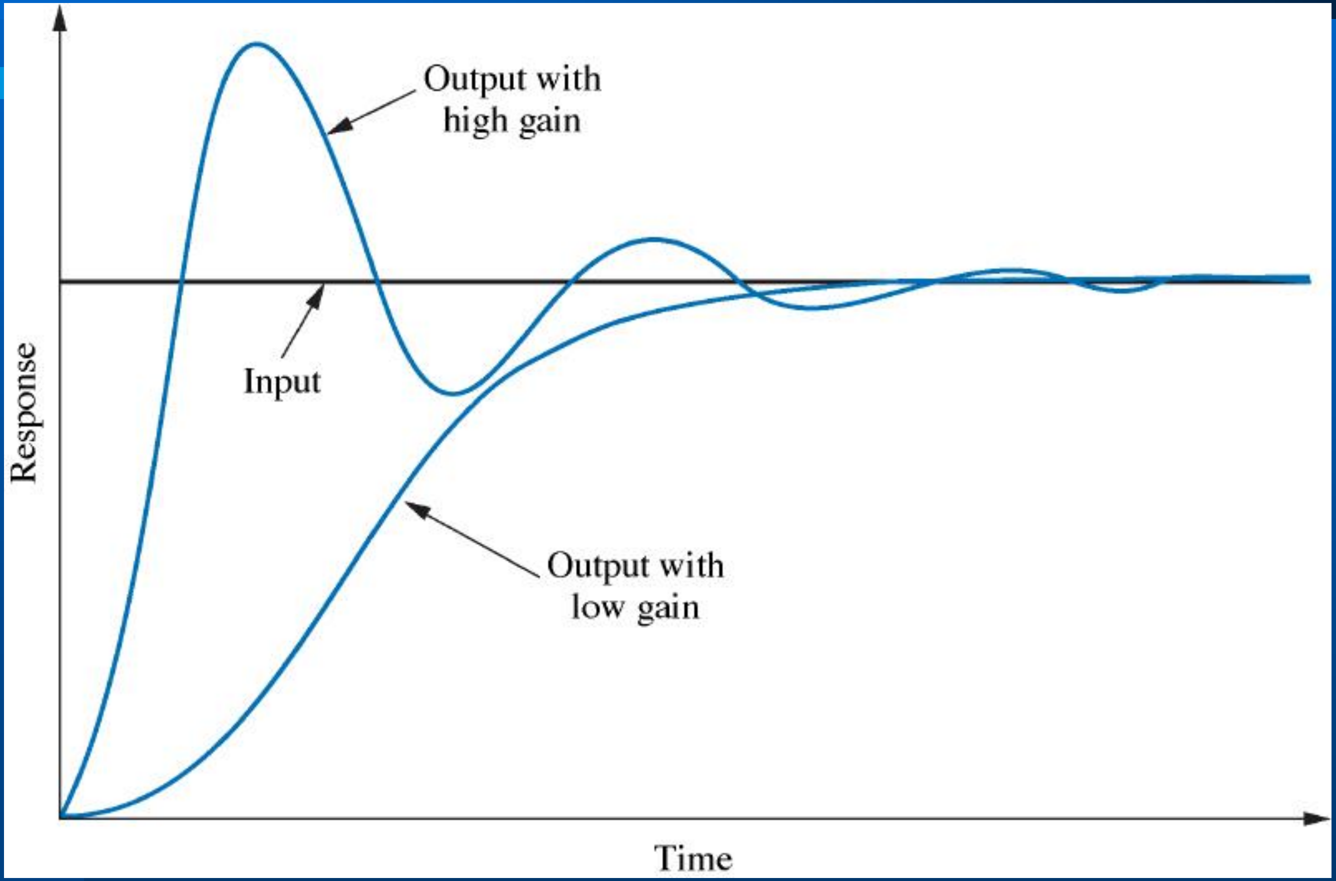


Simple Example: Move mass M from x = point A to Point B




Input command
4
Transient
response
Steady-state
response
Steady-state
error
Elevator location (floor)
Elevator response
1
Time

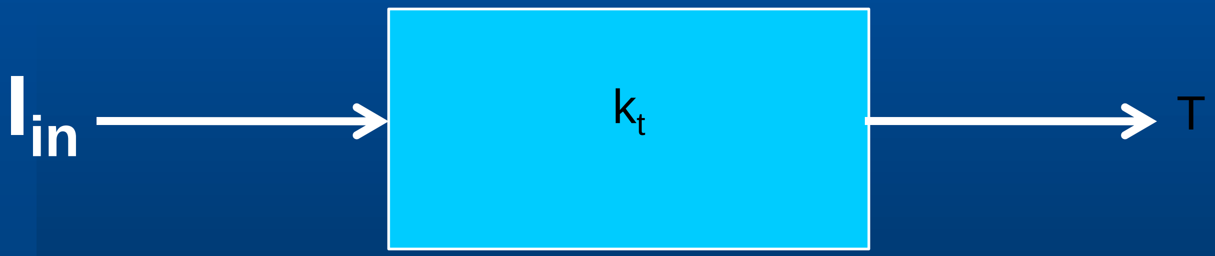

c(t)
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0
0.5
1
1.5
2
2.5
3
3.5
4
t
Undamped
Under-
damped
Critically
damped
Overdamped


Output with high gain
Input
Output with low gain
Response
Time

Permanent Magnet DC motor

$$T = k_t * I_{in}$$

where k_t = Torque Constant



Model of an ideal inertial load

$$\Sigma T = J^*\alpha$$

$$\alpha = d\omega/dt$$

$$\omega = d\theta/dt$$

where J = polar moment of inertia

α = angular acceleration

ω = angular velocity

θ = angular position

$$\Sigma T = J^*\alpha$$

$$\alpha = d\omega/dt$$

$$\Sigma T = J^*\, d\omega/dt$$

$$\omega = d\theta/dt$$

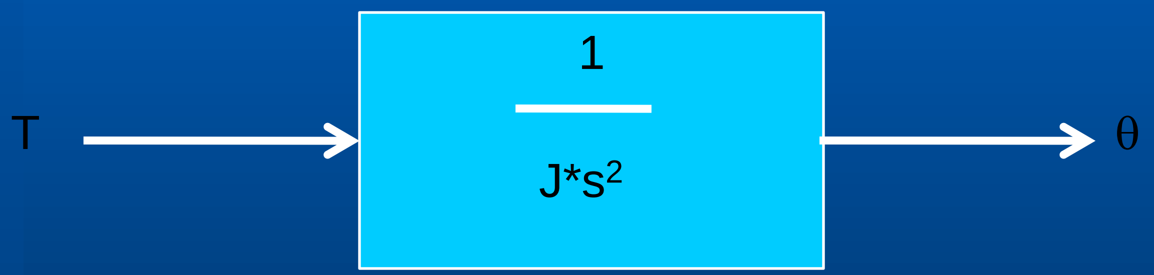
$$\Sigma T = J^*\, d^2\theta/dt^2$$

using Laplace Transform:

$$\Sigma T = J^*\, s^2\theta$$

$\Sigma T = J^* s^2\theta$

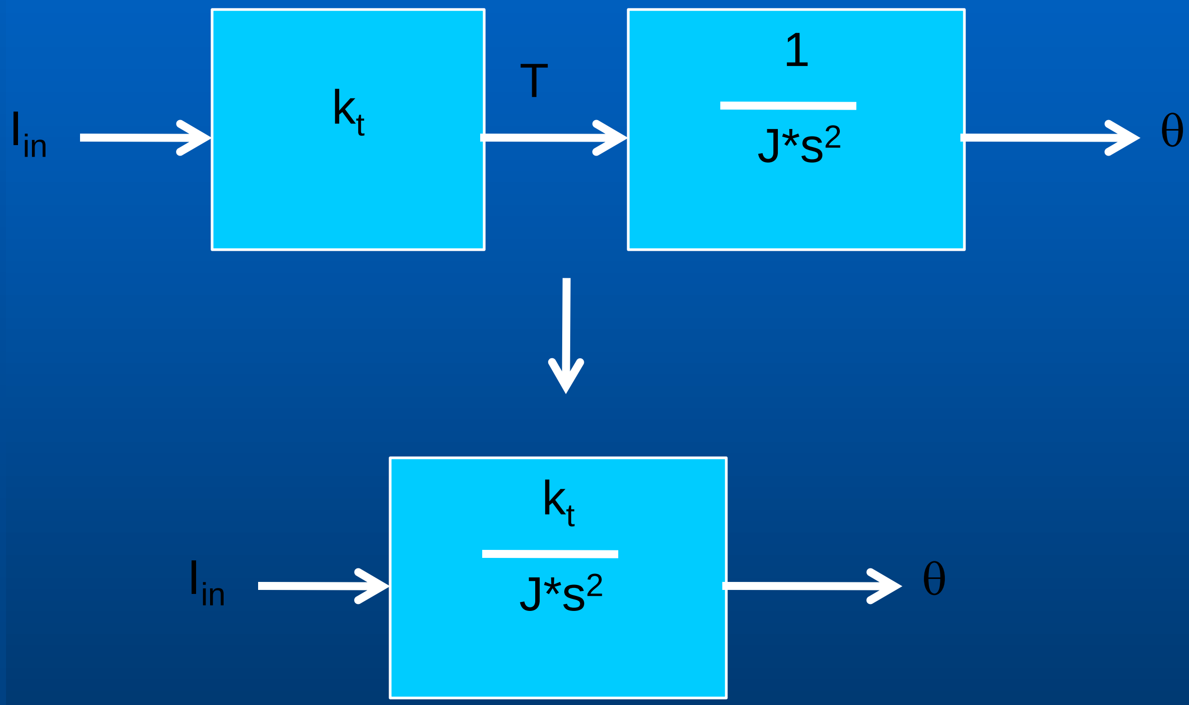
$\Sigma T = J^* s^2\theta$



T

$\dfrac{1}{J^*s^2}$

θ


I_{in}
k_t
T
$\frac{1}{J*s^2}$
θ
I_{in}
$\frac{k_t}{J*s^2}$
θ

LMD18200 H-Bridge:

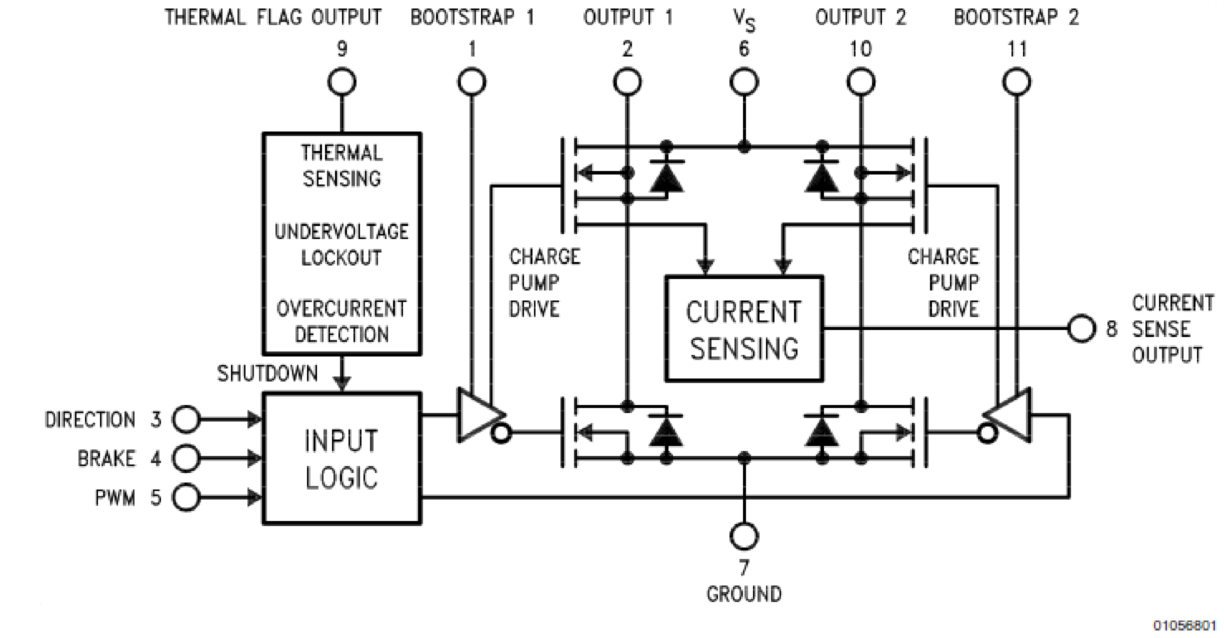
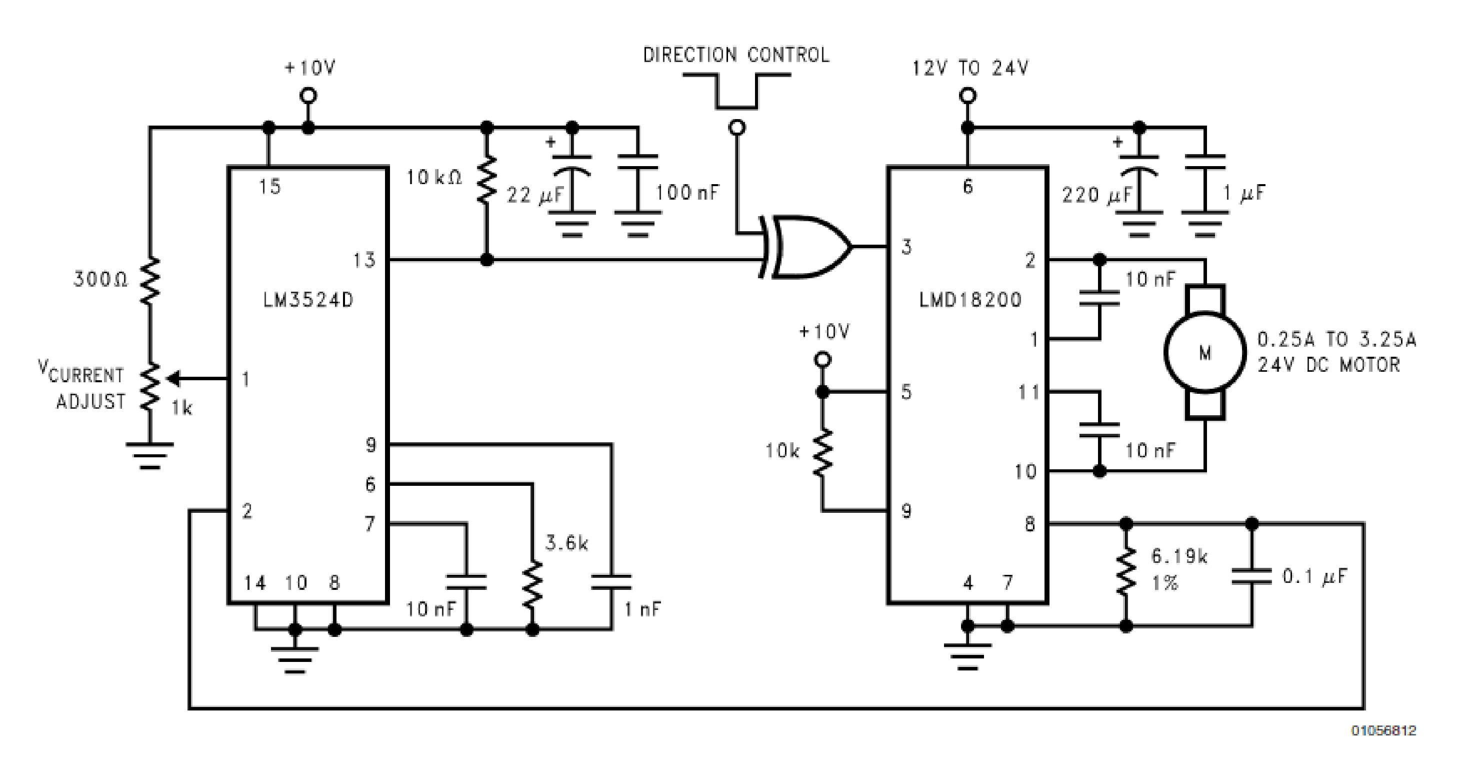
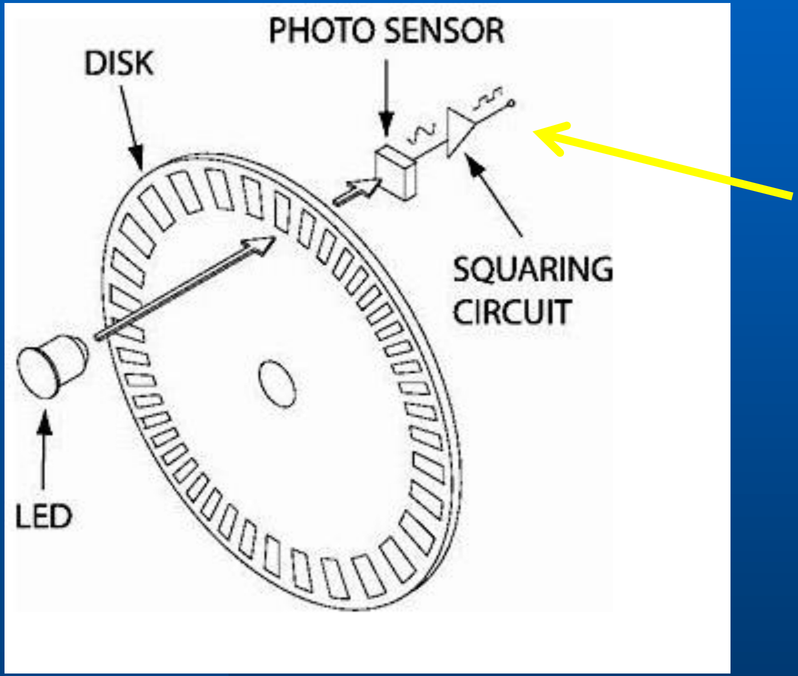


FIGURE 1. Functional Block Diagram of LMD18200

Motor with Current (Psuedo-Torque) Feedback:

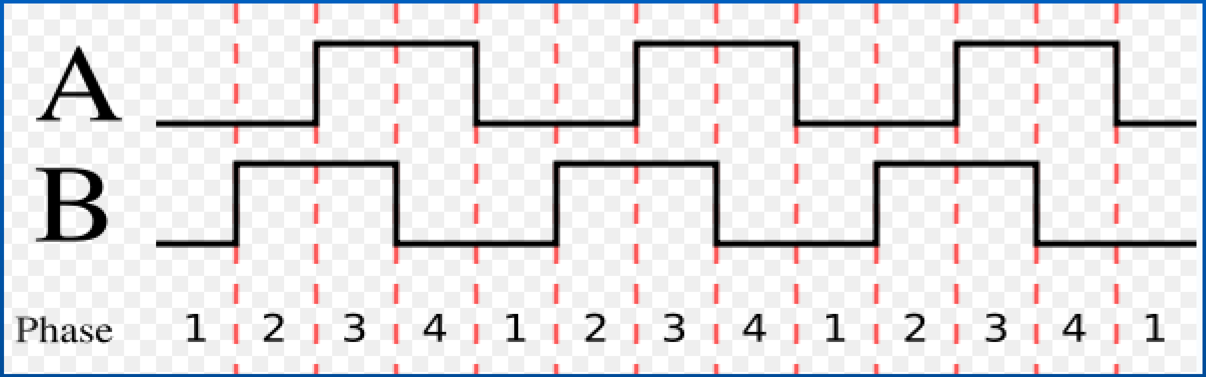




Single Channel
(Digital Tachometer)

Most Encoders has two channels

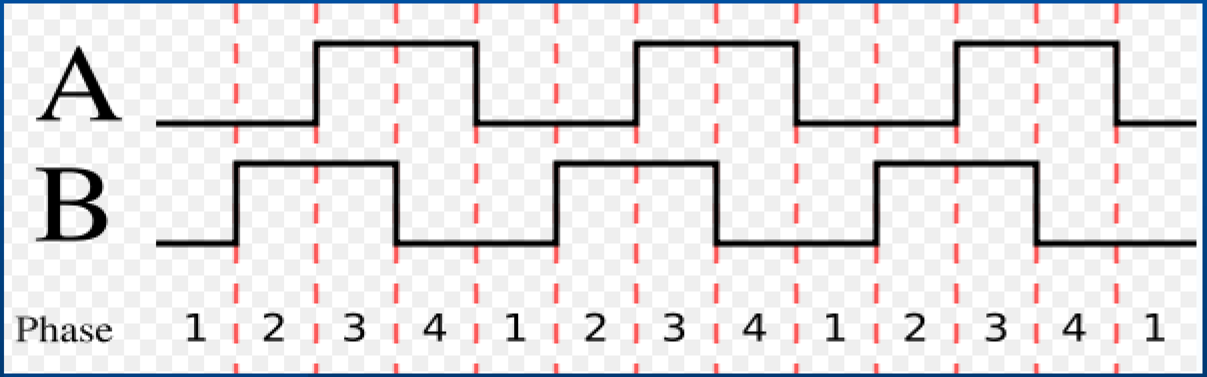
Phase A
Phase B



Quadrature : 4x resolution by counting every edge of the signal

Most common Encoders are considered as digital signals.

Ideally suitable of microprocessor or microcontroller use.



→ Digital Counter

→ Latched on rising edge of Phase A

If Phase B is high count up if low count down

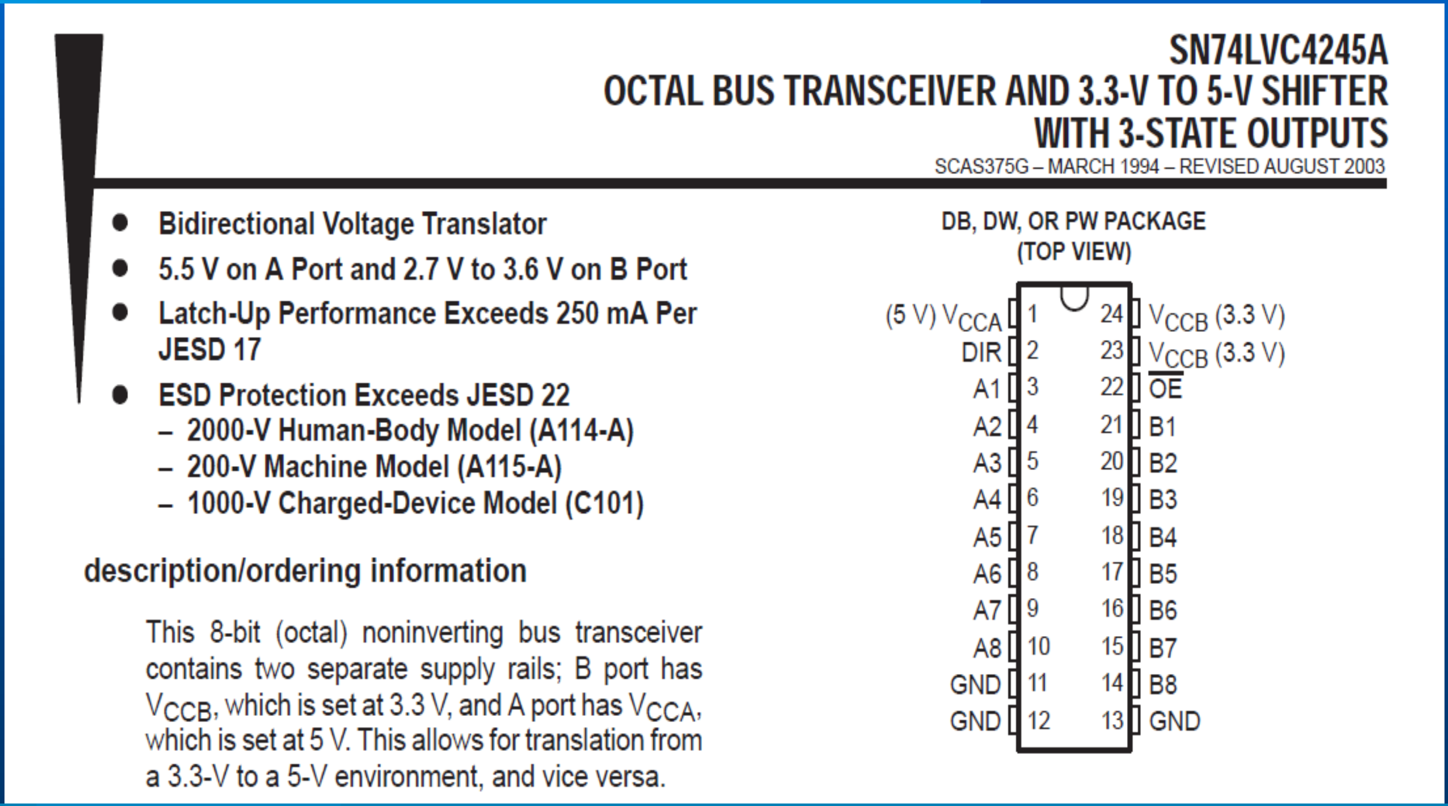
Motor Sizing and selection

- Will the motor start fast enough
  - Stall Torque
- Maximum speed requirement
  - No load Speed
- Power requirement
  - Motor Power

Other considerations

- **Operating Duty Cycle**
- **Available Power source**
- **Open or Closed Loop**
- **Required transmission**

3.3V to 5V Level Shifter:



SN74LVC4245A
OCTAL BUS TRANSCEIVER AND 3.3-V TO 5-V SHIFTER WITH 3-STATE OUTPUTS

SCAS375G – MARCH 1994 – REVISED AUGUST 2003

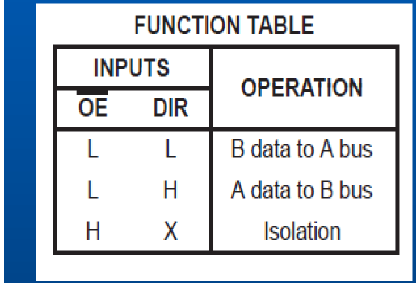
- **Bidirectional Voltage Translator**
- **5.5 V on A Port and 2.7 V to 3.6 V on B Port**
- **Latch-Up Performance Exceeds 250 mA Per JESD 17**
- **ESD Protection Exceeds JESD 22**
  - **2000-V Human-Body Model (A114-A)**
  - **200-V Machine Model (A115-A)**
  - **1000-V Charged-Device Model (C101)**

DB, DW, OR PW PACKAGE (TOP VIEW)

(5 V) V_{CCA}	1	24	V_{CCB} (3.3 V)
DIR	2	23	V_{CCB} (3.3 V)
A1	3	22	$\overline{OE}$
A2	4	21	B1
A3	5	20	B2
A4	6	19	B3
A5	7	18	B4
A6	8	17	B5
A7	9	16	B6
A8	10	15	B7
GND	11	14	B8
GND	12	13	GND

description/ordering information

This 8-bit (octal) noninverting bus transceiver contains two separate supply rails; B port has V_{CCB}, which is set at 3.3 V, and A port has V_{CCA}, which is set at 5 V. This allows for translation from a 3.3-V to a 5-V environment, and vice versa.



FUNCTION TABLE

INPUTS		OPERATION
$\overline{OE}$	DIR	
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

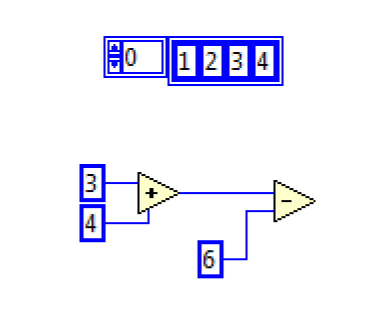
Means of combination :

C :

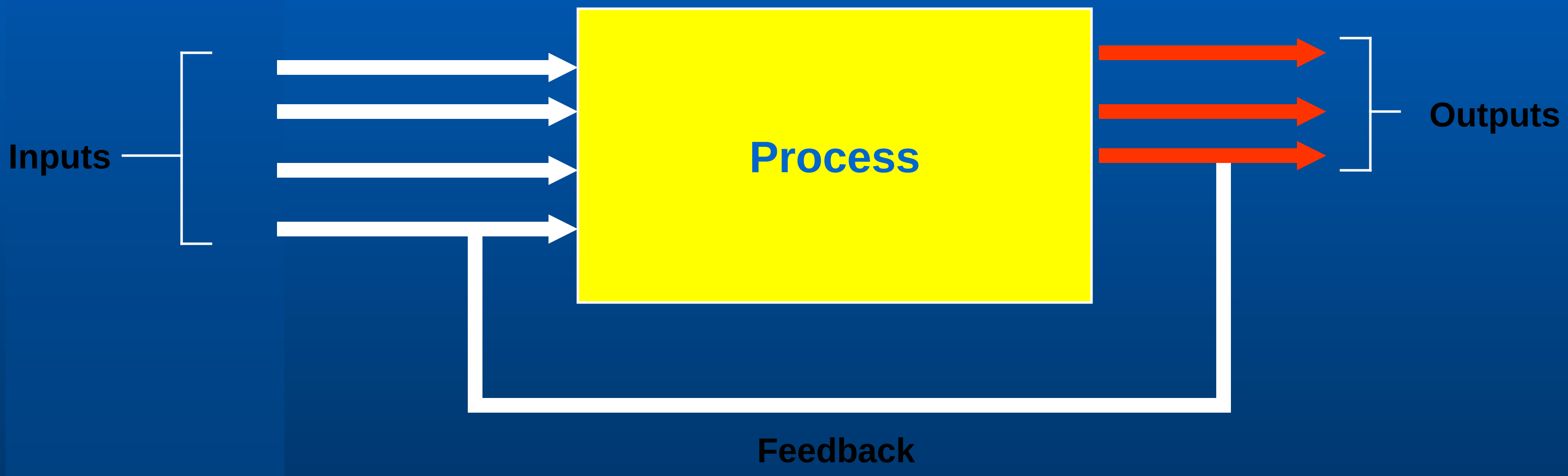
- Arrays – [1,2,3,4]

- Instructions -- 3+4-6….

LabVIEW:



Generalized Architecture :



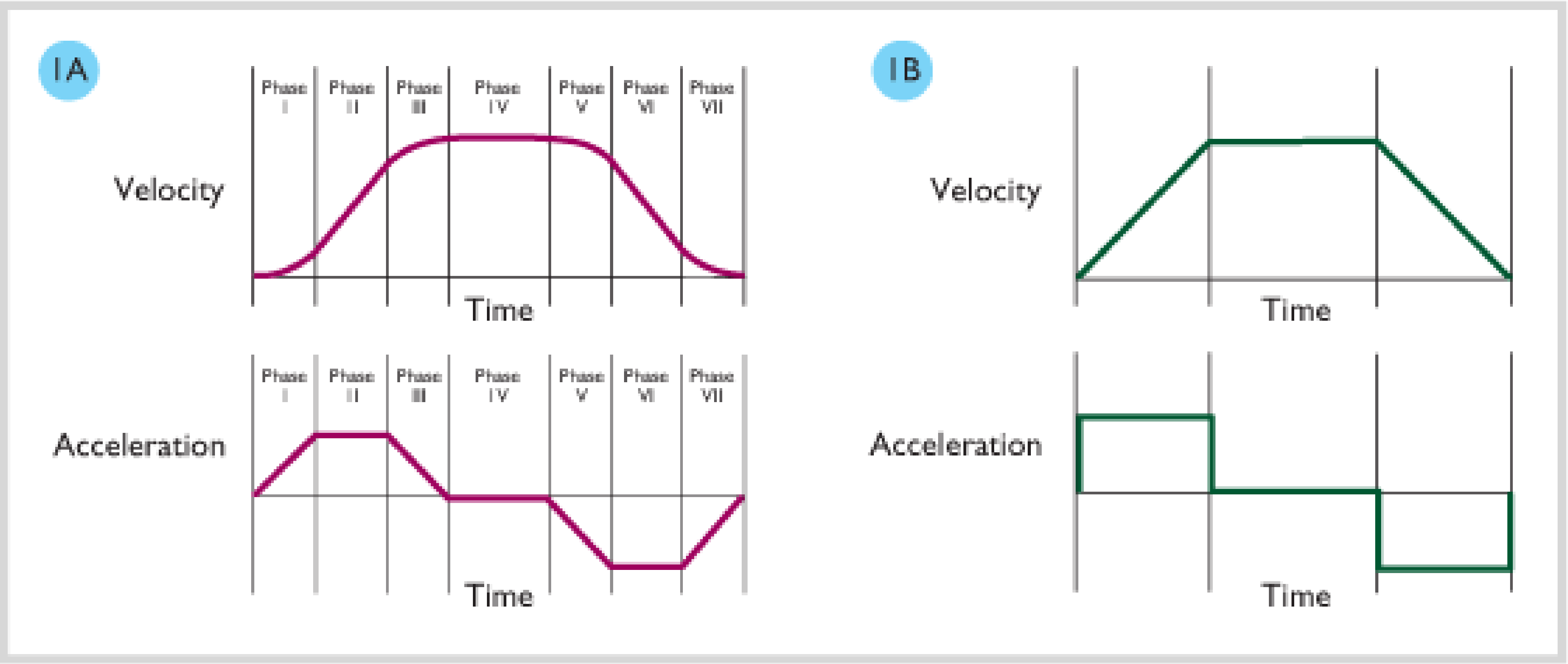
CLAD Exam:

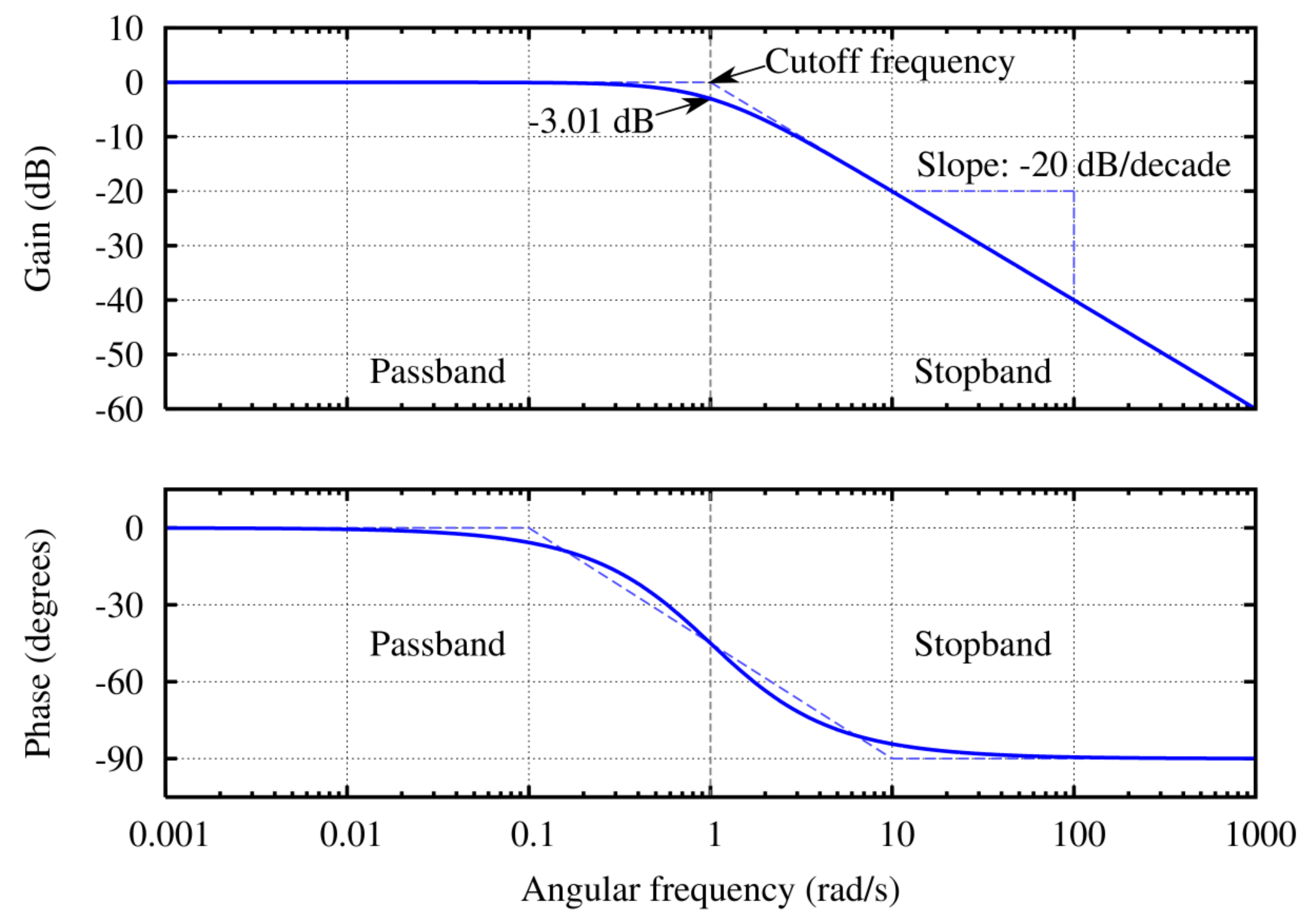
April 28th 8:00 AM 277 Cory
1 hour multiple choice
40 questions
Need 70% to pass
80% of test is from sample test

Final Presentations:

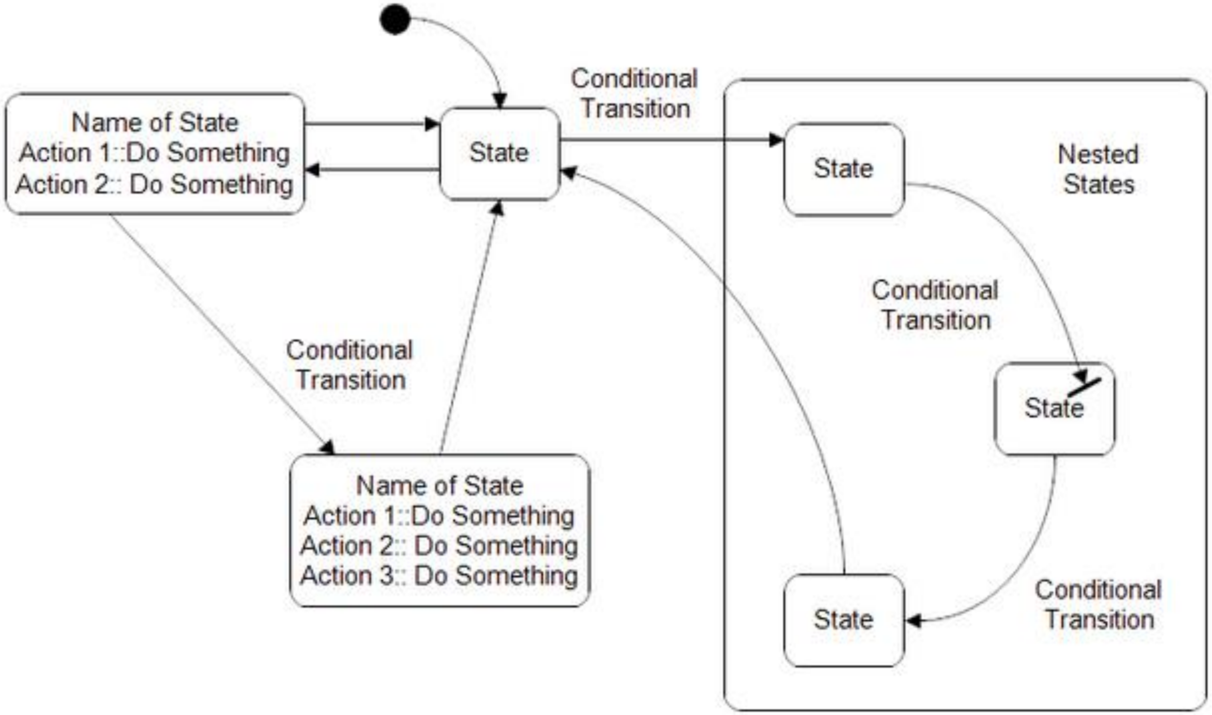
May 1st for those taking ME102B concurrently

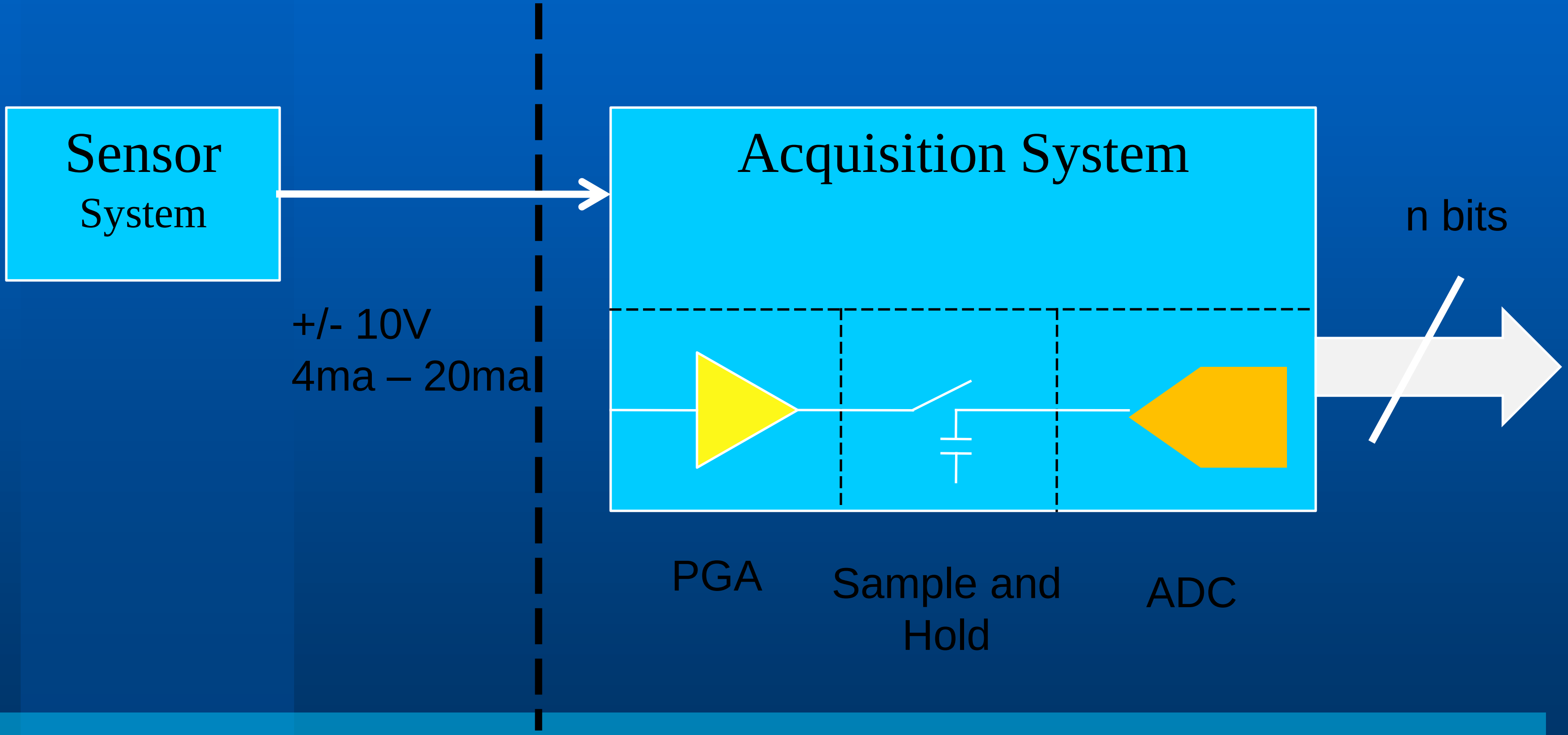
May 8th for others in 120 Hesse Hall
10:00 AM to whenever….


IA
Velocity
Acceleration
Phase I
Phase II
Phase III
Phase IV
Phase V
Phase VI
Phase VII
Time
Time
IB
Velocity
Acceleration
Time
Time


10
0
-10
-20
-30
-40
-50
-60
Gain (dB)
Cutoff frequency
-3.01 dB
Slope: -20 dB/decade
Passband
Stopband
0
-30
-60
-90
Phase (degrees)
Passband
Stopband
0.001
0.01
0.1
1
10
100
1000
Angular frequency (rad/s)

Software Design
State Transition Diagram





Sensor
System
+/- 10V
4ma – 20ma
Acquisition System
n bits
PGA
Sample and
Hold
ADC

PGA (Programmable Gain Amplifier)

The PGA accurately interfaces to and scales the signal presented at the connector for the analog-to-digital converter (ADC).

Sample and Hold

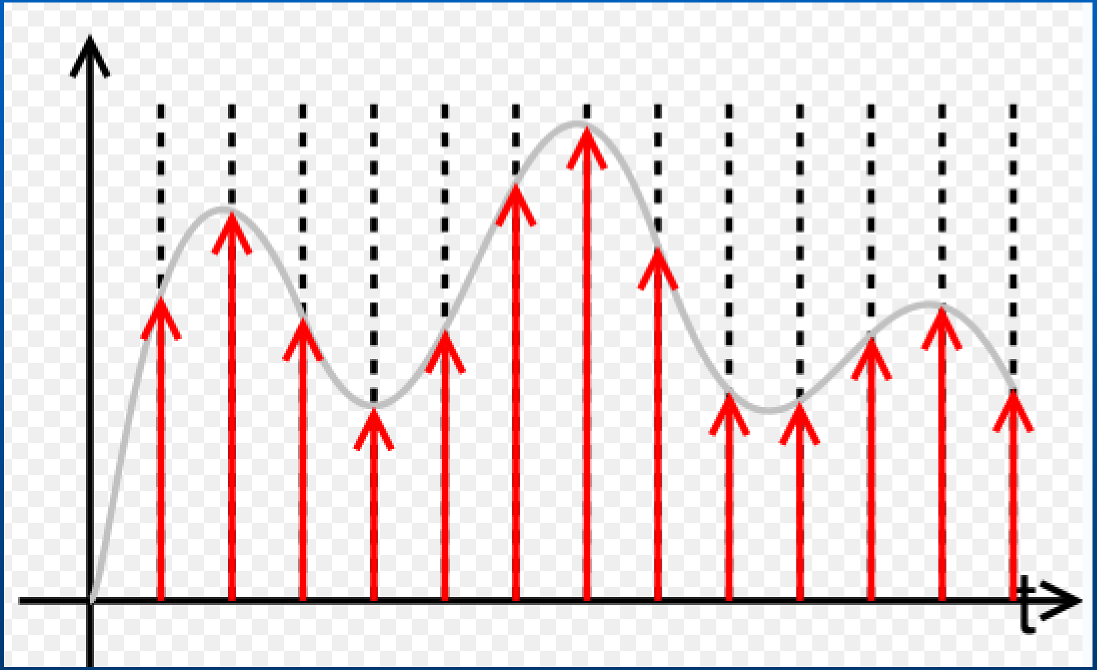
A circuit that acquires and stores an analog voltage on a capacitor for a short period of time. A high-quality sample and hold should hold that voltage constant for as long as possible until the A/D converter has "measured" the voltage.

(ADC) Analog to Digital Converter

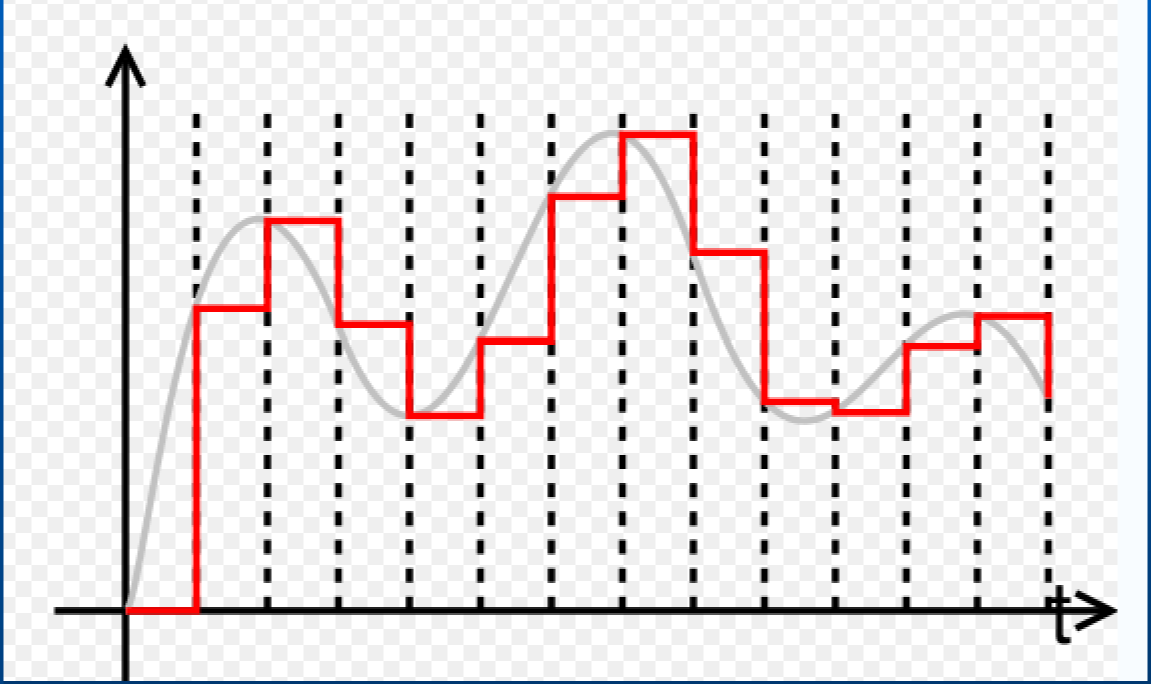
An electronic circuit that produces a digital output directly proportional to an analog signal input.

Two main parameters of interest in A/D converters are the rate at which the converter can sample analog values, and the resolution at which it can resolve the values.

Sampled Data



Quantized Data :



ADC Resolution:

N Bits	resolution	Signal Range for +/- 10V
8	256	~78 mv
10	1024	~19.6 mv
12	4096	~4.88 mv
16	65536	~0.305 mv

DIGITAL BITS:

	SIGNED		UNSIGNED	
N Bits	**MIN**	**MAX**	**MIN**	**MAX**
4	-8	7	0	15
8	-128	127	0	255
16	-32768	32767	0	65535
32	-2147483648	2147483647	0	4294967295
n	(-2^{n-1})	$(2^{n-1} -1)$	0	$(2^{n} -1)$

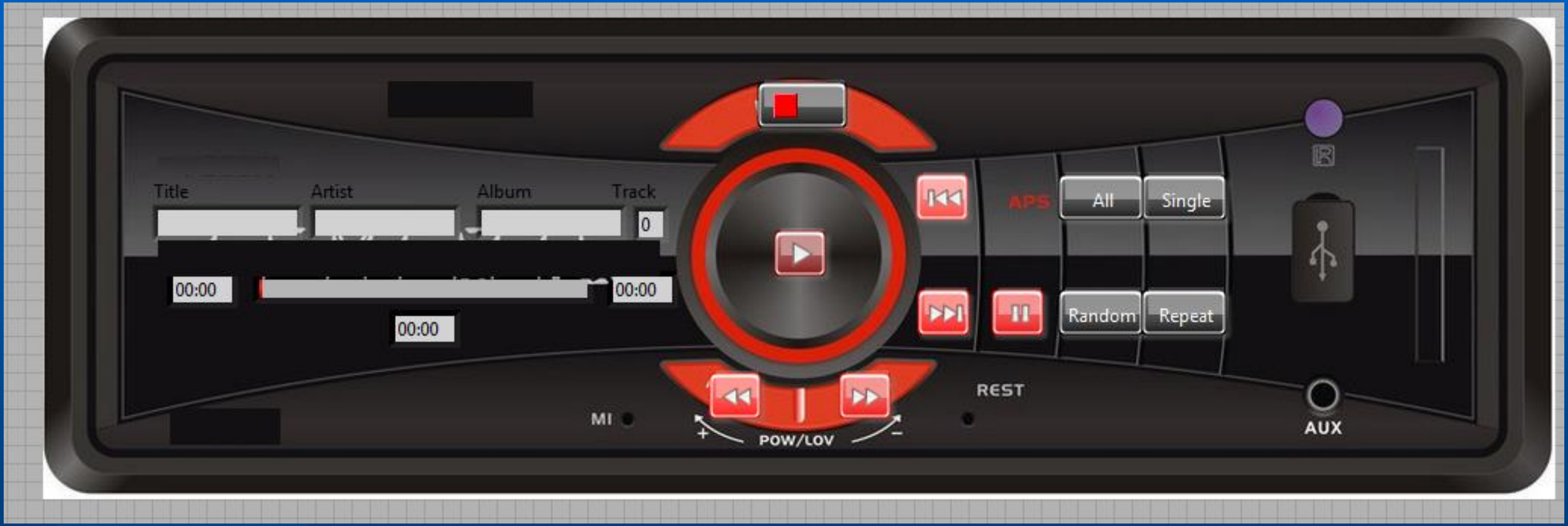
DIGITAL BITS:

	SIGNED		UNSIGNED	
N Bits	**MIN**	**MAX**	**MIN**	**MAX**
4	-8	7	0	15
8	-128	127	0	255
16	-32768	32767	0	65535
32	-2147483648	2147483647	0	4294967295
n	(-2^{n-1})	$(2^{n-1} -1)$	0	$(2^{n} -1)$

Putting Bits into perspective:

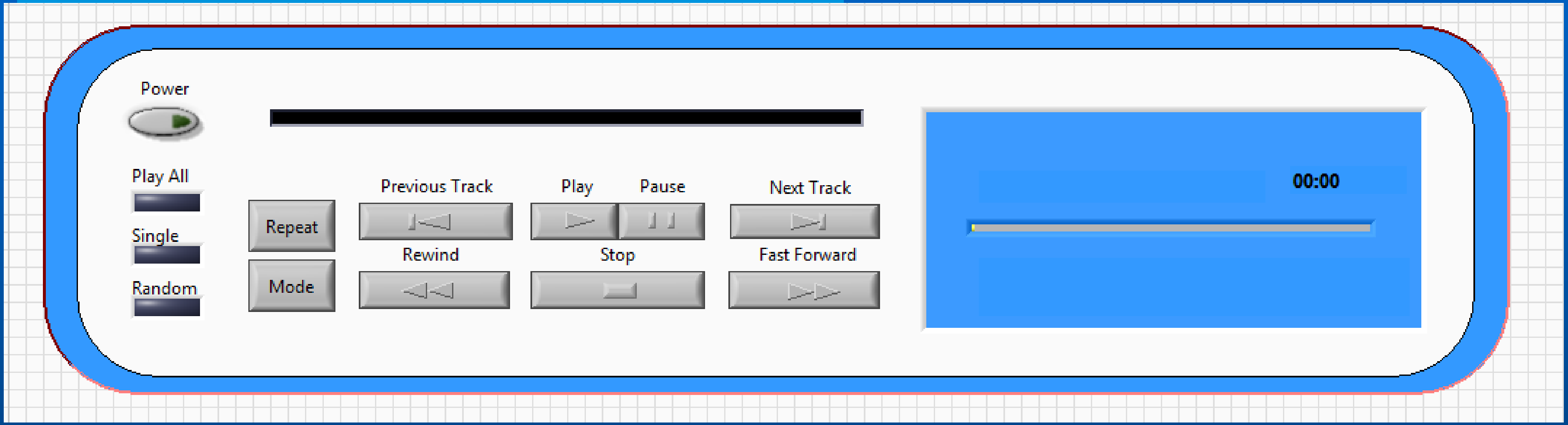
Measuring at 1 ms resolution using unsigned integer:

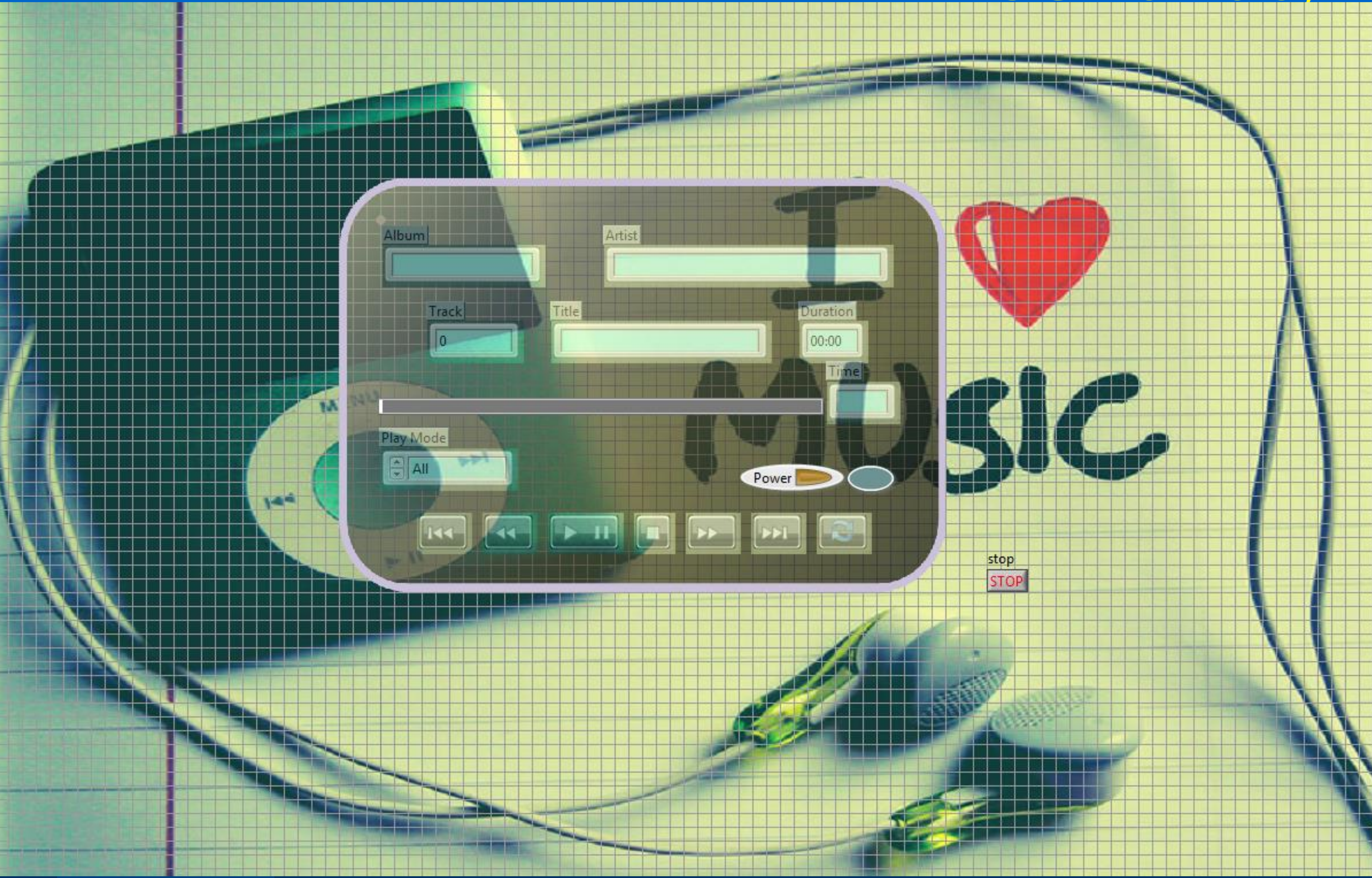
	ms	
4	15	0.015 s
8	255	0.255 s
16	65535	1 m 5.535 s
32	4294967295	~49.7 d
64	18446744073709551615	??
128	~3.4028236692093846346337460743177e+38	Nevermind



Title
Artist
Album
Track 0

00:00
00:00
00:00

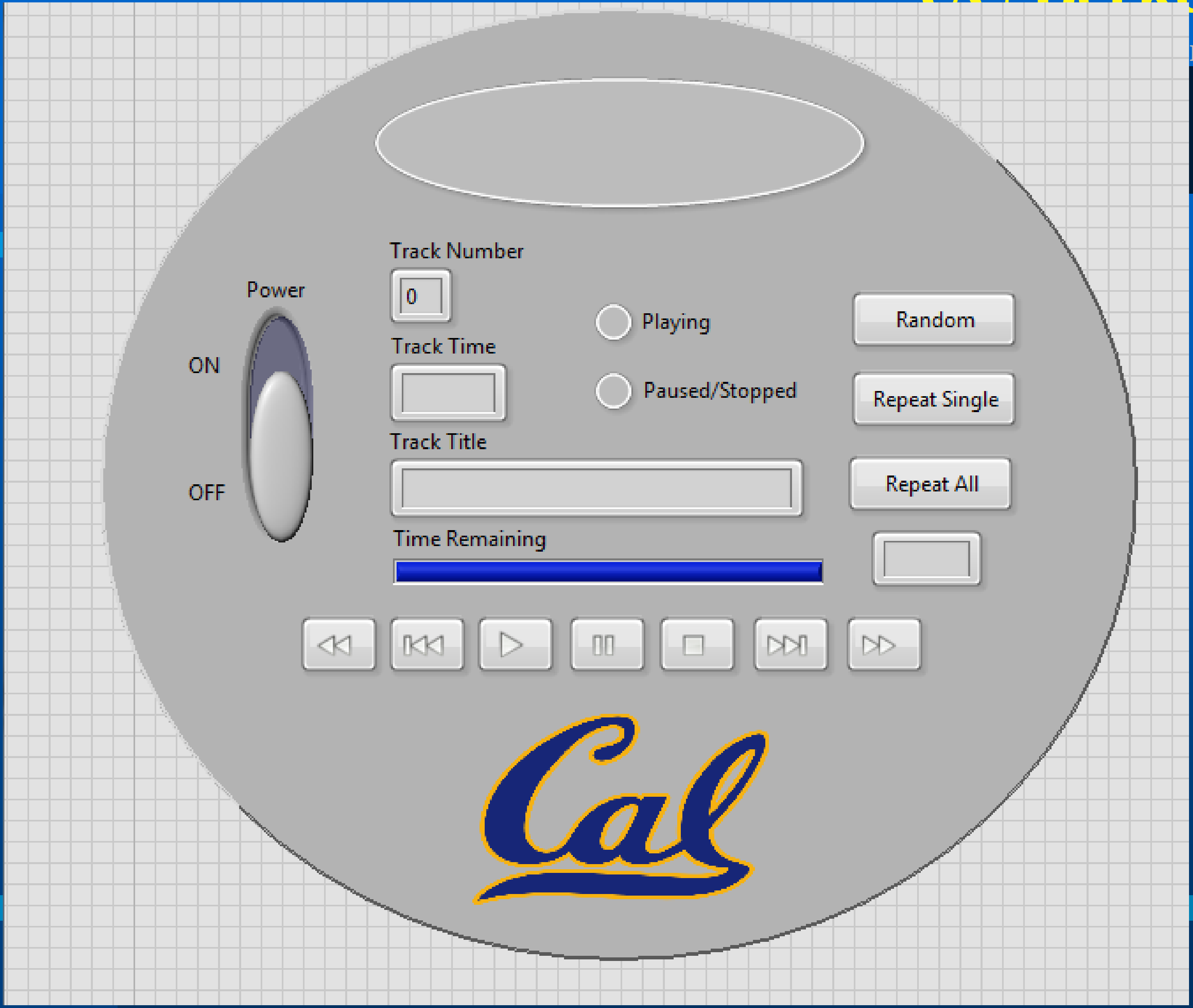

Power
Play All
Single
Random
Repeat
Mode
Previous Track
Play
Pause
Next Track
Rewind
Stop
Fast Forward
00:00



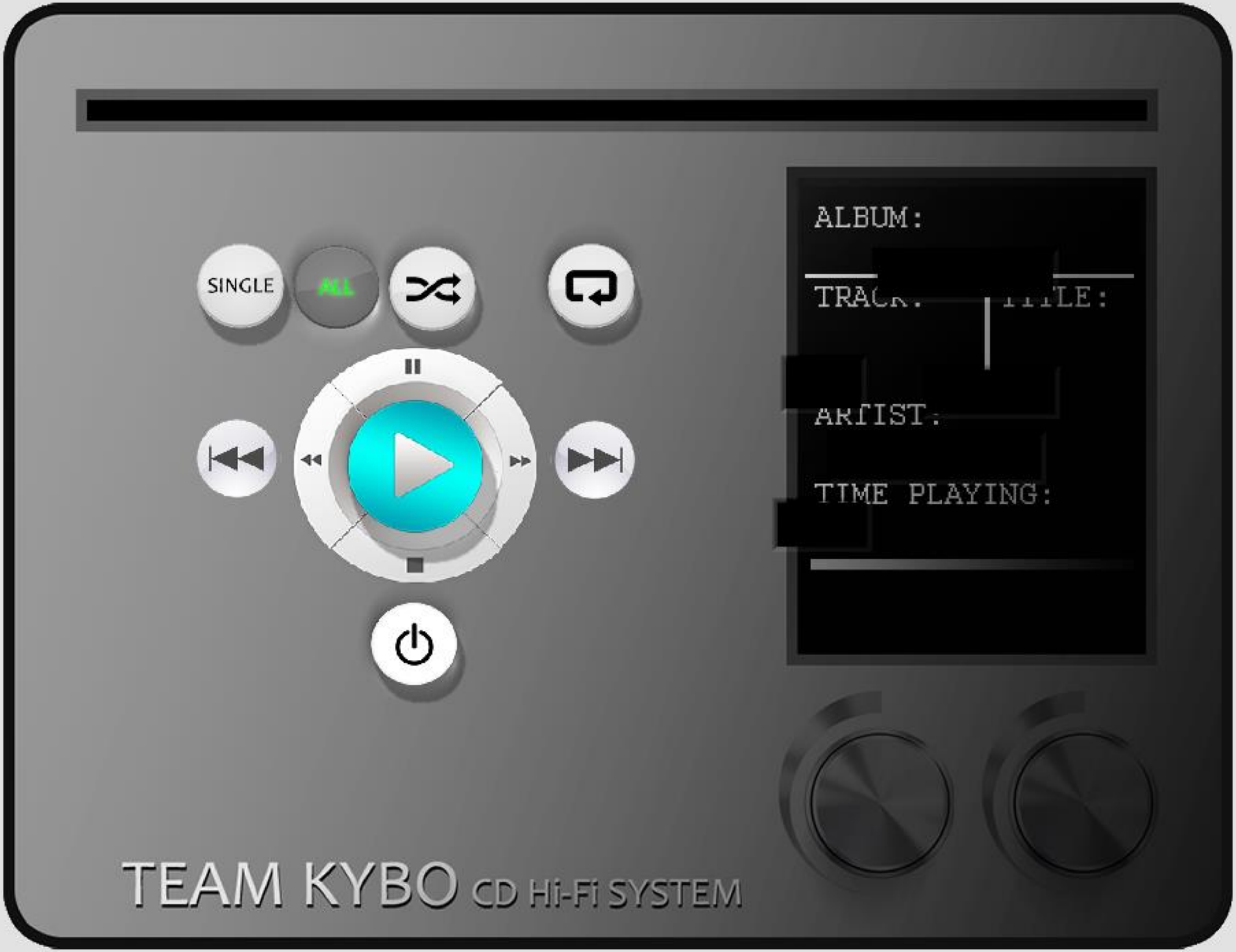
Track: 0

Duration: 00:00

Play Mode: All

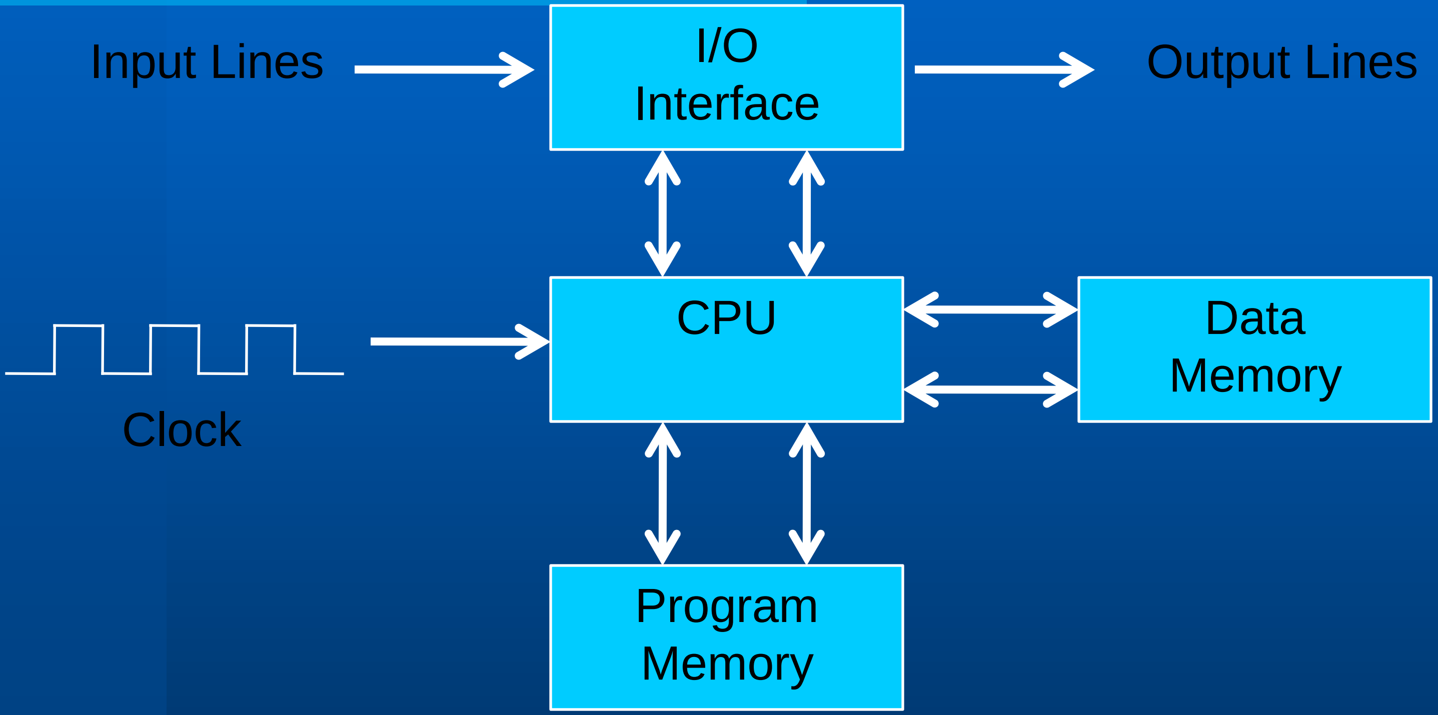


Track Number
0



SINGLE
ALL
ALBUM:
TRACK:
TITLE:
ARTIST:
TIME PLAYING:
TEAM KYBO CD Hi-Fi SYSTEM

Simplified Model of an Early Processor



Input Lines

I/O Interface

Output Lines

Clock

CPU

Data Memory

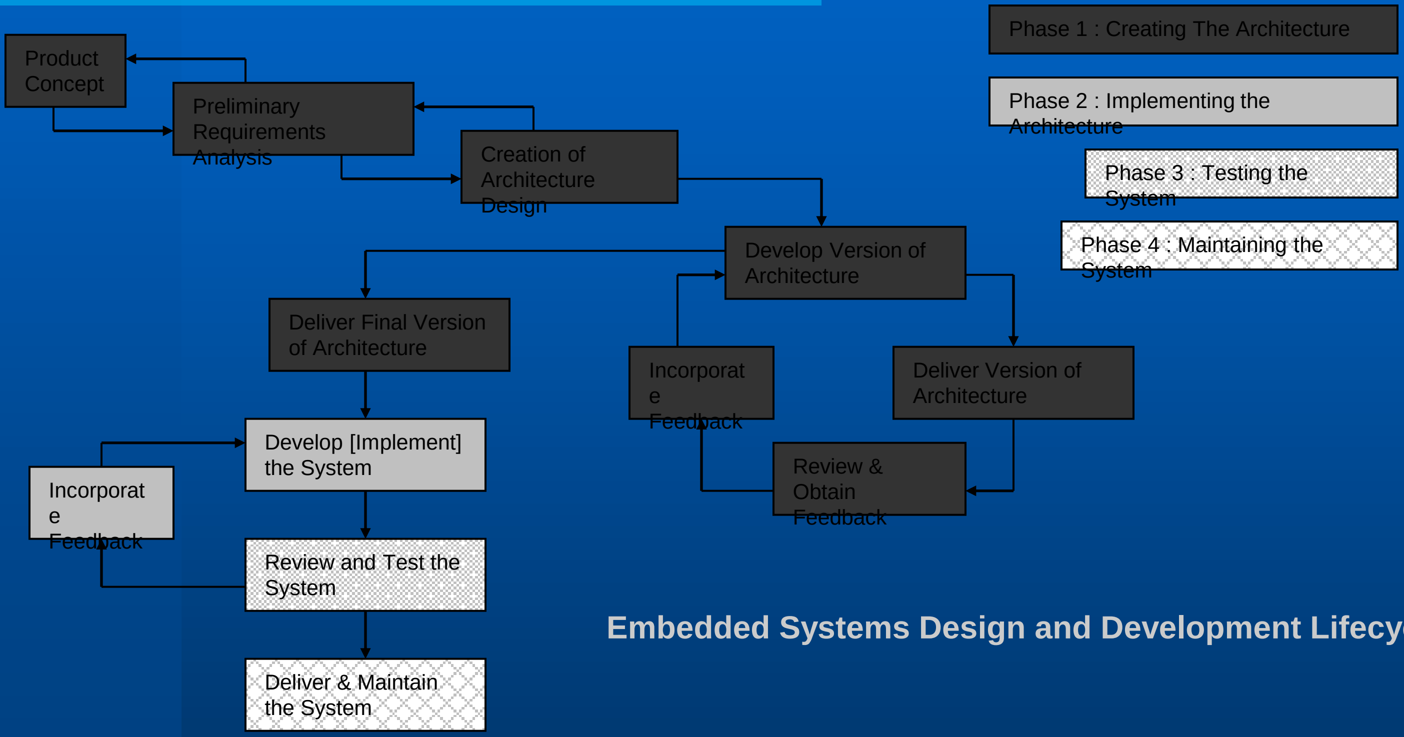
Program Memory

CPU : Central Processing Unit

UC Berkeley

Mechanical Engineering Department

Embedded Systems Design


Phase 1 : Creating The Architecture
Phase 2 : Implementing the Architecture
Phase 3 : Testing the System
Phase 4 : Maintaining the System
Product Concept
Preliminary Requirements Analysis
Creation of Architecture Design
Develop Version of Architecture
Incorporate Feedback
Deliver Version of Architecture
Review & Obtain Feedback
Deliver Final Version of Architecture
Develop [Implement] the System
Incorporate Feedback
Review and Test the System
Deliver & Maintain the System

Embedded Systems Design and Development Lifecycle Model

What is Embedded Systems Architecture?

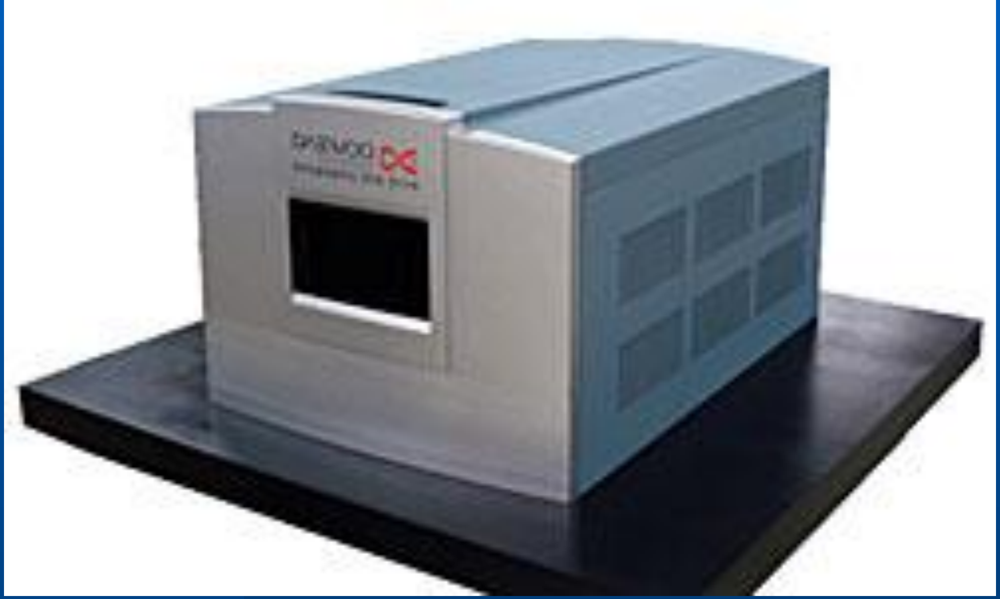
- *An abstraction of the embedded device that represents the embedded system as some combination of interacting elements.*
  - *- physically represented as structures*
  - *- many types of structures*
    - *Layered, Kernel, Decomposition, Client/Server, Process, …*

Sum of Structures = Embedded Architecture

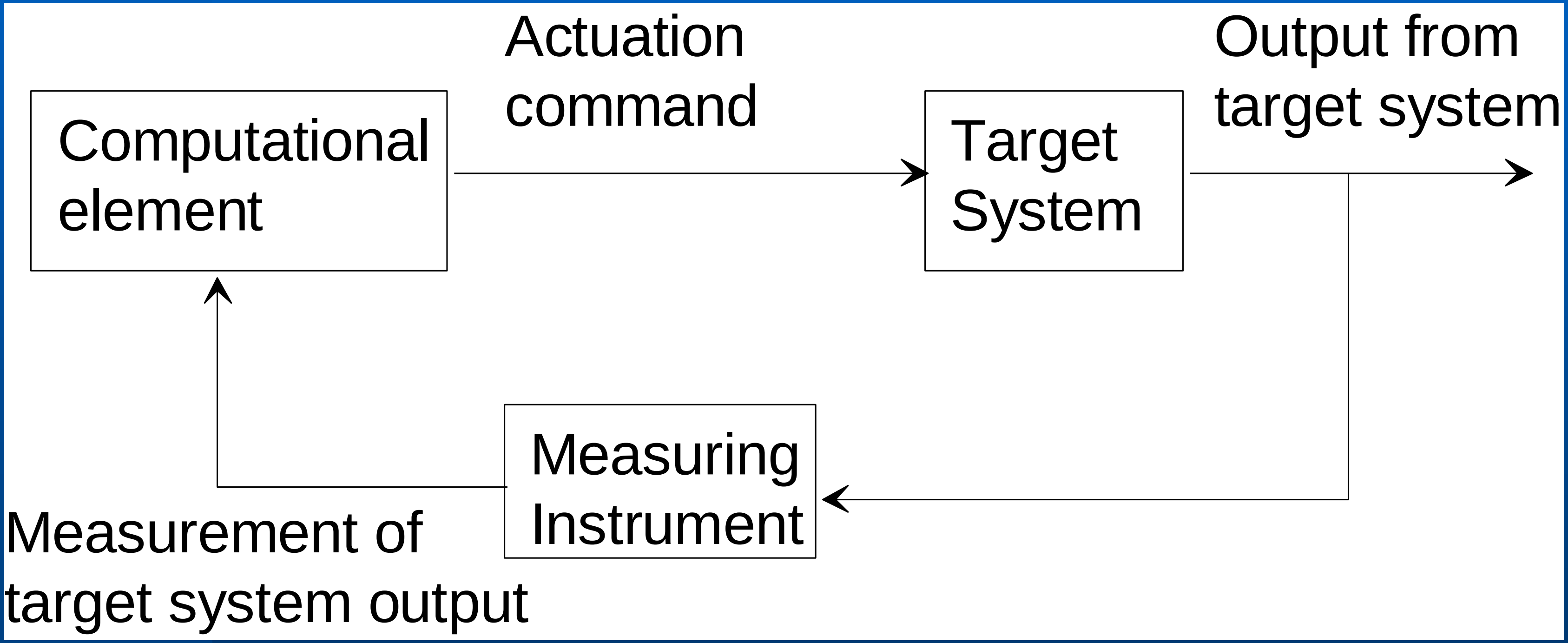
- *Why care about the architecture of an embedded system?*

6 Stages of Creating an Embedded Architecture

- **Many industry popular methodologies for creating architectures (adaptable to embedded systems)**
  - **Rational Unified Process (RUP), Attribute Driven Design (ADD), Object Oriented Process (OOP), …**
- **More Pragmatic Approach [the best of all worlds]**
  - **Stage 1 : Having a Solid Technical Base**
  - **Stage 2 : Understanding the ABCs of Embedded Systems**
  - **Stage 3 : Defining the Architectural Patterns & Reference Models**
  - **Stage 4 : Creating the Architectural Structures**
  - **Stage 5 : Documenting the Architecture**
  - **Stage 6 : Analyzing & Evaluating the Architecture**



Daewoo Halographic Data Storage System



Computational element

Actuation command

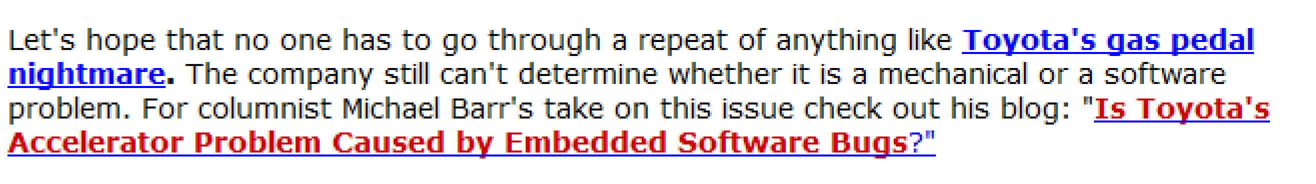
Target System

Output from target system

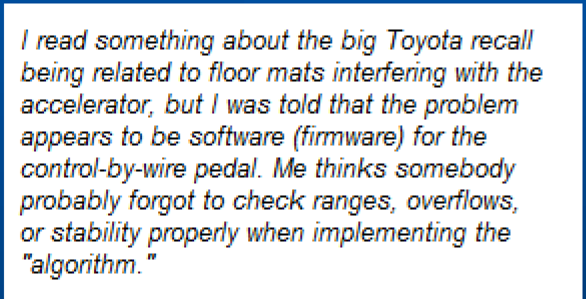
Measuring Instrument

Measurement of target system output

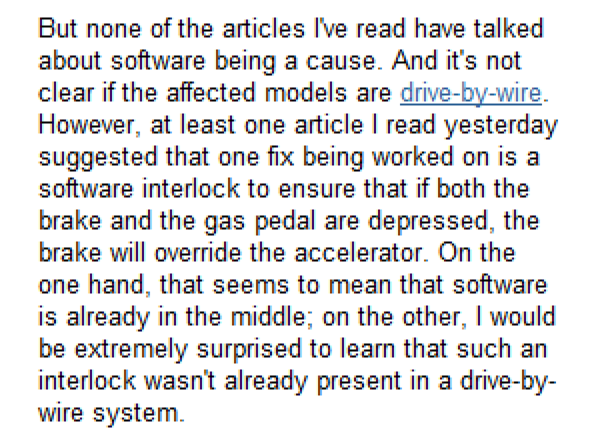
Today in the News:



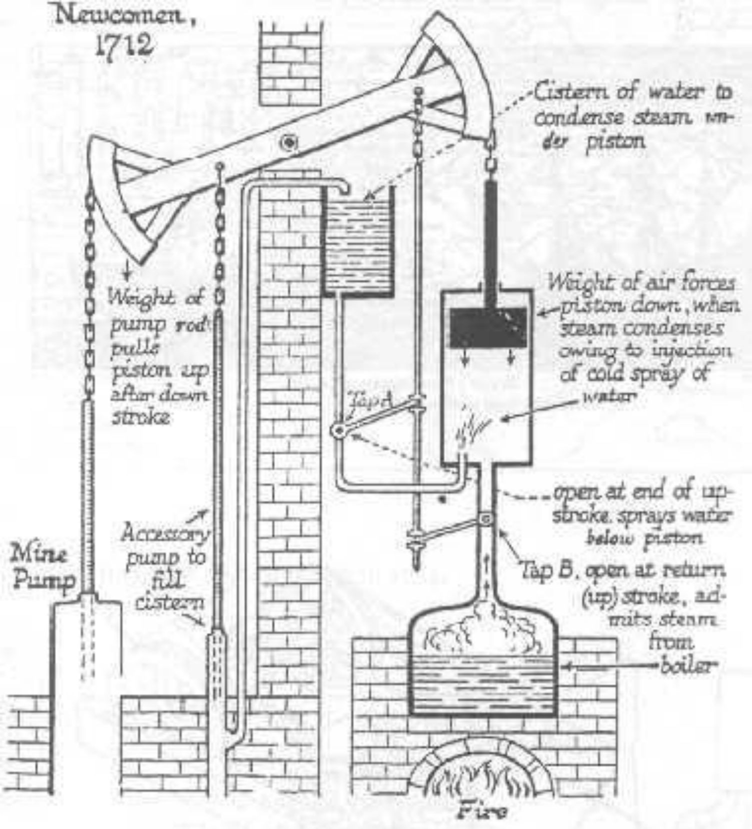
Let's hope that no one has to go through a repeat of anything like **Toyota's gas pedal nightmare**. The company still can't determine whether it is a mechanical or a software problem. For columnist Michael Barr's take on this issue check out his blog: "**Is Toyota's Accelerator Problem Caused by Embedded Software Bugs?**"



I read something about the big Toyota recall being related to floor mats interfering with the accelerator, but I was told that the problem appears to be software (firmware) for the control-by-wire pedal. Me thinks somebody probably forgot to check ranges, overflows, or stability properly when implementing the "algorithm."



But none of the articles I've read have talked about software being a cause. And it's not clear if the affected models are drive-by-wire. However, at least one article I read yesterday suggested that one fix being worked on is a software interlock to ensure that if both the brake and the gas pedal are depressed, the brake will override the accelerator. On the one hand, that seems to mean that software is already in the middle; on the other, I would be extremely surprised to learn that such an interlock wasn't already present in a drive-by-wire system.


Newcomen, 1712
Cistern of water to condense steam under piston
Weight of pump rod pulls piston up after down stroke
Weight of air forces piston down, when steam condenses owing to injection of cold spray of water
Tap A
open at end of upstroke, sprays water below piston
Mine Pump
Accessory pump to fill cistern
Tap B, open at return (up) stroke, admits steam from boiler
Fire

Diagrammatic View of Newcomen's Atmospheric or Fire Engine (1712)

- Atmospheric steam engine
- Used water spray to condense steam in cylinder
- Control of valve based on walking beam position
- 1712, invented first usable steam engine

Components of a Programming language

- Means for expressing and manipulation of data
  - Data types (int, char, double, float)
  - Operators (+,-,*,/,%)
  - Expressions (a+b, a+b*c)
- Methods of Controlling Program Flow
  - Statements and Blocks: functions and subroutines, VI's
  - If-Else ; Else-If
  - Switch
  - Loops – Whiles and For, Do-while
- Input and Output
- Advanced Features
  - Pointers and Arrays
  - Structures

Computer Program :

Simply a collection of instructions for a computer

Computer :

A machine that manipulates data according to a list of instructions

Source : Wikipedia

Sequential Flow Languages:

Almost all traditional text based languages, C, C++ Java…..

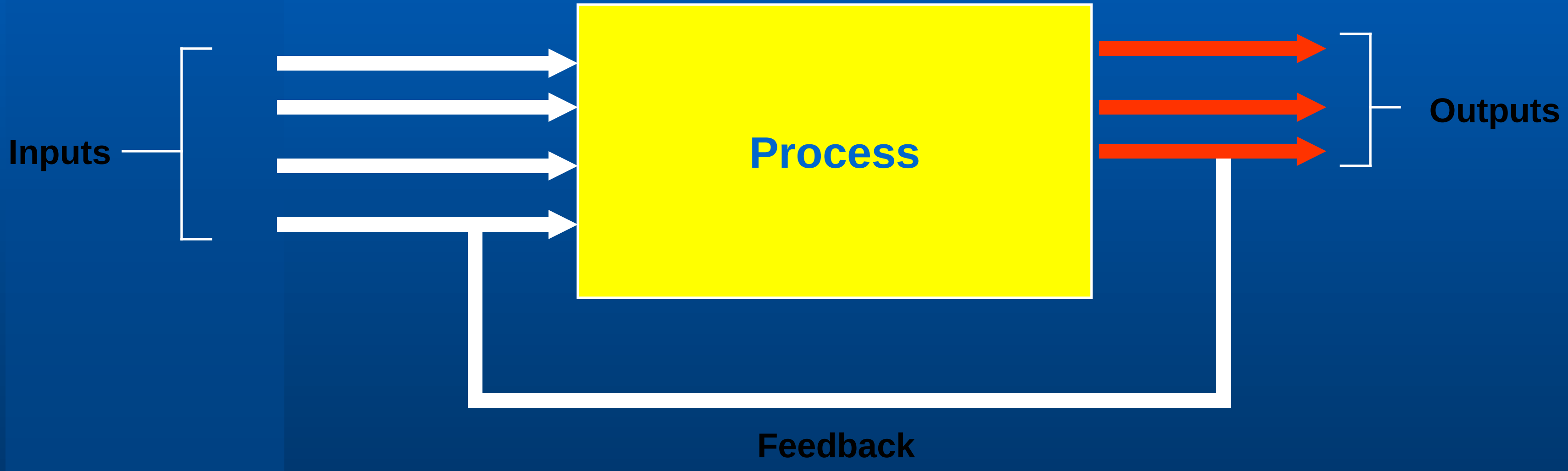
Program flow dictated by the order in which instructions are listed.

Data Flow Languages :

Execution dictated by the readiness of inputs to a set of instructions.

LabVIEW is a data flow language.

Generalized Architecture :



Final Project

- Group Effort (4 members optimal)
- Demonstrate the use of real time software
- Design and development of Host GUI software
- Components running on multiple CPU’s or Cores
- Interaction with the external world through sensors and actuators
- Must be multitasking

Multitasking:

- Human Multitasking – The ability for someone to perform more than one task at a time

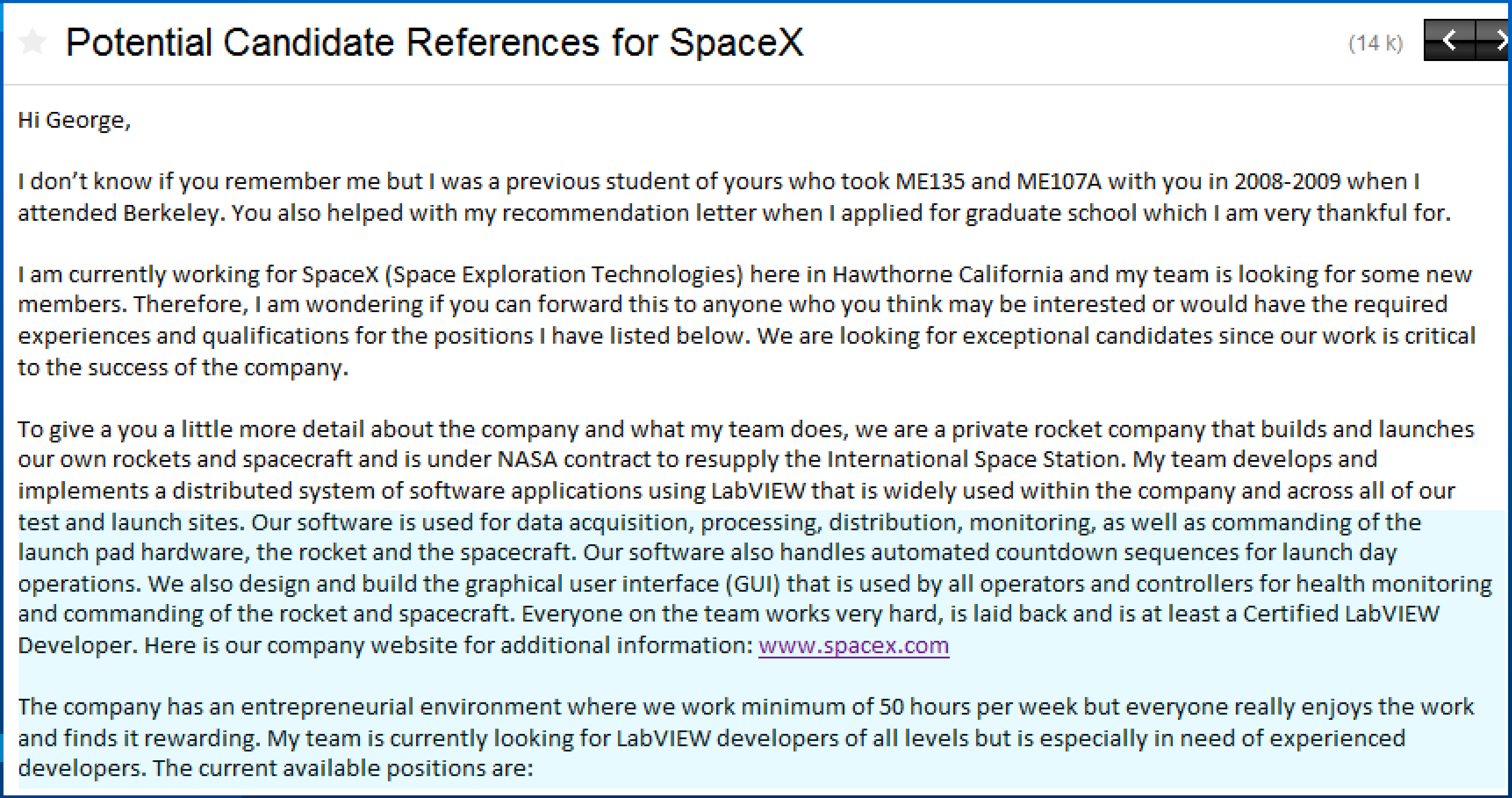
- Computer Multitasking – The apparent simultaneous performance of one or more task by a CPU

Variables Declarations :

- Size of the data
- How to decode the data

int
float
double
long

signed int
unsigned int



Potential Candidate References for SpaceX

Hi George,

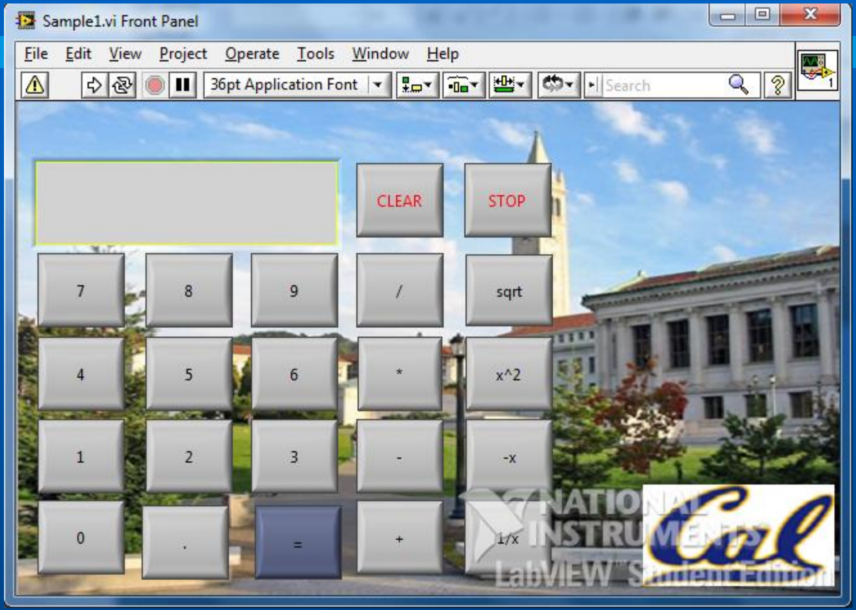
I don't know if you remember me but I was a previous student of yours who took ME135 and ME107A with you in 2008-2009 when I attended Berkeley. You also helped with my recommendation letter when I applied for graduate school which I am very thankful for.

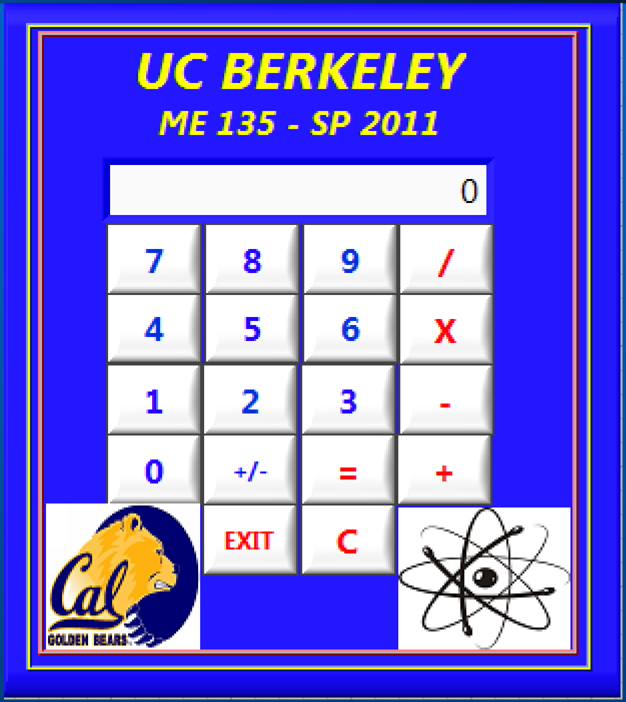
I am currently working for SpaceX (Space Exploration Technologies) here in Hawthorne California and my team is looking for some new members. Therefore, I am wondering if you can forward this to anyone who you think may be interested or would have the required experiences and qualifications for the positions I have listed below. We are looking for exceptional candidates since our work is critical to the success of the company.

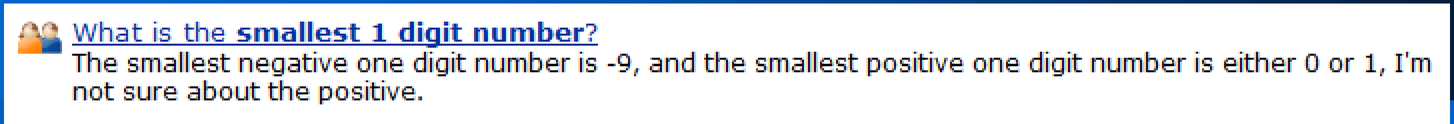
To give a you a little more detail about the company and what my team does, we are a private rocket company that builds and launches our own rockets and spacecraft and is under NASA contract to resupply the International Space Station. My team develops and implements a distributed system of software applications using LabVIEW that is widely used within the company and across all of our test and launch sites. Our software is used for data acquisition, processing, distribution, monitoring, as well as commanding of the launch pad hardware, the rocket and the spacecraft. Our software also handles automated countdown sequences for launch day operations. We also design and build the graphical user interface (GUI) that is used by all operators and controllers for health monitoring and commanding of the rocket and spacecraft. Everyone on the team works very hard, is laid back and is at least a Certified LabVIEW Developer. Here is our company website for additional information: www.spacex.com

The company has an entrepreneurial environment where we work minimum of 50 hours per week but everyone really enjoys the work and finds it rewarding. My team is currently looking for LabVIEW developers of all levels but is especially in need of experienced developers. The current available positions are:

Invoke LabVIEW 2011

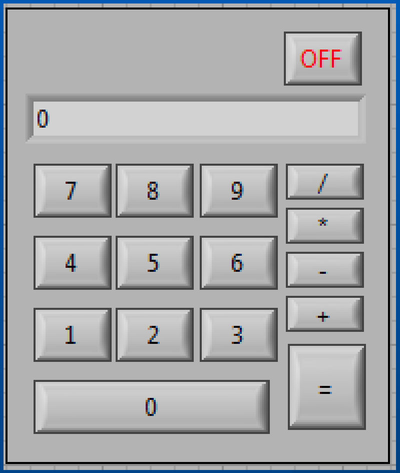

Sample1.vi Front Panel
File Edit View Project Operate Tools Window Help
36pt Application Font
Search
CLEAR
STOP
7
8
9
/
sqrt
4
5
6
*
x^2
1
2
3
-
-x
0
.
=
+
1/x
NATIONAL INSTRUMENTS
LabVIEW Student Edition
Cal

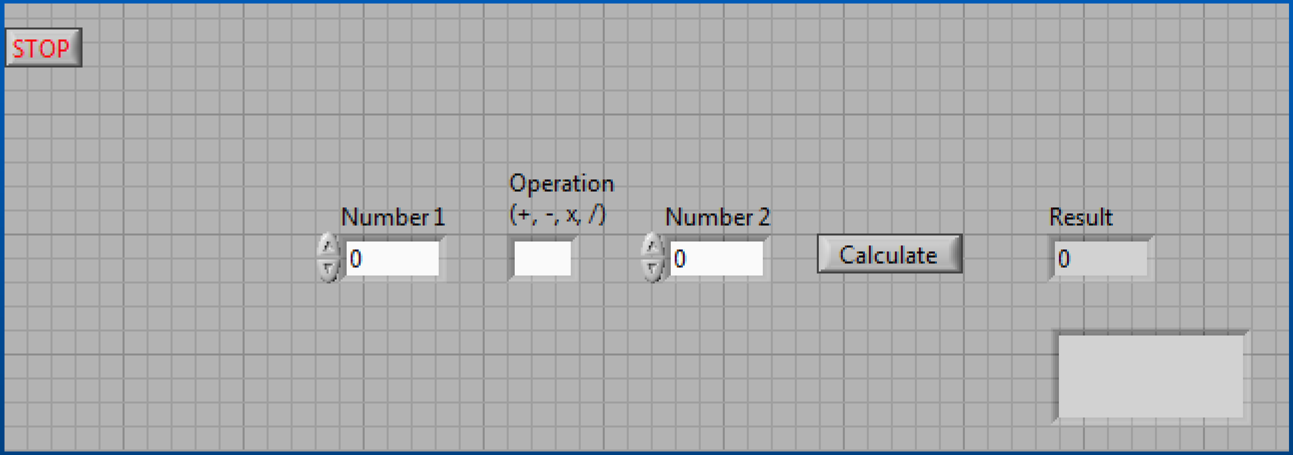

UC BERKELEY
ME 135 - SP 2011
0
7
8
9
/
4
5
6
X
1
2
3
-
0
+/-
=
+
EXIT
C
Cal
GOLDEN BEARS

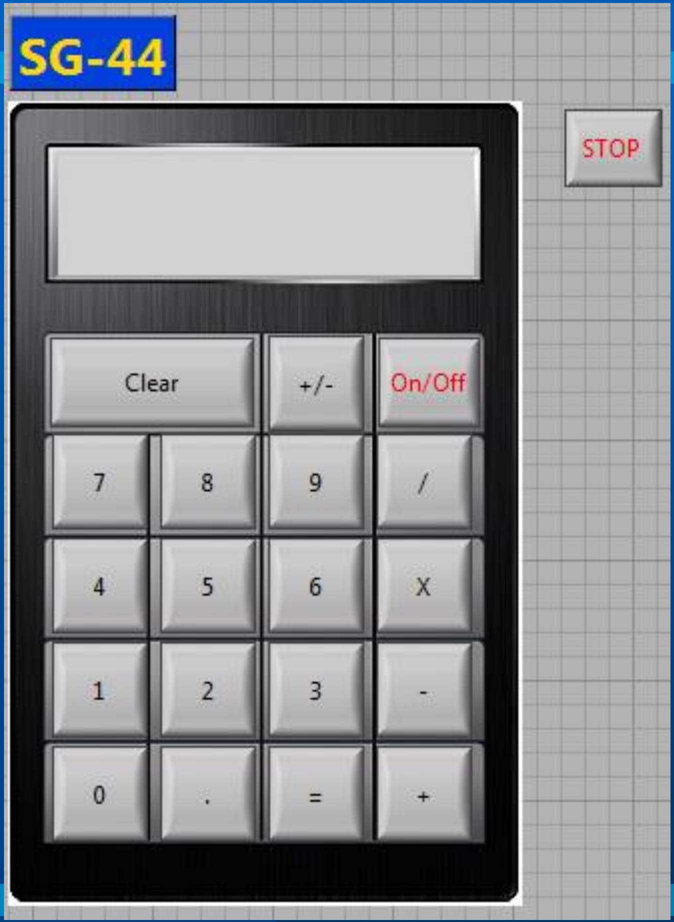


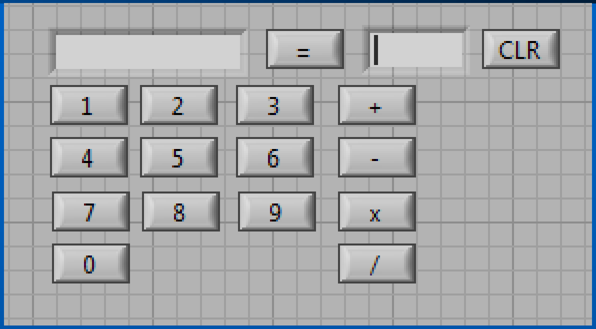
What is the **smallest 1 digit number**?
The smallest negative one digit number is -9, and the smallest positive one digit number is either 0 or 1, I'm not sure about the positive.

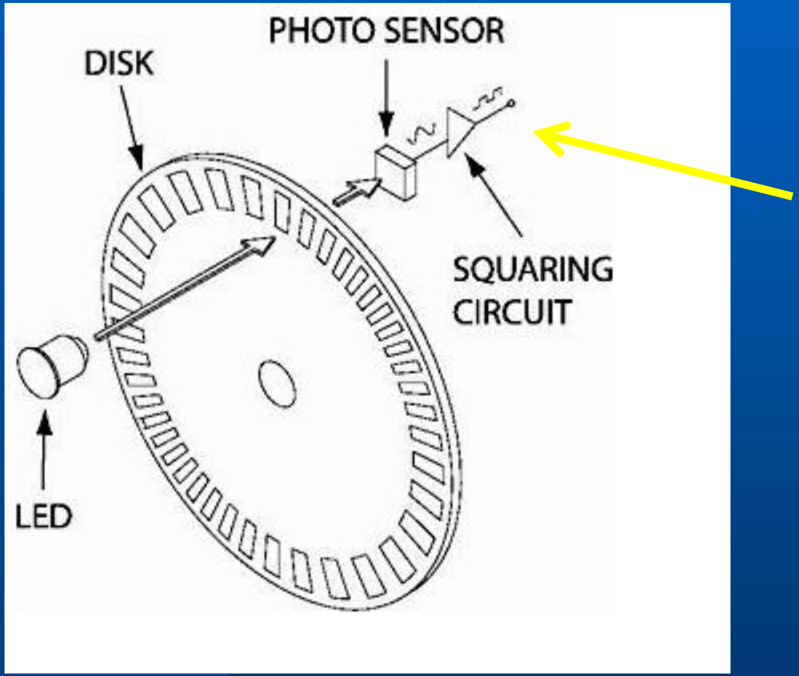
Source: Answer.com






SG-44
STOP
Clear
+/-
On/Off
7
8
9
/
4
5
6
X
1
2
3
-
0
.
=
+

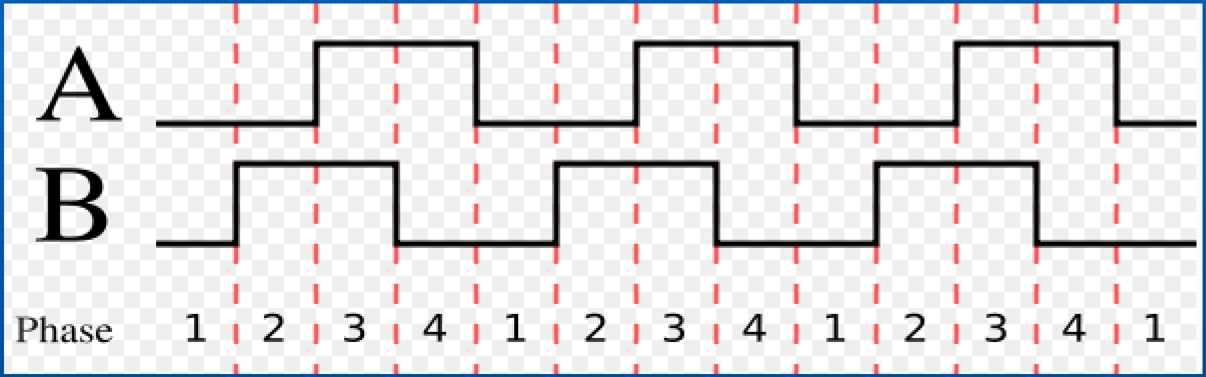

=
CLR
1
2
3
+
4
5
6
-
7
8
9
x
0
/



Single Channel
(Digital Tachometer)

Most Encoders has two channels

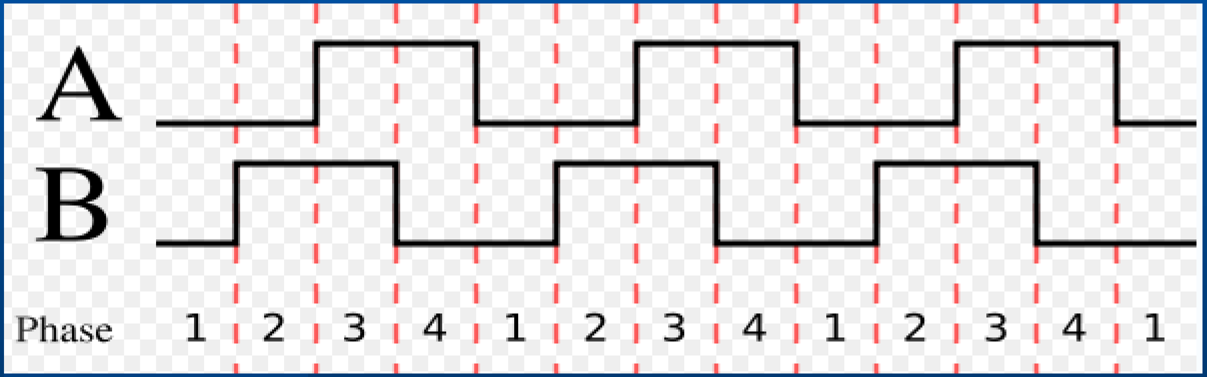
Phase A
Phase B



Quadrature : 4x resolution by counting every edge of the signal

Most common Encoders are considered as digital signals.

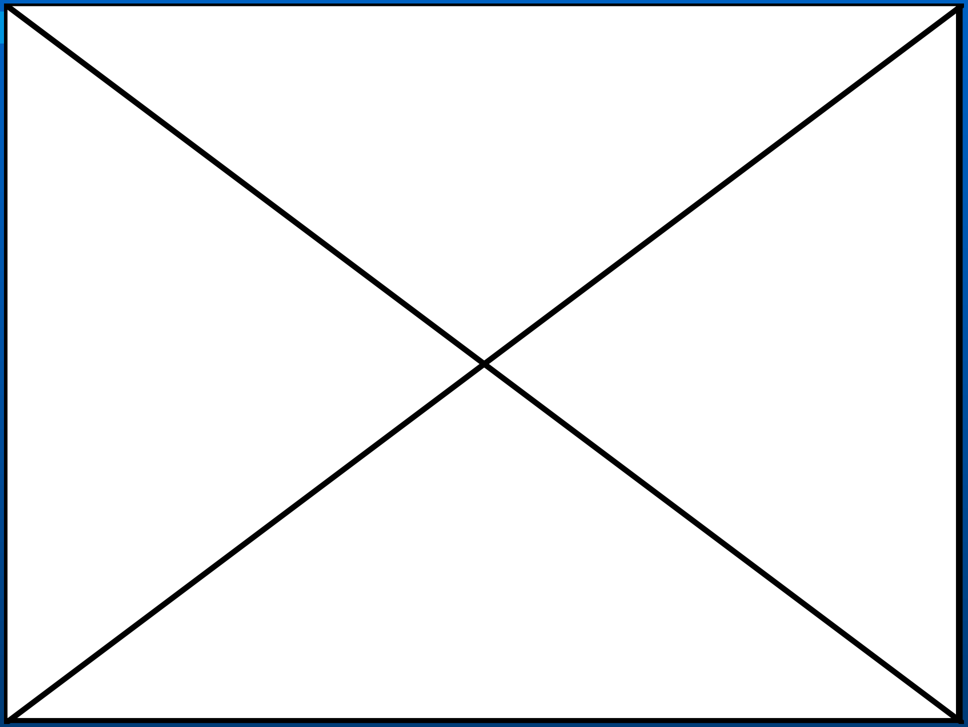
Ideally suitable of microprocessor or microcontroller use.



→ Digital Counter

→ Latched on rising edge of Phase A

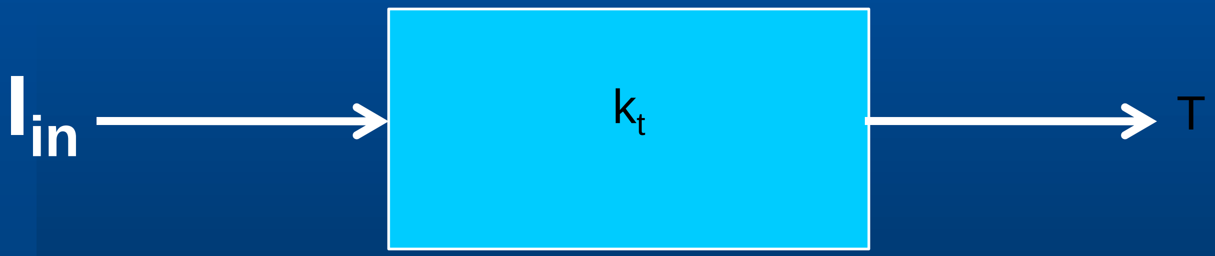
If Phase B is high count up if low count down



Permanent Magnet DC motor

$$T = k_t * I_{in}$$

where k_t = Torque Constant



Model of an ideal inertial load

$$\Sigma T = J*\alpha$$

$$\alpha = d\omega/dt$$

$$\omega = d\theta/dt$$

where J = polar moment of inertia

α = angular acceleration

ω = angular velocity

θ = angular position

$$\Sigma T = J^*\alpha$$
$$\alpha = d\omega/dt$$

$$\Sigma T = J^* d\omega/dt$$
$$\omega = d\theta/dt$$

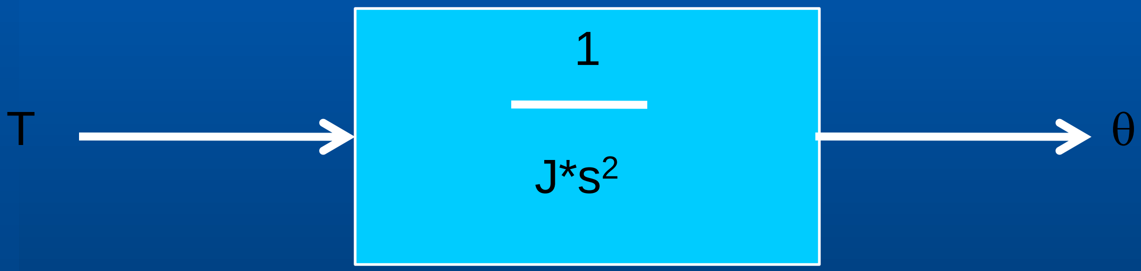
$$\Sigma T = J^* d^2\theta/dt^2$$

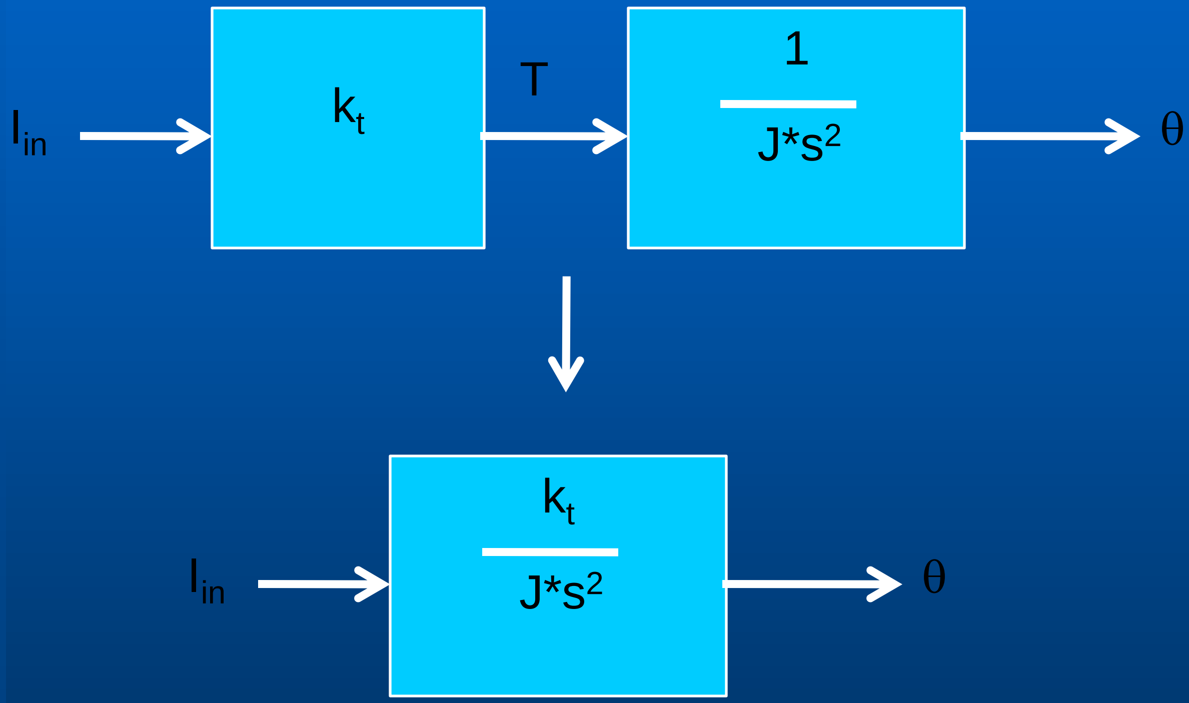
using Laplace Transform:

$$\Sigma T = J^* s^2\theta$$

$\Sigma T = J^* s^2\theta$

$\Sigma T = J^* s^2\theta$




I_{in}
k_t
T
$\frac{1}{J*s^2}$
θ
I_{in}
$\frac{k_t}{J*s^2}$
θ

Timeline: Aim to be done on May 3rd

12 days left
(192 or 288 hours left w/o sleep)
(768 or 1152 FPGA compiles)

Average Group Size: 4 → 768 or 1152 engineering hours

Amount of Shop Time: 64 hours

Realistically based on 4 hours/day: 48 Hrs / group member

Project Grading

What am I looking for :- Perfection

Demo Day is May 4th.
Presentation will start at 10:00 AM until the end…..
Location : Hesse lounge, 120, 122, and 50-A Hesse

There are 38 Projects. So stack your spot early. You do not need to be there all day.

There will be 5 graders:

The 3 GSI's, myself, and a very special guest (Most important person to impress).

What will happen on Demo Day:

Starting at 10:00 AM, the 5 graders will go to each demo, to evaluate your demo.

You will have the opportunity to demonstrate your project in its entirety as you see fit.

Each of us may or may not ask you questions about your implementation.

My focus of course will be the software. You will have to check with each of the other grader in terms of what is their focus.

You will not have an opportunity to know the focus of the Special Guest.

Grade Determination:

Demo and Projects will be ranked by each grader.

The projects will be evaluated on a curve and the major emphasis will be on how the following software concepts were presented:

- GUI
- Real-time
- Multitasking

And

- Presentation
- Report

Keep in mind the evaluation is still on a curve meaning the average grade of the class starts from the assumption that the above components will be done by everyone.

If your hardware is not working by now:

As a backup, it is highly recommended that you think about having a purely software demonstration.

A Producer-Consumer, and State Machine implementations are good examples of software driven test.

Common Motion Trajectory Generation Methods:

- Trapezoidal Trajectory :

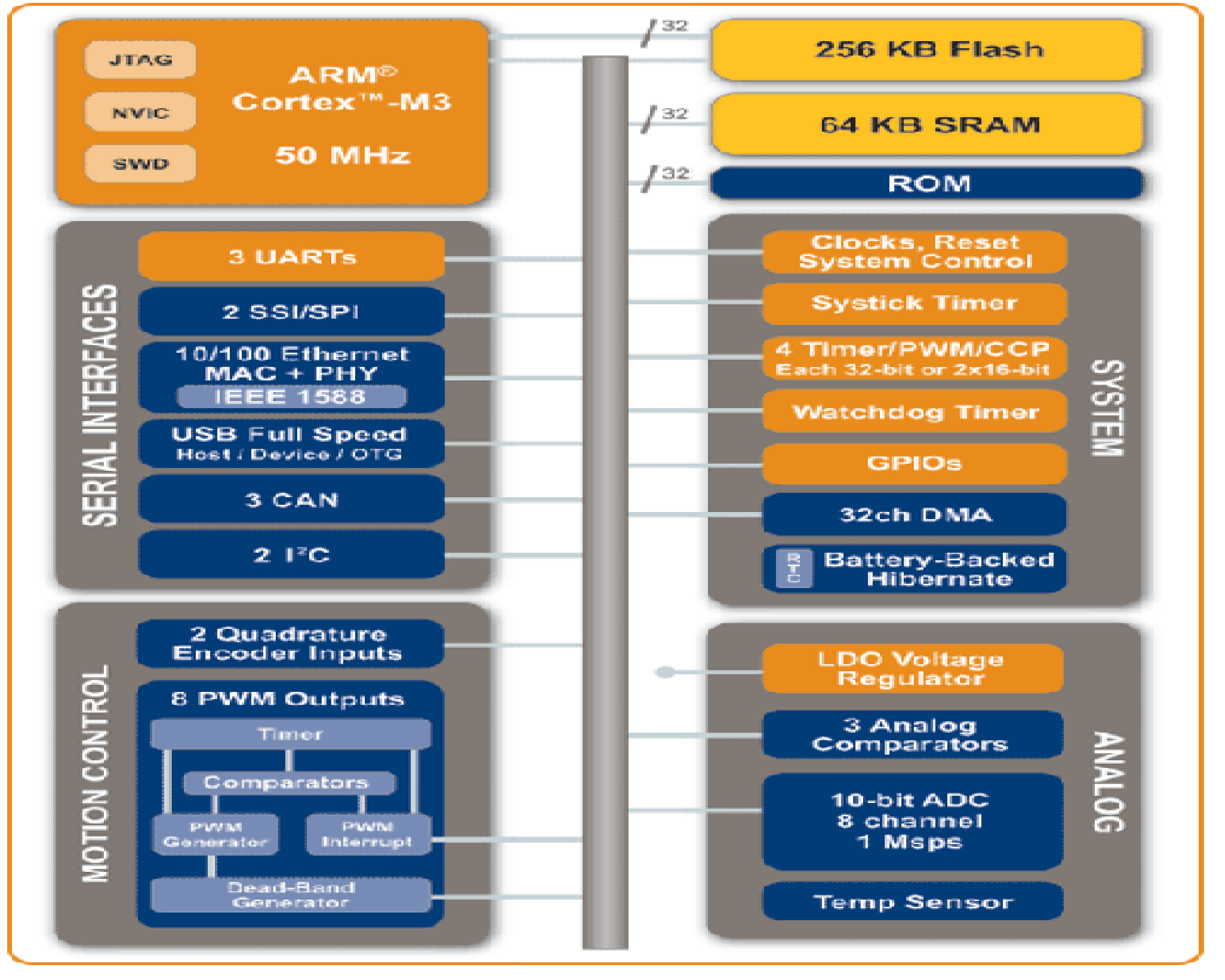
Constant Acceleration

Acc(k) = C;
Vel(k) = Vel(k-1) + Acc(k);
Pos(k) = Pos(k-1) +Vel(k);

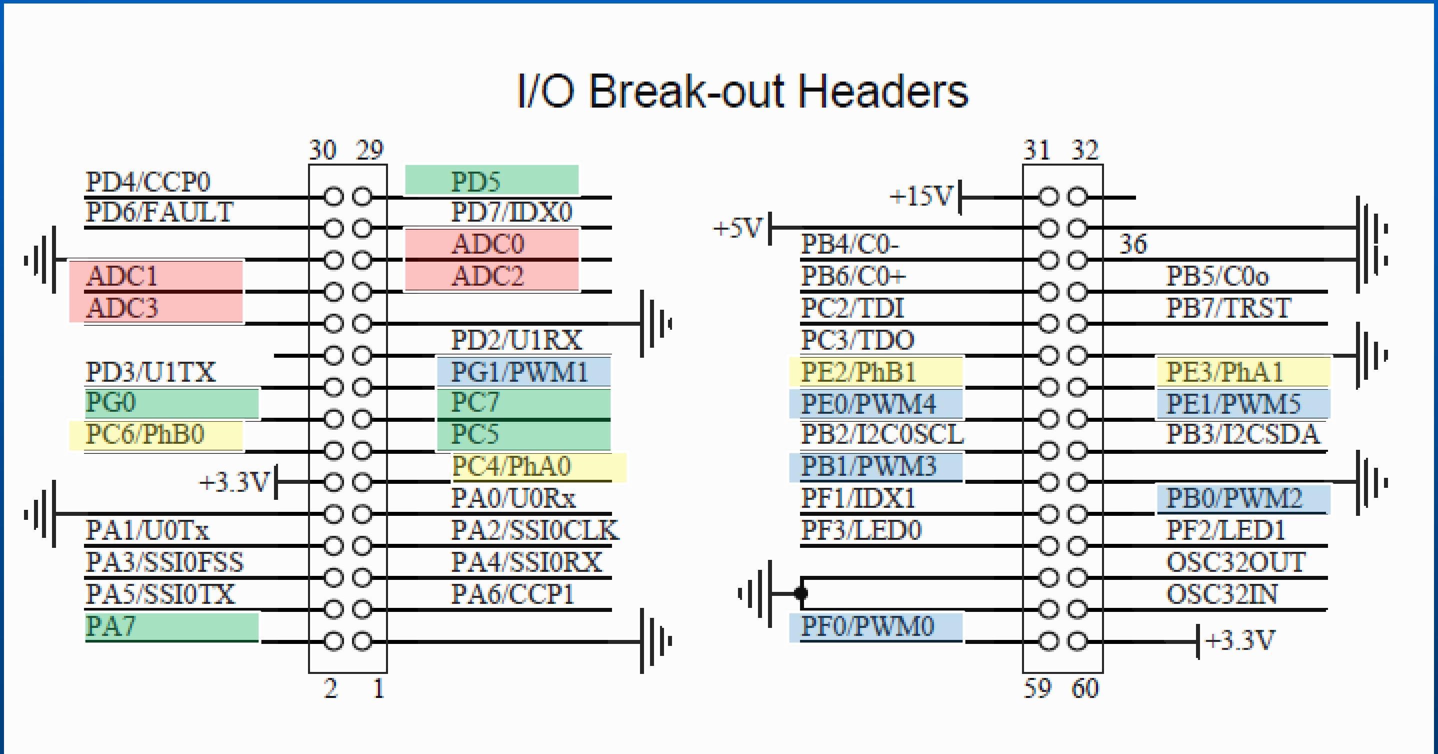
- Decision points:

- $Vel(k) > V_{cruising}$, $Pos_{switch} = Pos(k)$
- $Pos_{switch2} = Pos_{destination} - Pos_{switch}$

Luminary Block Diagram:

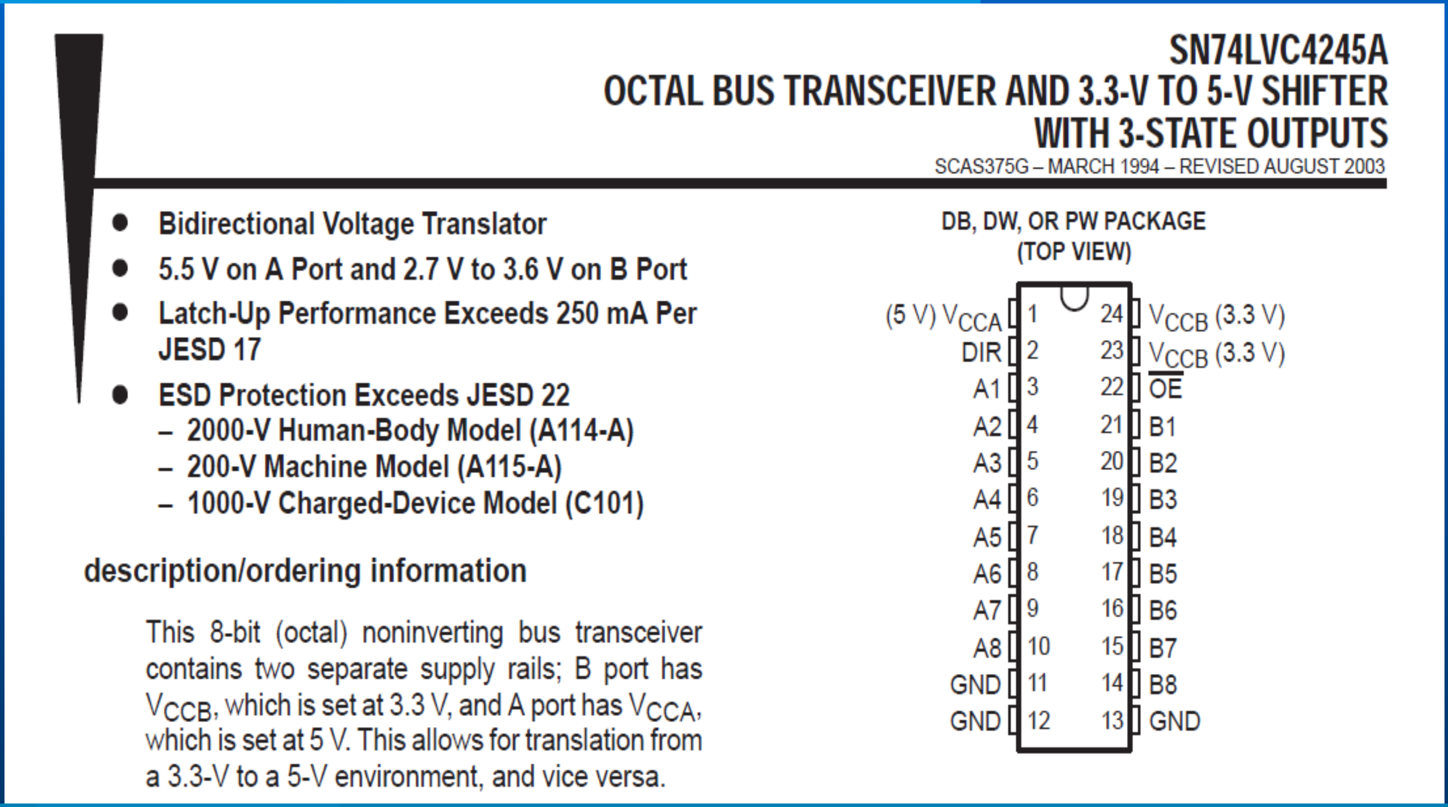


Luminary I/O Pinouts:



I/O Break-out Headers

3.3V to 5V Level Shifter:



SN74LVC4245A
OCTAL BUS TRANSCEIVER AND 3.3-V TO 5-V SHIFTER WITH 3-STATE OUTPUTS

SCAS375G – MARCH 1994 – REVISED AUGUST 2003

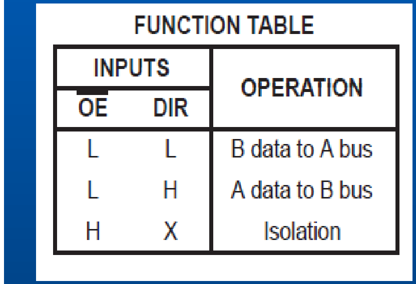
- **Bidirectional Voltage Translator**
- **5.5 V on A Port and 2.7 V to 3.6 V on B Port**
- **Latch-Up Performance Exceeds 250 mA Per JESD 17**
- **ESD Protection Exceeds JESD 22**
  - **2000-V Human-Body Model (A114-A)**
  - **200-V Machine Model (A115-A)**
  - **1000-V Charged-Device Model (C101)**

description/ordering information

This 8-bit (octal) noninverting bus transceiver contains two separate supply rails; B port has V_{CCB}, which is set at 3.3 V, and A port has V_{CCA}, which is set at 5 V. This allows for translation from a 3.3-V to a 5-V environment, and vice versa.

DB, DW, OR PW PACKAGE (TOP VIEW)

(5 V) V_{CCA}	1	24	V_{CCB} (3.3 V)
DIR	2	23	V_{CCB} (3.3 V)
A1	3	22	$\overline{OE}$
A2	4	21	B1
A3	5	20	B2
A4	6	19	B3
A5	7	18	B4
A6	8	17	B5
A7	9	16	B6
A8	10	15	B7
GND	11	14	B8
GND	12	13	GND



FUNCTION TABLE

INPUTS		OPERATION
$\overline{OE}$	DIR	
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

Luminary I/O:

General Purpose I/O:

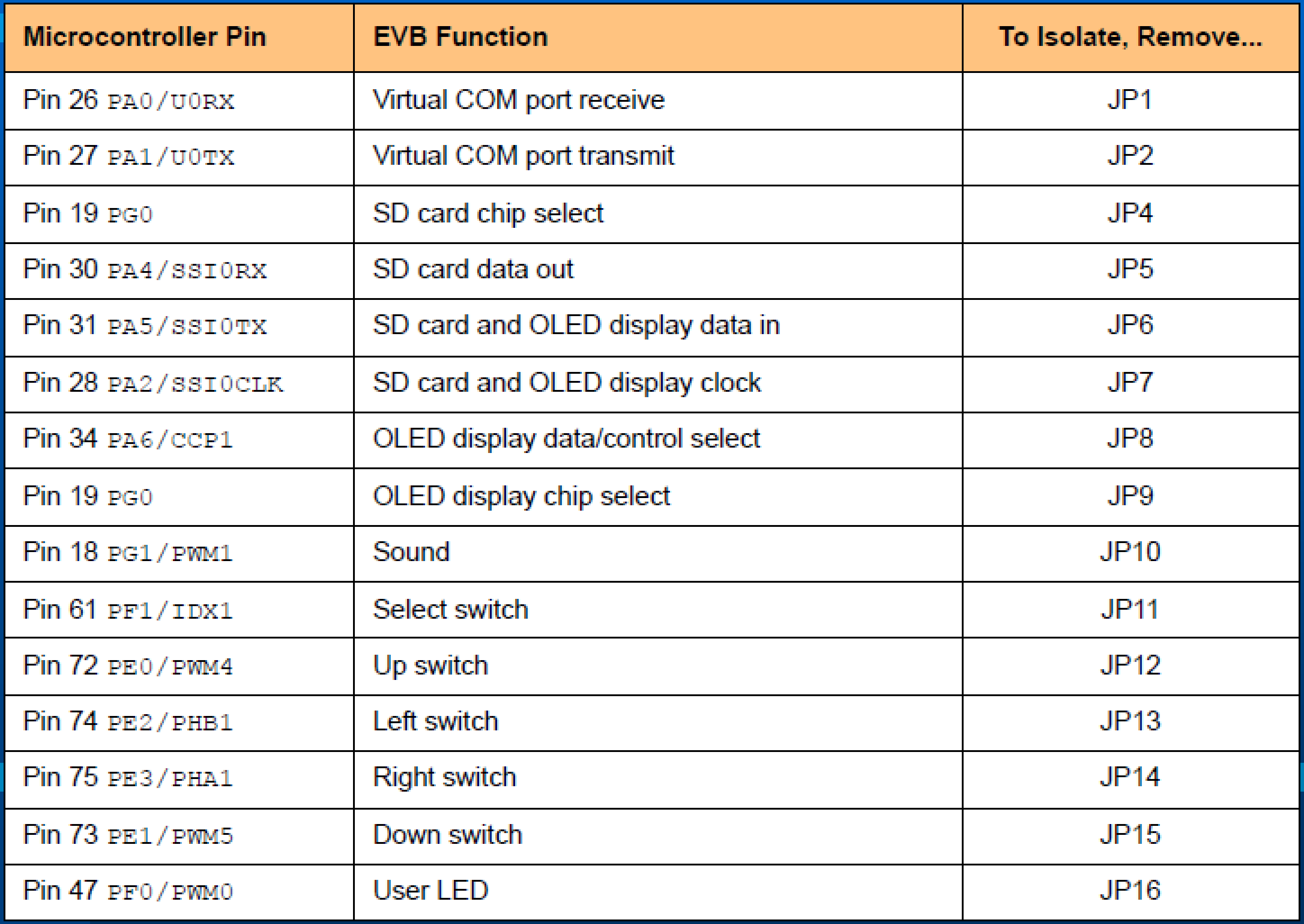
42 Input-Output Lines over 5 ports
Some lines are shared

PA0-PA7
PB0-PB7
PC0-PC7
PD0-PD7
PE0-PE3
PF0-PF3
PG0-PG1

Totally Free Lines:

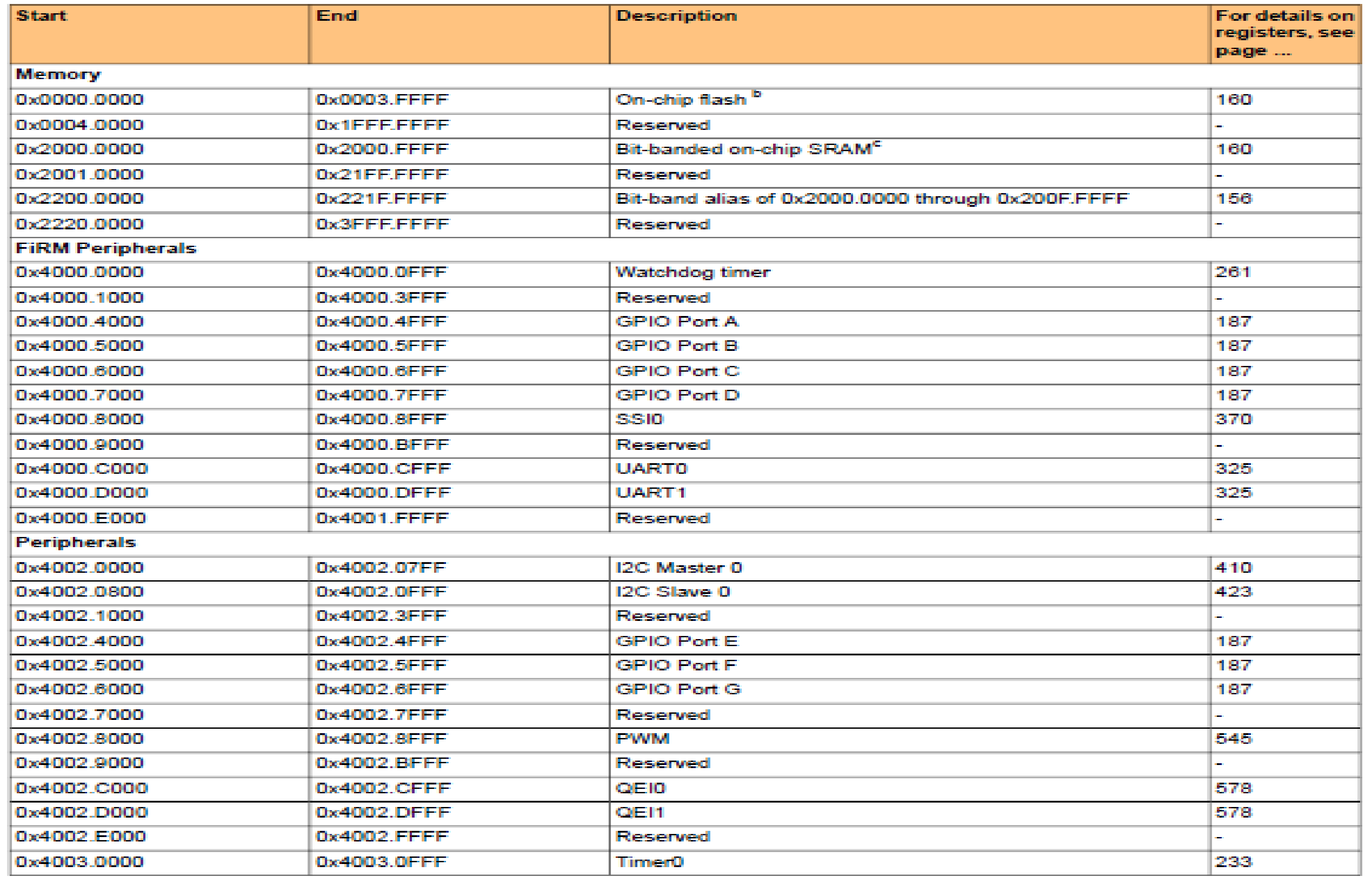
PA7
PC5
PC7
PD5
PG0

Luminary On Board Hardware :



Microcontroller Pin	EVB Function	To Isolate, Remove...
Pin 26 PA0/U0RX	Virtual COM port receive	JP1
Pin 27 PA1/U0TX	Virtual COM port transmit	JP2
Pin 19 PG0	SD card chip select	JP4
Pin 30 PA4/SSI0RX	SD card data out	JP5
Pin 31 PA5/SSI0TX	SD card and OLED display data in	JP6
Pin 28 PA2/SSI0CLK	SD card and OLED display clock	JP7
Pin 34 PA6/CCP1	OLED display data/control select	JP8
Pin 19 PG0	OLED display chip select	JP9
Pin 18 PG1/PWM1	Sound	JP10
Pin 61 PF1/IDX1	Select switch	JP11
Pin 72 PE0/PWM4	Up switch	JP12
Pin 74 PE2/PHB1	Left switch	JP13
Pin 75 PE3/PHA1	Right switch	JP14
Pin 73 PE1/PWM5	Down switch	JP15
Pin 47 PF0/PWM0	User LED	JP16

Luminary Memory Map:

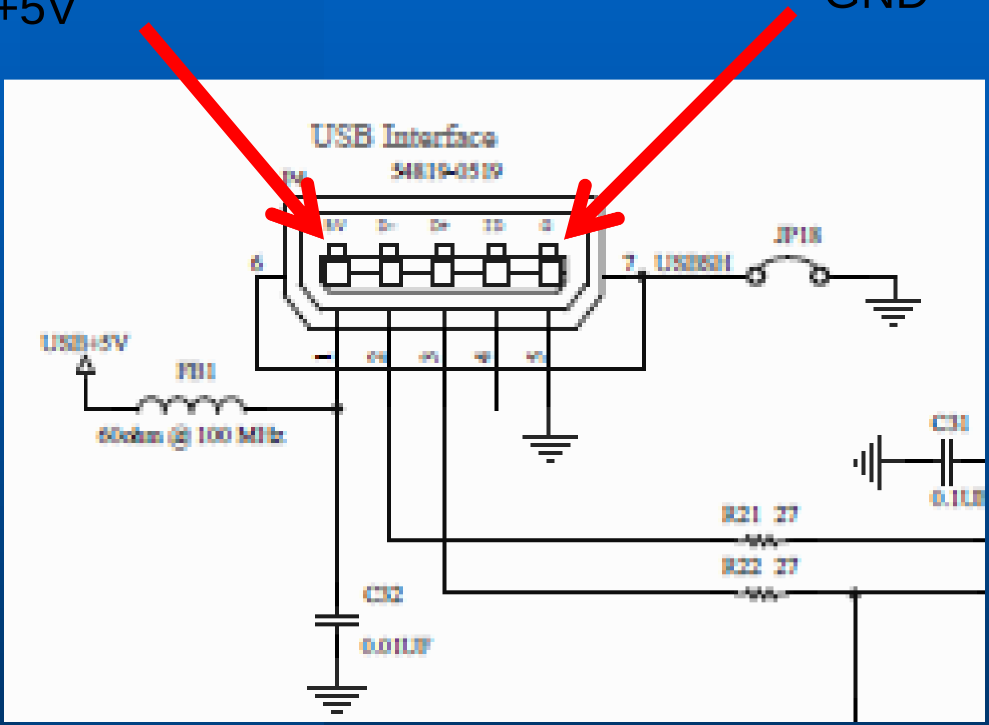


Start	End	Description	For details on registers, see page ...
Memory			
0x0000.0000	0x0003.FFFF	On-chip flash [b]	160
0x0004.0000	0x1FFF.FFFF	Reserved	-
0x2000.0000	0x2000.FFFF	Bit-banded on-chip SRAM[c]	160
0x2001.0000	0x21FF.FFFF	Reserved	-
0x2200.0000	0x221F.FFFF	Bit-band alias of 0x2000.0000 through 0x200F.FFFF	156
0x2220.0000	0x3FFF.FFFF	Reserved	-
FiRM Peripherals			
0x4000.0000	0x4000.0FFF	Watchdog timer	261
0x4000.1000	0x4000.3FFF	Reserved	-
0x4000.4000	0x4000.4FFF	GPIO Port A	187
0x4000.5000	0x4000.5FFF	GPIO Port B	187
0x4000.6000	0x4000.6FFF	GPIO Port C	187
0x4000.7000	0x4000.7FFF	GPIO Port D	187
0x4000.8000	0x4000.8FFF	SSI0	370
0x4000.9000	0x4000.BFFF	Reserved	-
0x4000.C000	0x4000.CFFF	UART0	325
0x4000.D000	0x4000.DFFF	UART1	325
0x4000.E000	0x4001.FFFF	Reserved	-
Peripherals			
0x4002.0000	0x4002.07FF	I2C Master 0	410
0x4002.0800	0x4002.0FFF	I2C Slave 0	423
0x4002.1000	0x4002.3FFF	Reserved	-
0x4002.4000	0x4002.4FFF	GPIO Port E	187
0x4002.5000	0x4002.5FFF	GPIO Port F	187
0x4002.6000	0x4002.6FFF	GPIO Port G	187
0x4002.7000	0x4002.7FFF	Reserved	-
0x4002.8000	0x4002.8FFF	PWM	545
0x4002.9000	0x4002.BFFF	Reserved	-
0x4002.C000	0x4002.CFFF	QEI0	578
0x4002.D000	0x4002.DFFF	QEI1	578
0x4002.E000	0x4002.FFFF	Reserved	-
0x4003.0000	0x4003.0FFF	Timer0	233

Luminary USB Power Input:

+5V

GND



4 pin mini USB Jack

Pin	Name	Cable color	Description
1	VCC	Red	+5 VDC
2	D-	White	Data -
3	D+	Green	Data +
4	GND	Black	Ground

Embedded Software Architecture:

Single Task Operation:

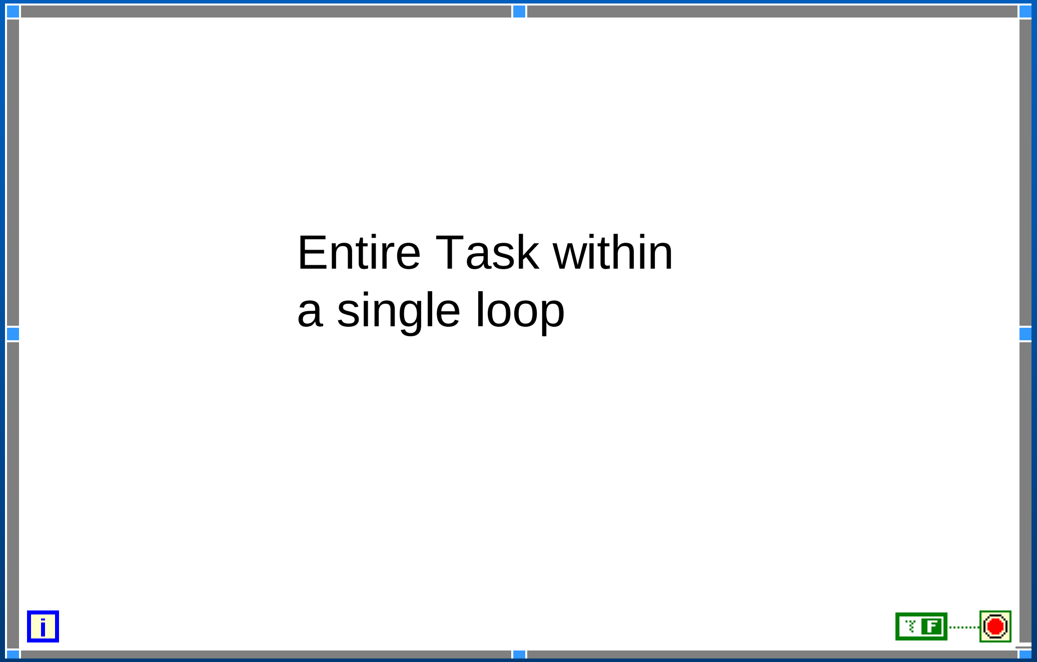
1. Polling Loop
2. Interrupt Driven

MultiTask Operation:

Multiple Single Task Operations

1. Round-Robin
2. Pre-emptive Multitasking

Polling Loop:

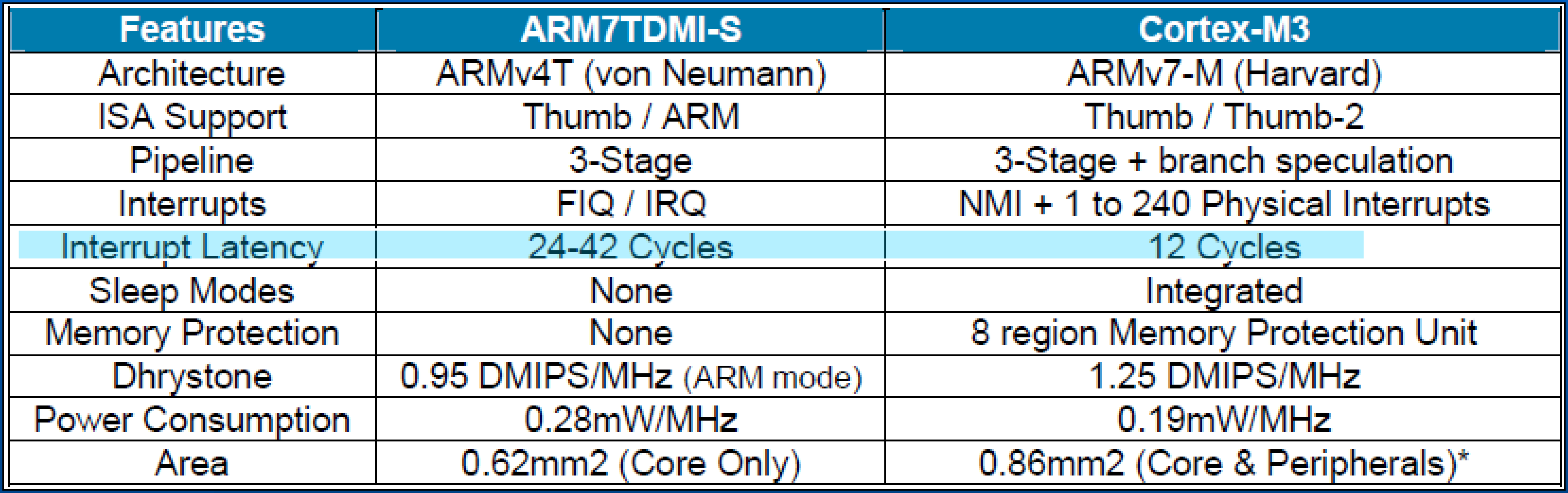


Entire Task within a single loop

Interrupts:

an **interrupt** is an asynchronous signal from hardware indicating the need for attention or a synchronous event in software indicating the need for a change in execution

Luminary Interrupt Latency:



Features	ARM7TDMI-S	Cortex-M3
Architecture	ARMv4T (von Neumann)	ARMv7-M (Harvard)
ISA Support	Thumb / ARM	Thumb / Thumb-2
Pipeline	3-Stage	3-Stage + branch speculation
Interrupts	FIQ / IRQ	NMI + 1 to 240 Physical Interrupts
Interrupt Latency	24-42 Cycles	12 Cycles
Sleep Modes	None	Integrated
Memory Protection	None	8 region Memory Protection Unit
Dhrystone	0.95 DMIPS/MHz (ARM mode)	1.25 DMIPS/MHz
Power Consumption	0.28mW/MHz	0.19mW/MHz
Area	0.62mm2 (Core Only)	0.86mm2 (Core & Peripherals)*

Multitasking:

multitasking is a method by which multiple tasks, also known as processes, share common processing resources such as a CPU.

Common BUGs

- Assignment instead of equality test
- Divide by zero
- NULL pointer dereference
- Infinite loops
- Arithmetic overflow or underflow
- Using an uninitialized variable
- Buffer overflow
- Off by one error
- Race condition
- Loss of precision in type conversion

Final Report : End of the day May 7th

- Background and Motivation
- Technical Challenges prior to proposal
- Actual technical challenges
- Now that you know what you know---

What would you do different if you had access to a time machine

- Your code
- Journal if any

Demonstration Day : May 6th 1-5 pm Hesse Hall

- Your time to shine
- Plan on where you would like to set up
- Use the benches as booth in a trade show
- Representatives from National Instruments will be present
- Faculty and Staff may also visit

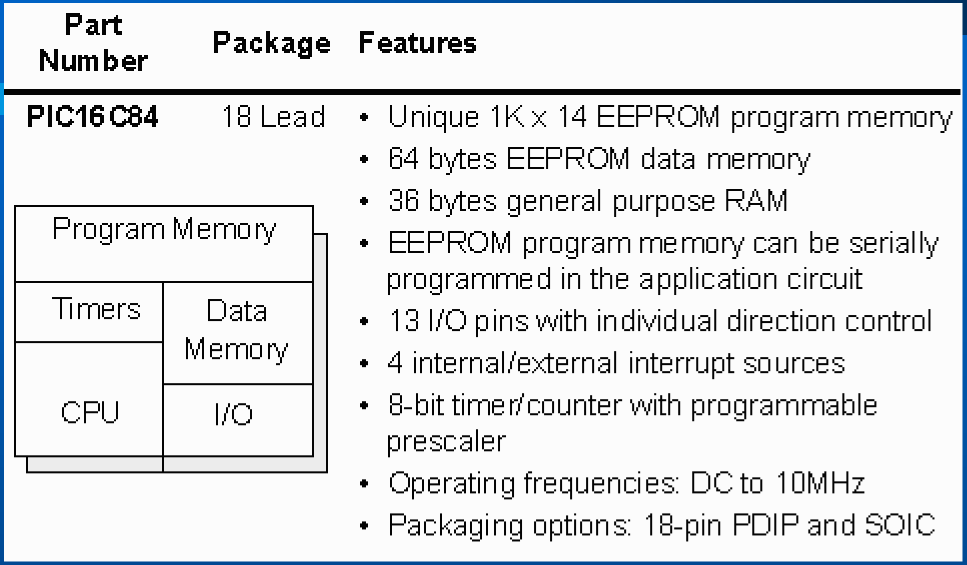
Thank you all for all your hard work up to now and we look forward to your demo next week.

Von-Neumann Architecture

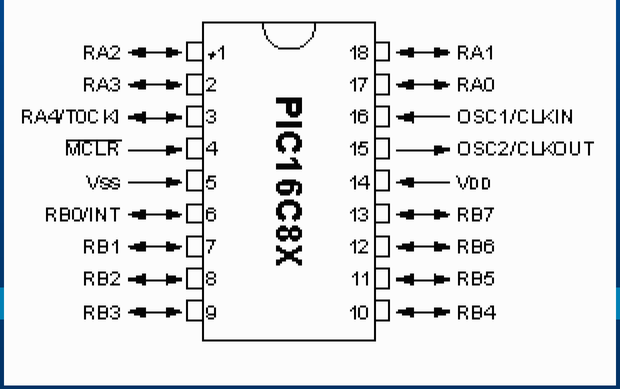
- Most common processor architecture
- Data and Program memory share the same space
- Single data path to CPU

Harvard Architecture

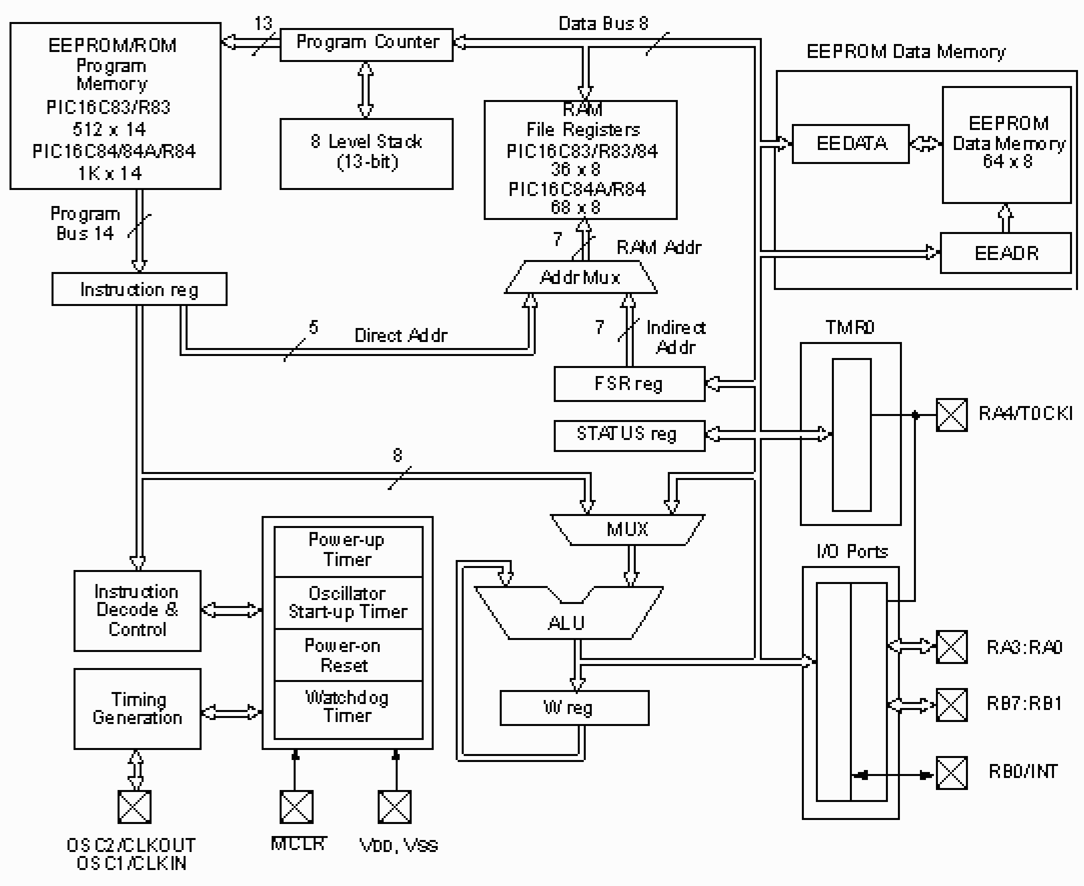
- Most common DSP architecture
- Data and Program have separate memory banks
- Separate path for data and instruction
- Parallel access to memory



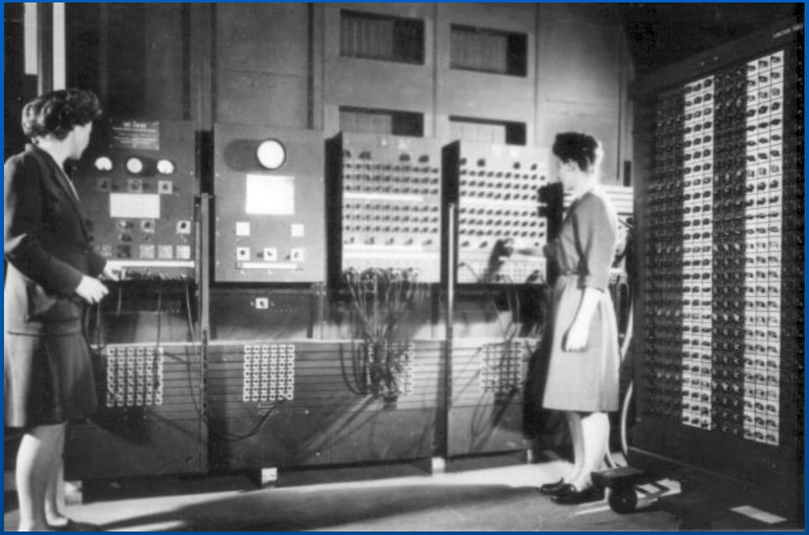
Part Number	Package	Features
PIC16C84	18 Lead	• Unique 1K x 14 EEPROM program memory • 64 bytes EEPROM data memory • 36 bytes general purpose RAM • EEPROM program memory can be serially programmed in the application circuit • 13 I/O pins with individual direction control • 4 internal/external interrupt sources • 8-bit timer/counter with programmable prescaler • Operating frequencies: DC to 10MHz • Packaging options: 18-pin PDIP and SOIC



PIC16C8x Block Diagram:

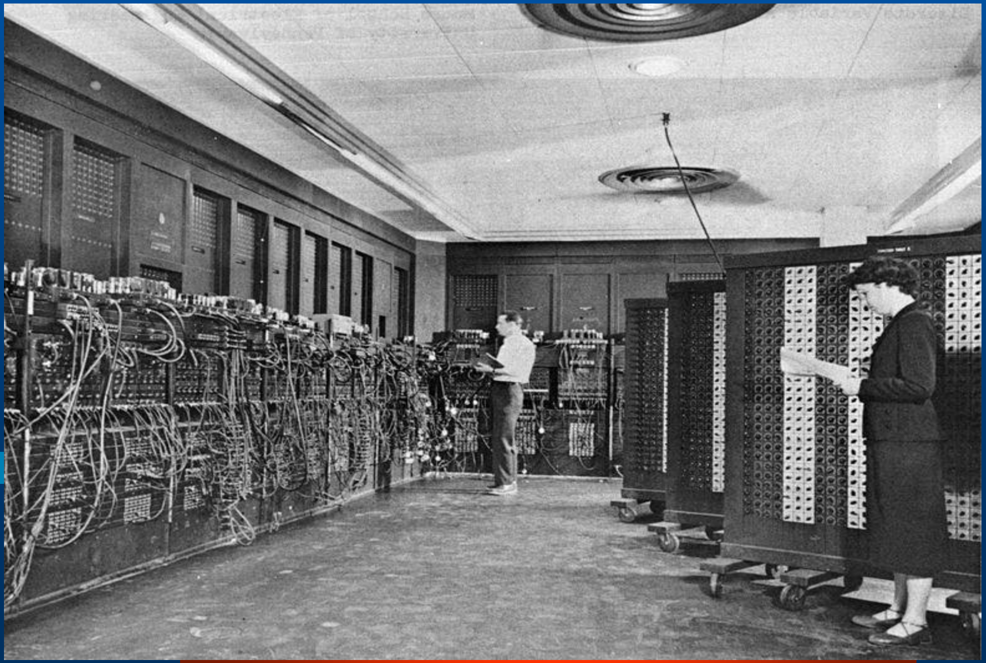


The First Electronic Computer



ENIAC :
Electronic Numerical Integrator
And Computer

Complete in 1946 to calculate
Artillery Firing Table

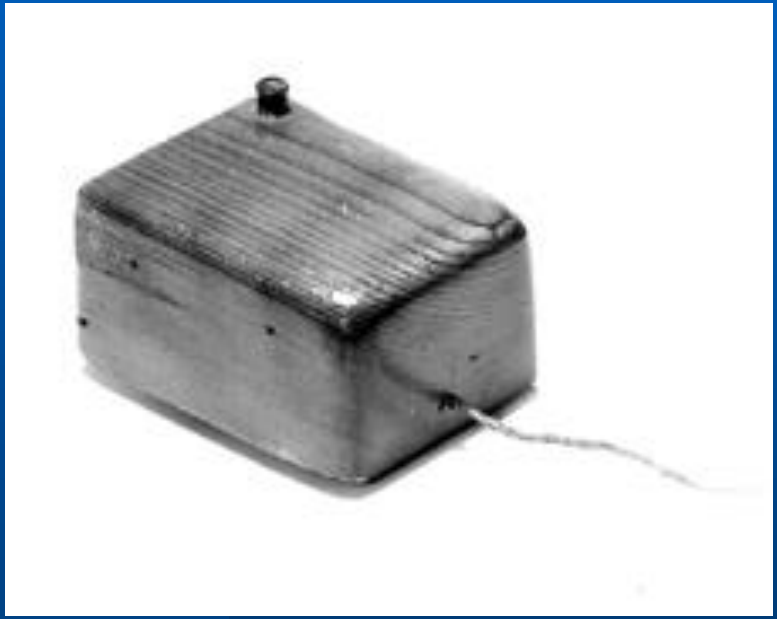


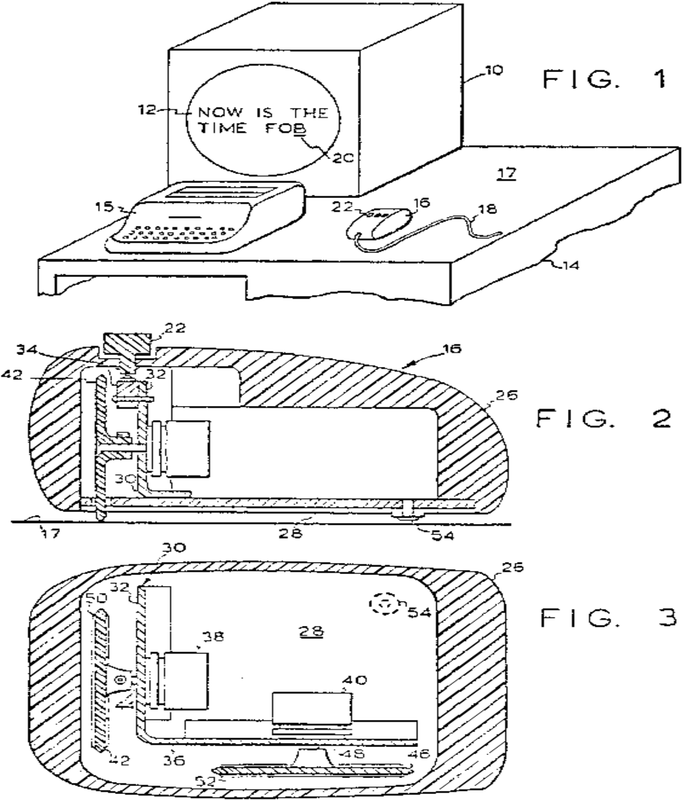
ENIAC

- **17,468 Vacuum Tubes**
- **7,200 Crystal Diodes**
- **1,500 Relays**
- **70,000 Resistors**
- **10,000 Capacitors**
- **5,000,000 Hand-solder joints**
- **Weight** 27 Tons
- **Size** 8.5 Ft by 3.0 Ft by 80 Ft
- **Power** 150 Kw

Obviously not a laptop

The First Mouse (1964)





Source: Picture courtesy of SRI International

The First Bug

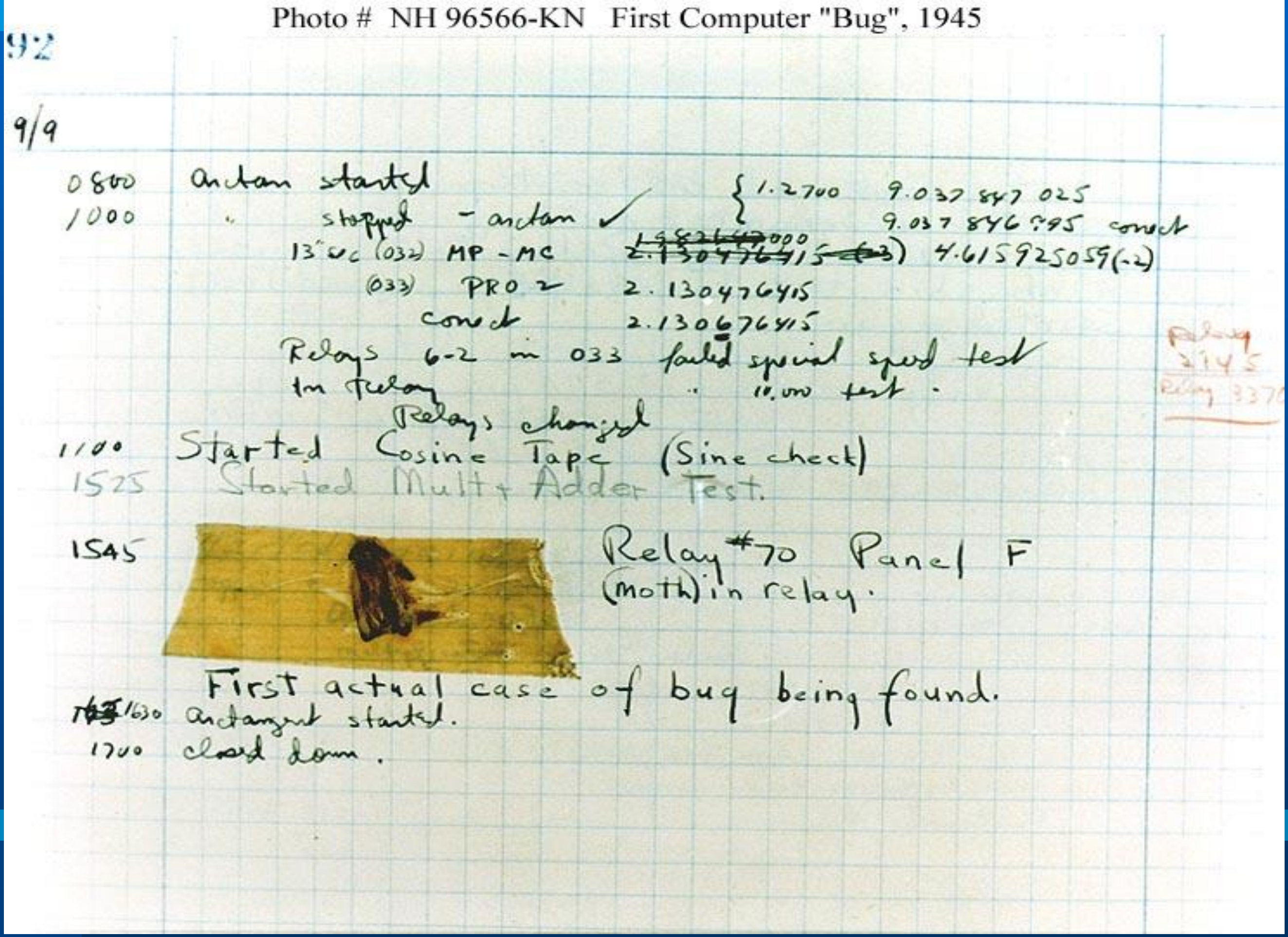


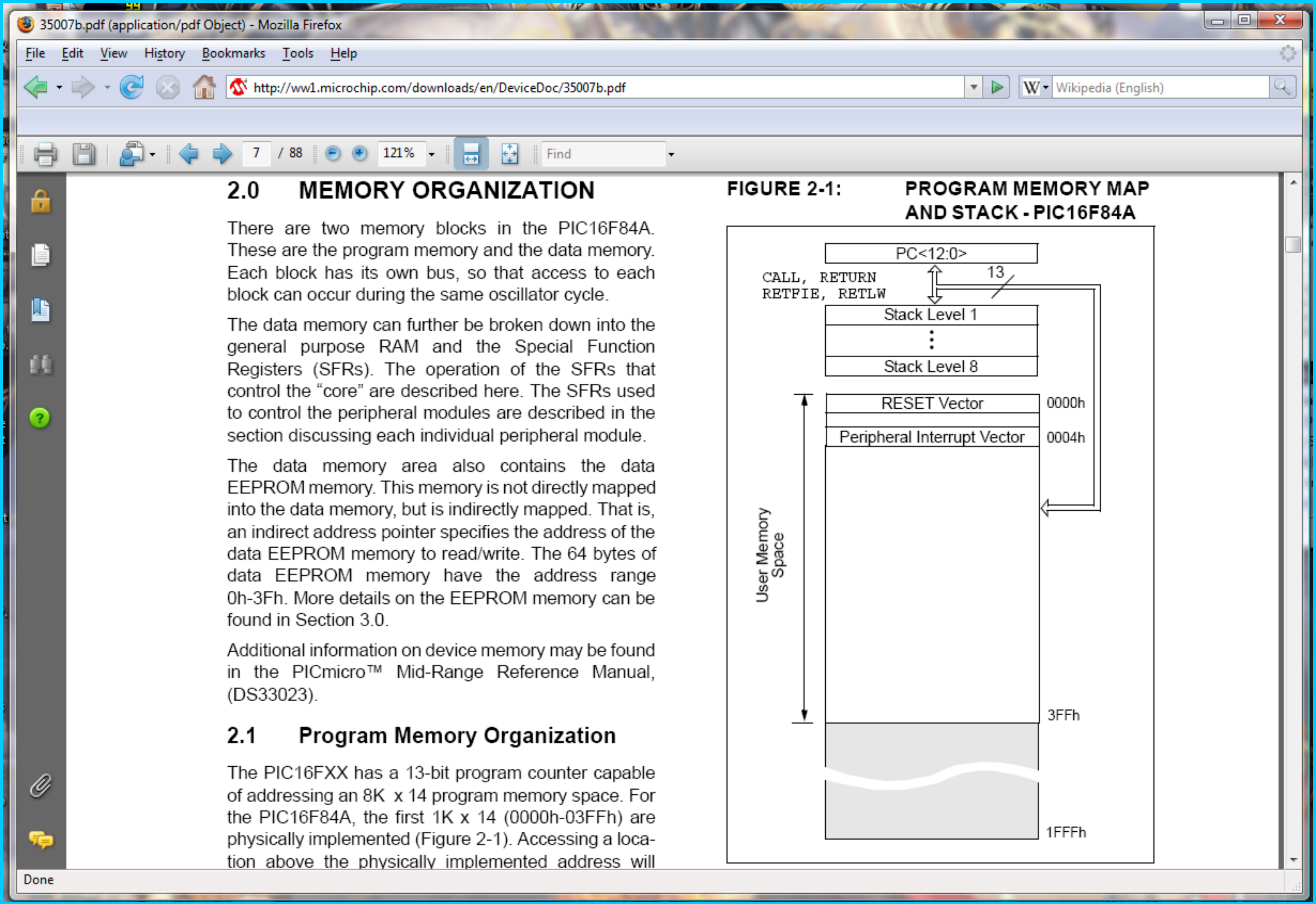
Photo # NH 96566-KN First Computer "Bug", 1945

92

9/9

0800 arctan started
1000 " stopped - arctan ✓ {1.2700 9.037 847 025
9.037 846 995 correct
13" uc (032) MP - MC ~~1.982647000~~ ~~2.130476415 (-3)~~ 4.615925059(-2)
(033) PRO 2 2.130476415
correct 2.130676415
Relays 6-2 in 033 failed special speed test
In relay " 10,000 test.
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult + Adder Test.
1545 Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
1630 arctangent started.
1700 closed down.

Relay 2145
Relay 3370



35007b.pdf (application/pdf Object) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://ww1.microchip.com/downloads/en/DeviceDoc/35007b.pdf
Wikipedia (English)
7 / 88
121%
Find
Done

2.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC16F84A. These are the program memory and the data memory. Each block has its own bus, so that access to each block can occur during the same oscillator cycle.

The data memory can further be broken down into the general purpose RAM and the Special Function Registers (SFRs). The operation of the SFRs that control the "core" are described here. The SFRs used to control the peripheral modules are described in the section discussing each individual peripheral module.

The data memory area also contains the data EEPROM memory. This memory is not directly mapped into the data memory, but is indirectly mapped. That is, an indirect address pointer specifies the address of the data EEPROM memory to read/write. The 64 bytes of data EEPROM memory have the address range 0h-3Fh. More details on the EEPROM memory can be found in Section 3.0.

Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

2.1 Program Memory Organization

The PIC16FXX has a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16F84A, the first 1K x 14 (0000h-03FFh) are physically implemented (Figure 2-1). Accessing a location above the physically implemented address will

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK - PIC16F84A

Task

- A set of program instructions define to complete a given process

Examples of Tasks

- GUI
- Data Acquisition
- Controls
- Trajectory Generation

A Real-time operating System (RTOS) :

- An RTOS facilitates the creation of a real-time system, but does not guarantee the final result will be real-time.

- An RTOS does not necessarily have high throughput

Timeline: Aim to be done on May 7th

A little over 5 weeks left until May 9th Demo Day 10 AM - 2 PM
120 Hesse Hall
(864 hours left w/o sleep)
(849 hours left w/o sleep for project: 15 hours for lecture)

Average Group Size: 4 ➔ 3396 engineering hours

Amount of Shop Time: 208 hours

Realistically based on 4 hours/day: 144 hours / group member
(288 FPGA compiles)

Office and Lab Hours:

Week of 4/1: OH 8:30 – 11:00 PM W-F
LH 8 AM – 5 PM, 8:30 – 11:00 PM
Week of 4/8: OH 9:30 – 11:00 AM TU-TH
LH 8 AM – 5 PM
Week of 4/15: OH 8:30 – 11:00 PM M-F
LH 8 AM – 5 PM, 8:30 – 11:00 PM
Week of 4/22: OH 9:30 – 11:00 AM TU-TH
LH 8 AM – 5 PM
Week of 4/29: OH 8:30 – 11:00 PM M-F
LH 8 AM – 5 PM, 8:30 – 11:00 PM
Week of 5/6: OH 9:30 – 11:00 AM TU
LH 8 AM – 5 PM

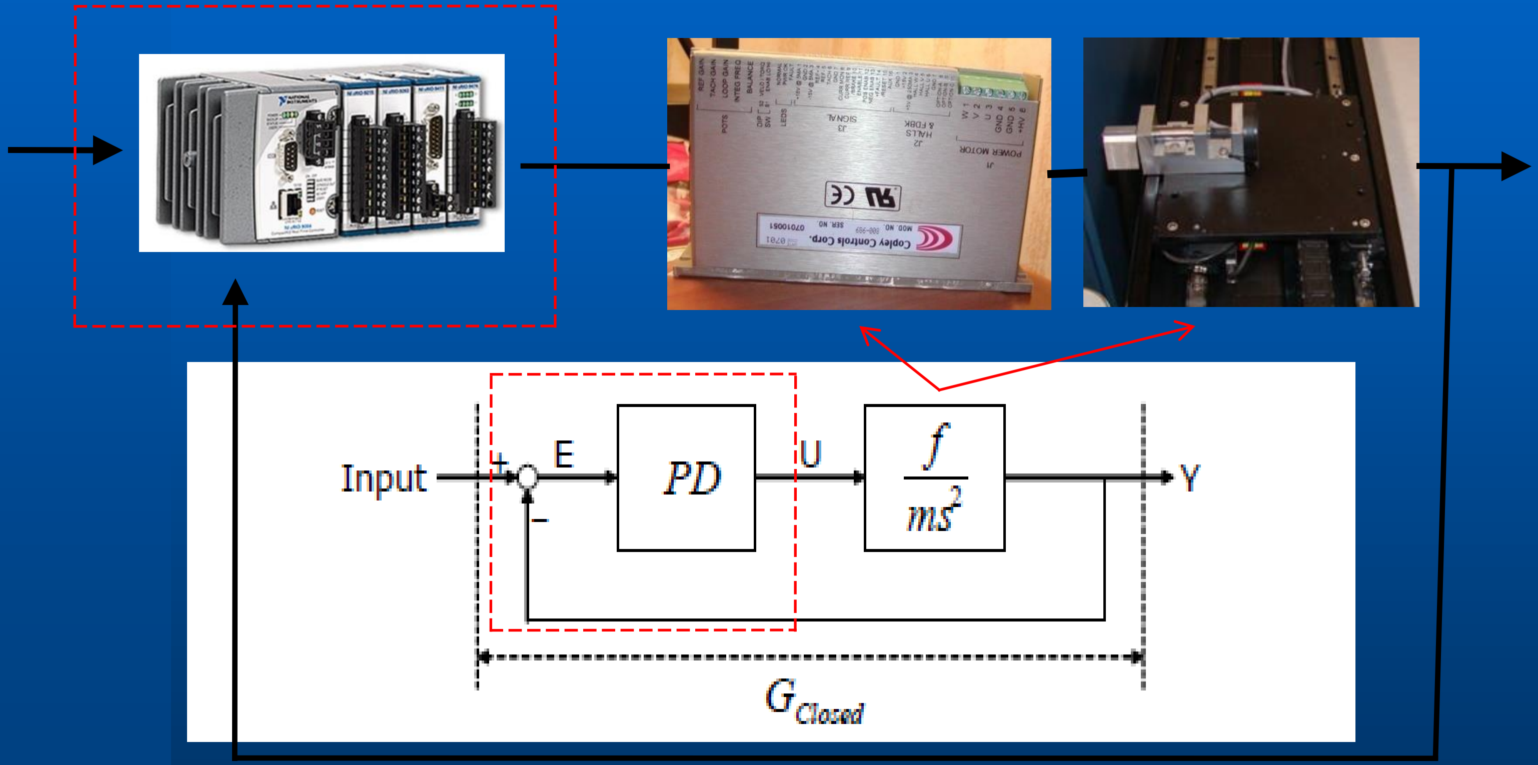
Stage: Belleverman LOWBOY 260

Motor: Trilogy Direct Drive
Linear Brushless Motor
96 Volts
5 Amps Peak
3 Amps Continuous

Drive: Copley Controls
Brushless DC Amplifier

Encoder: Renishaw RG22H
1 um resolution

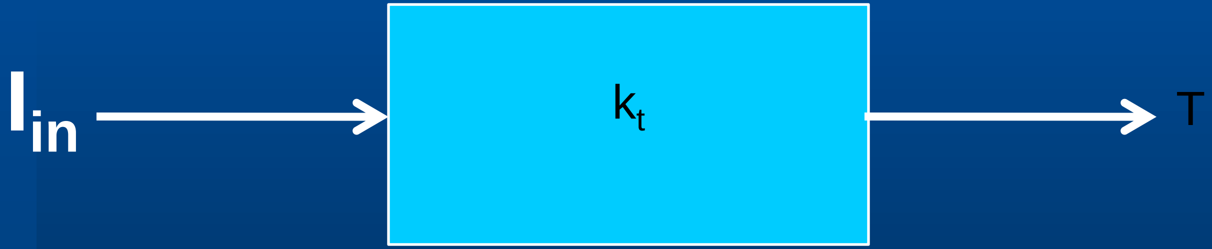
SYSTEM ID



Permanent Magnet DC motor

$$T = k_t * I_{in}$$

where k_t = Torque Constant



Model of an ideal inertial load

$$\Sigma T = J*\alpha$$

$$\alpha = d\omega/dt$$

$$\omega = d\theta/dt$$

where J = polar moment of inertia

α = angular acceleration

ω = angular velocity

θ = angular position

$$\Sigma T = J^*\alpha$$
$$\alpha = d\omega/dt$$

$$\Sigma T = J^*\, d\omega/dt$$
$$\omega = d\theta/dt$$

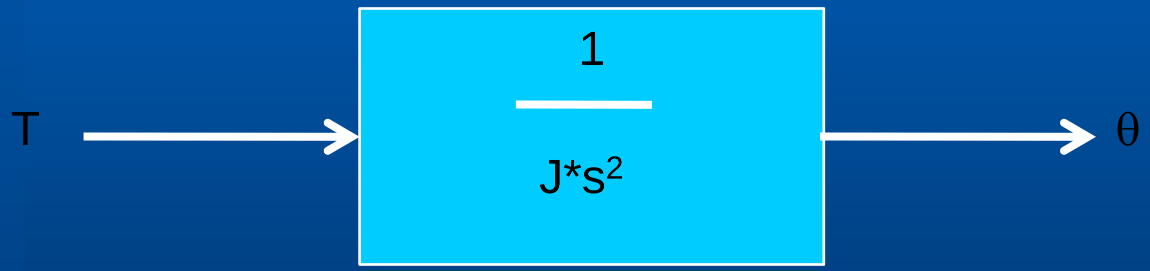
$$\Sigma T = J^*\, d^2\theta/dt^2$$

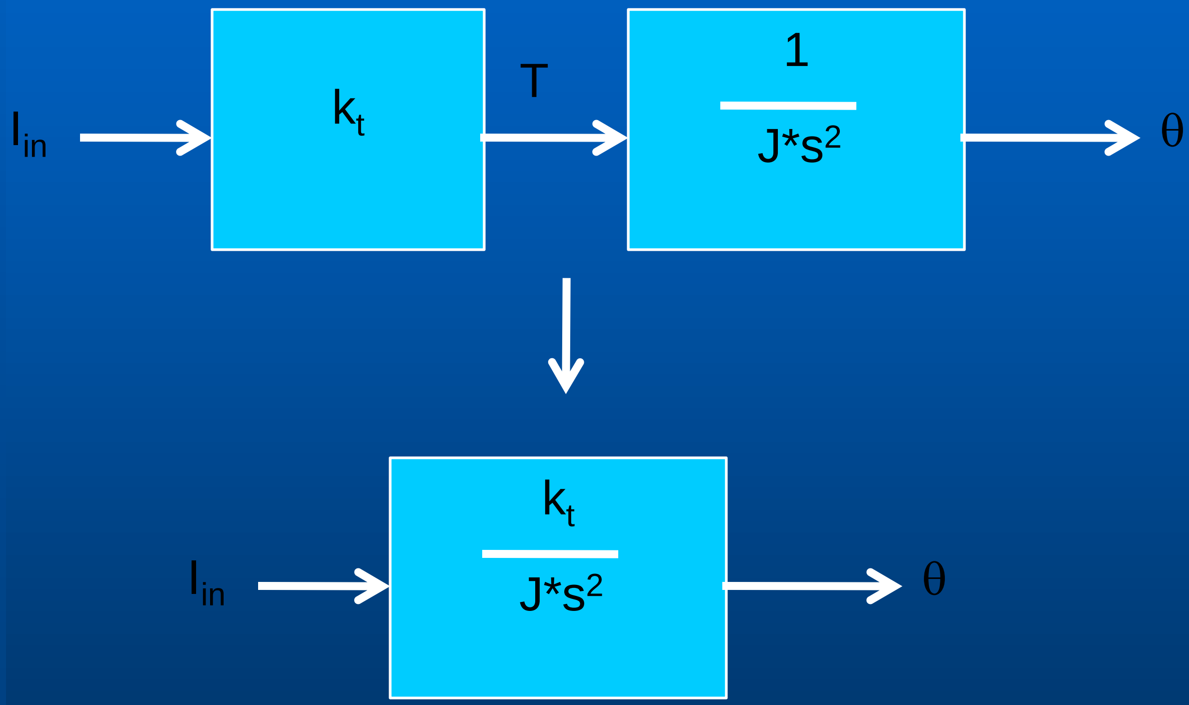
using Laplace Transform:

$$\Sigma T = J^*\, s^2\theta$$

$$\Sigma T = J^* s^2\theta$$

$$\Sigma T = J^* s^2\theta$$





I_{in} → k_t → T → $\frac{1}{J*s^2}$ → θ

I_{in} → $\frac{k_t}{J*s^2}$ → θ

Timeline: Aim to be done on May 7th

30 days left
(480 or 720 hours left w/o sleep)
(960 or 1440 FPGA compiles)

Average Group Size: 4 → 1920 or 2880 engineering hours

Amount of Shop Time: 176 hours

Realistically based on 4 hours/day: 120 Hrs / group member

Common Motion Trajectory Generation Methods:

- Trapezoidal Trajectory :
Constant Acceleration

Acc(k) = C;
Vel(k) = Vel(k-1) + Acc(k);
Pos(k) = Pos(k-1) +Vel(k);

- Decision points:

- Vel(k) > $V_{cruising}$, $Pos_{switch} = Pos(k)$
- $Pos_{switch2} = Pos_{destination} - Pos_{switch}$

Sensors

A sensor is a type of transducer which uses one type of energy, a signal of some sort, and converts it into a reading for the purpose of information transfer.

Source: Wikipedia

Temperature

- Thermocouples
- Thermistors
- RTD’s
- QCM’s

Position/Velocity

- Encoders (Linear and Rotary Digital/Analog)
- Potentiometers
- Laser Interferometer

- Tachometers
- Differentiating Encoders

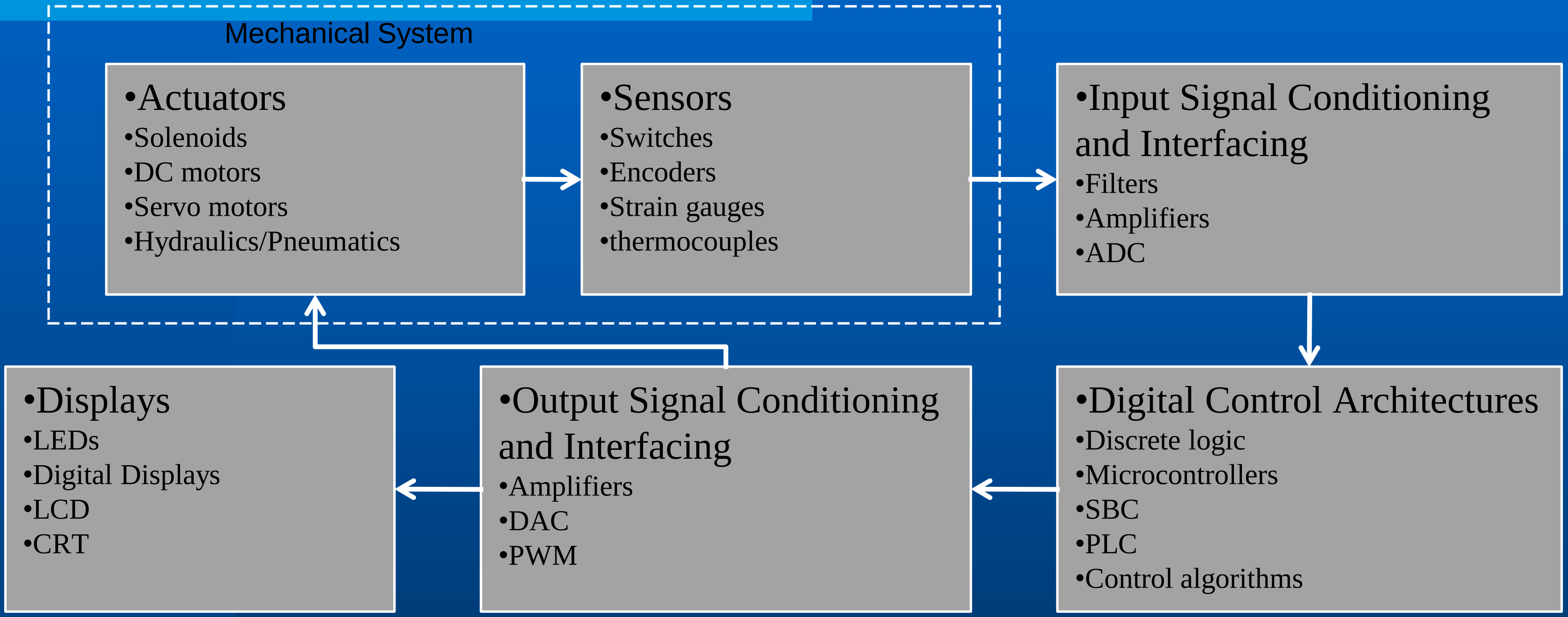
Force/Torques

- Strain gauges
- Load Cell

Acceleration

- Accelerometers (Piezo Type and Mems)

Mechatronic system components:



Mechanical System

- **Actuators**
  - Solenoids
  - DC motors
  - Servo motors
  - Hydraulics/Pneumatics

- **Sensors**
  - Switches
  - Encoders
  - Strain gauges
  - thermocouples

- **Input Signal Conditioning and Interfacing**
  - Filters
  - Amplifiers
  - ADC

- **Digital Control Architectures**
  - Discrete logic
  - Microcontrollers
  - SBC
  - PLC
  - Control algorithms

- **Output Signal Conditioning and Interfacing**
  - Amplifiers
  - DAC
  - PWM

- **Displays**
  - LEDs
  - Digital Displays
  - LCD
  - CRT

Luminary I/O:

General Purpose I/O:

42 Input-Output Lines over 5 ports
Some lines are shared

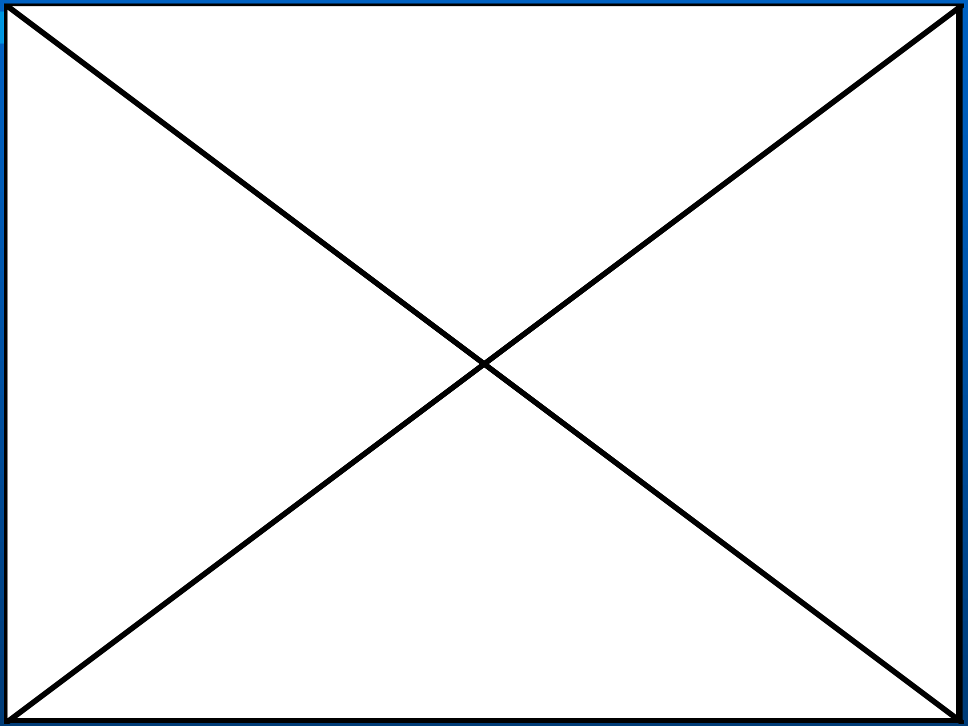
PA0-PA7
PB0-PB7
PC0-PC7
PD0-PD7
PE0-PE3
PF0-PF3
PG0-PG1

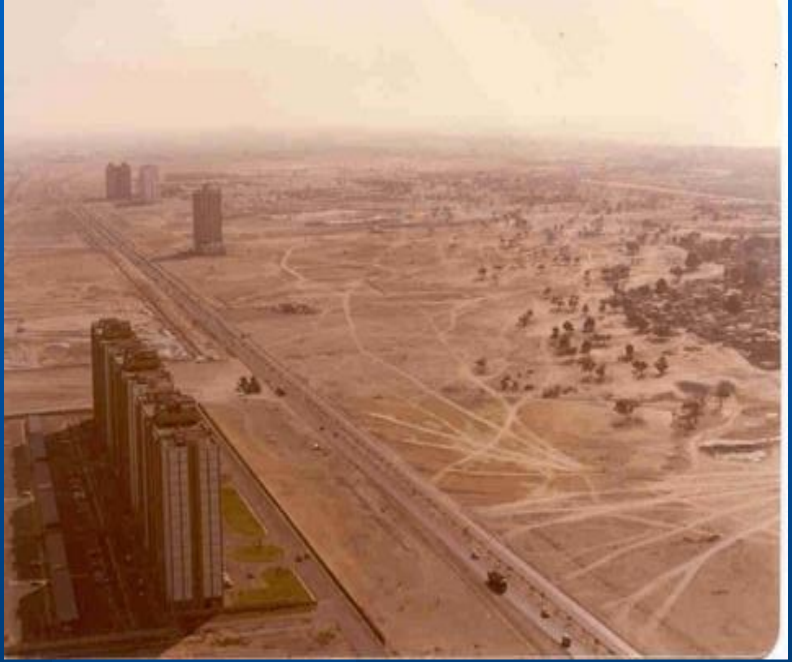
Totally Free Lines:

PA7
PC5
PC7
PD5
PG0

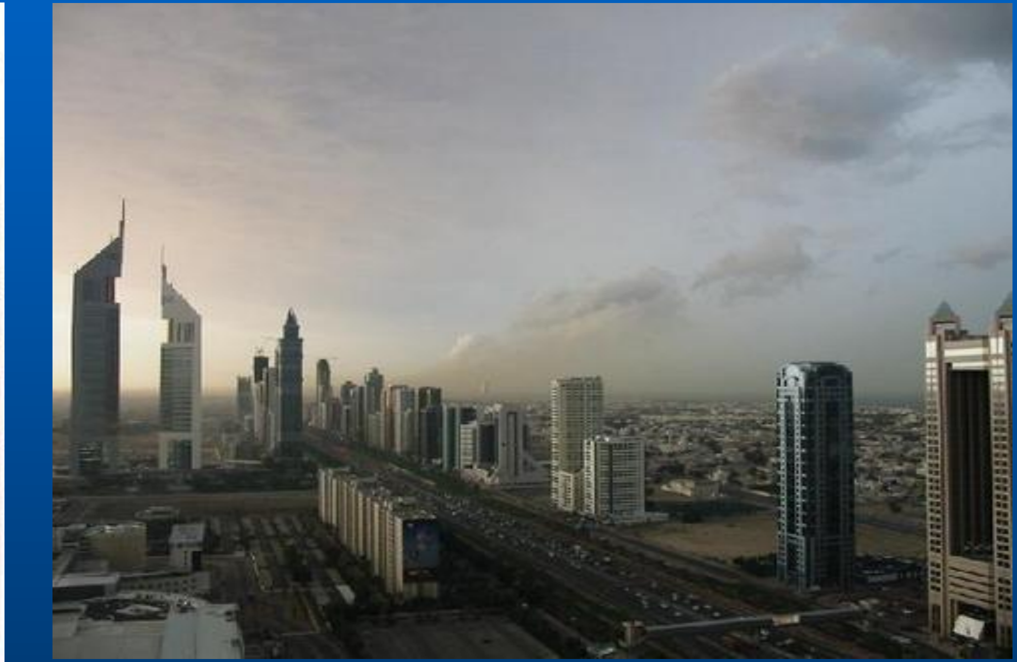
See you Thursday

Reminder: Proposals are due Thursday
First assignment due Friday





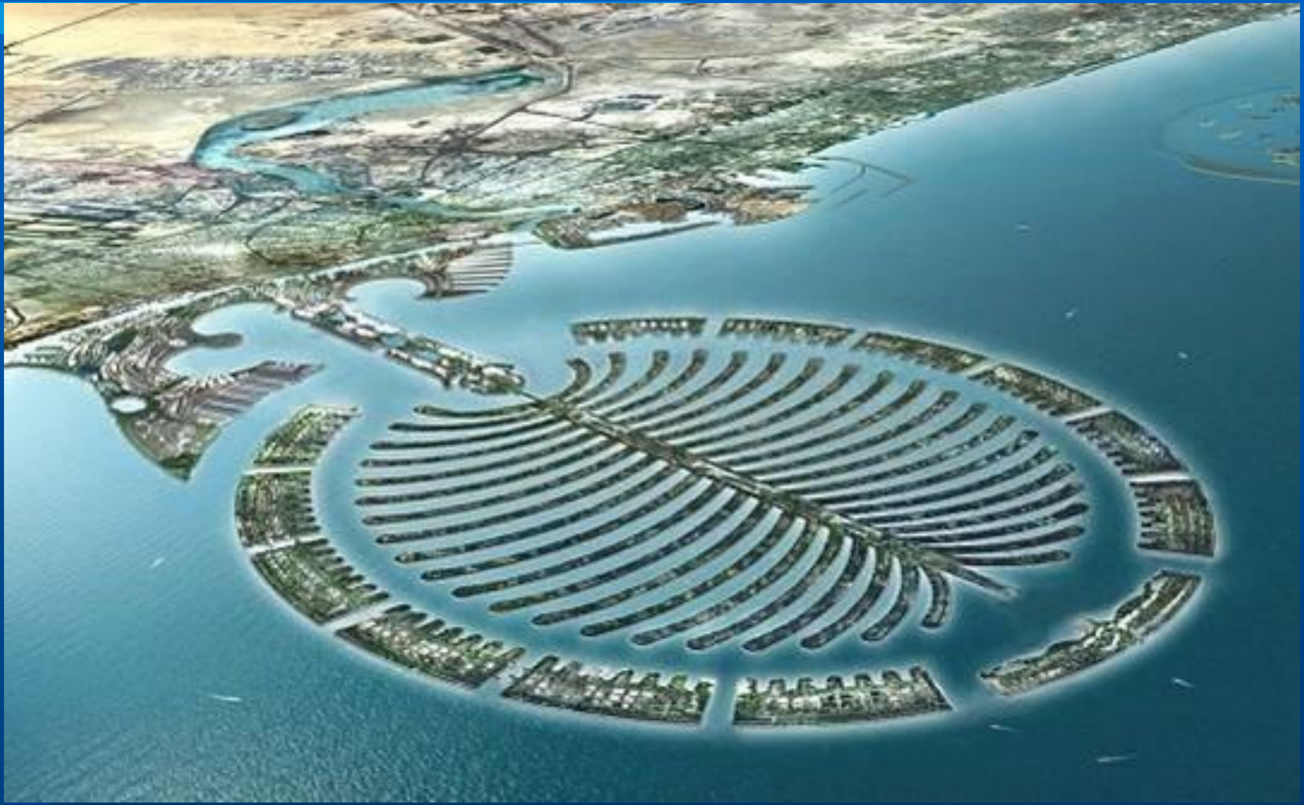
Dubai 1990



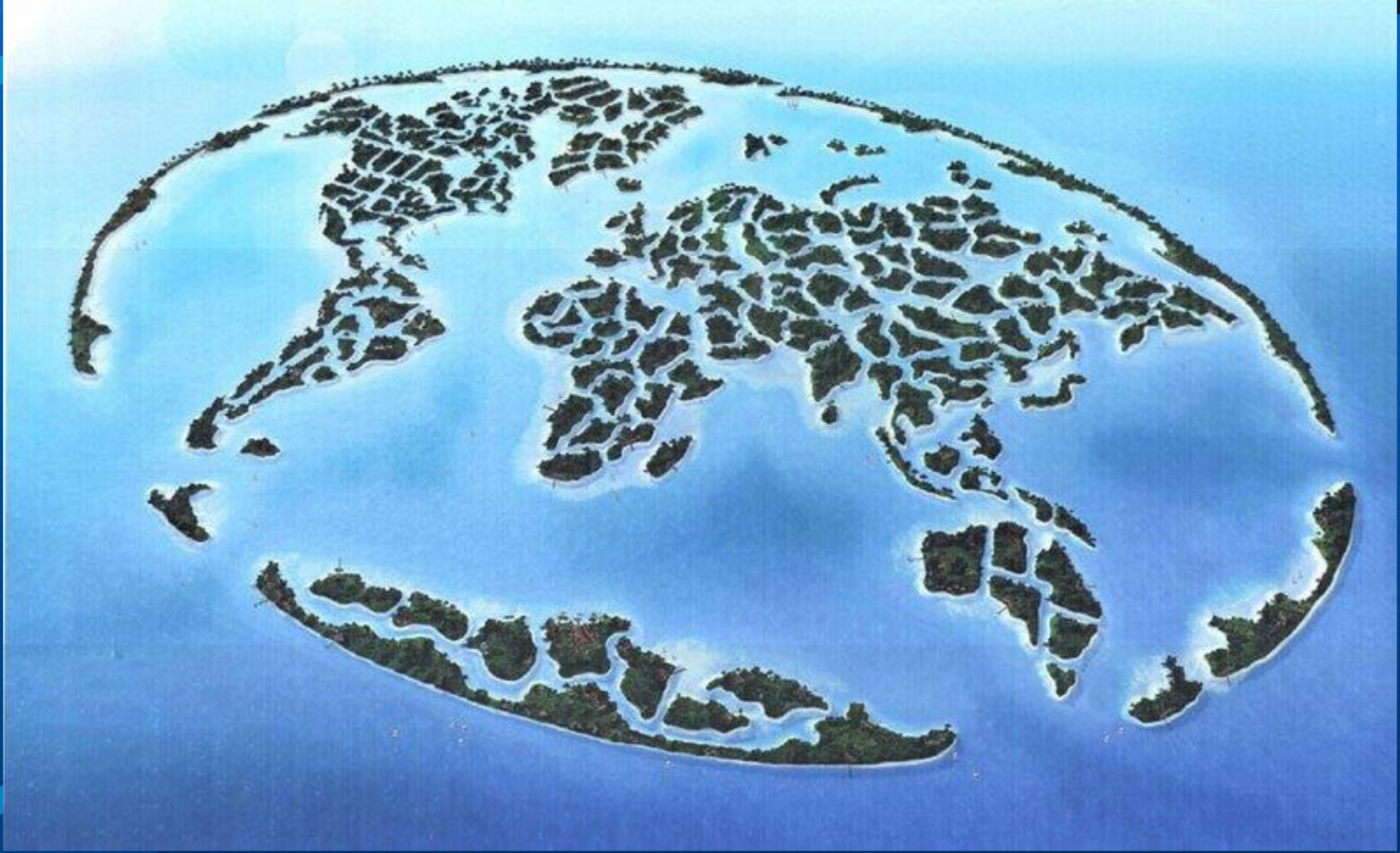
Same Street 2003

2007

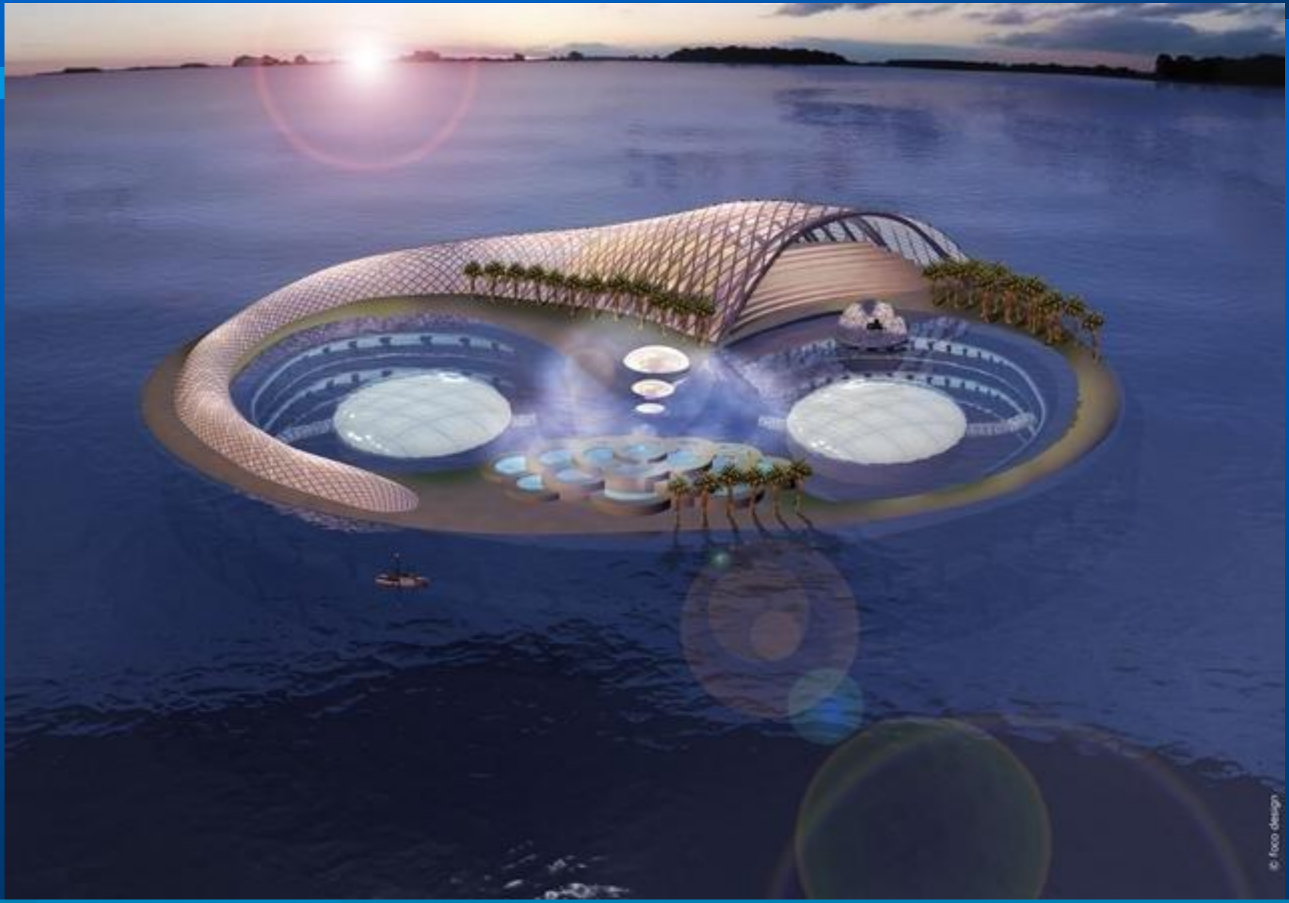
15%-20% of the world’ s cranes



Claim can be seen from the Moon

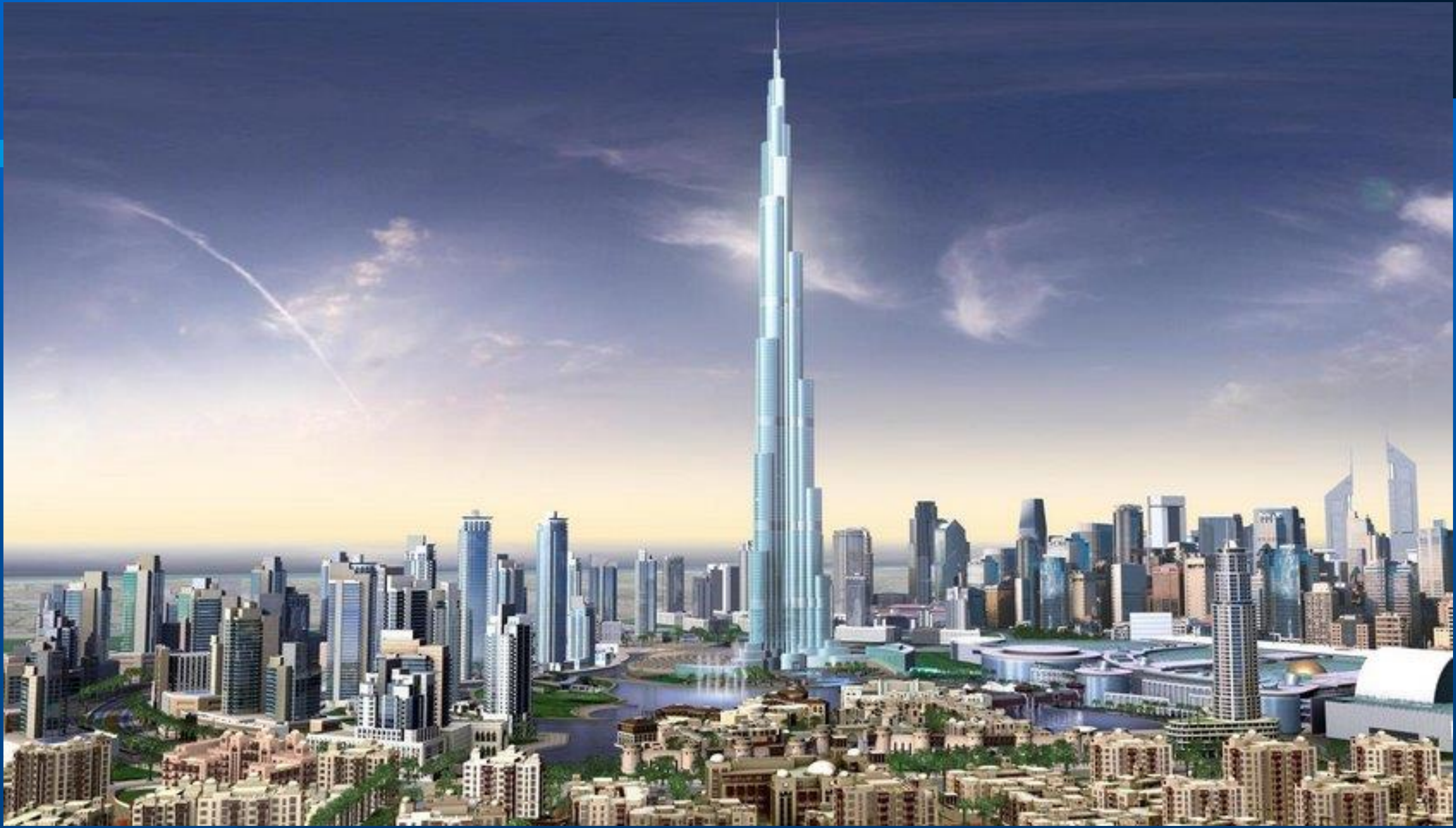


World Tallest Hotel on an Man made Island



First Underwater
Hotel

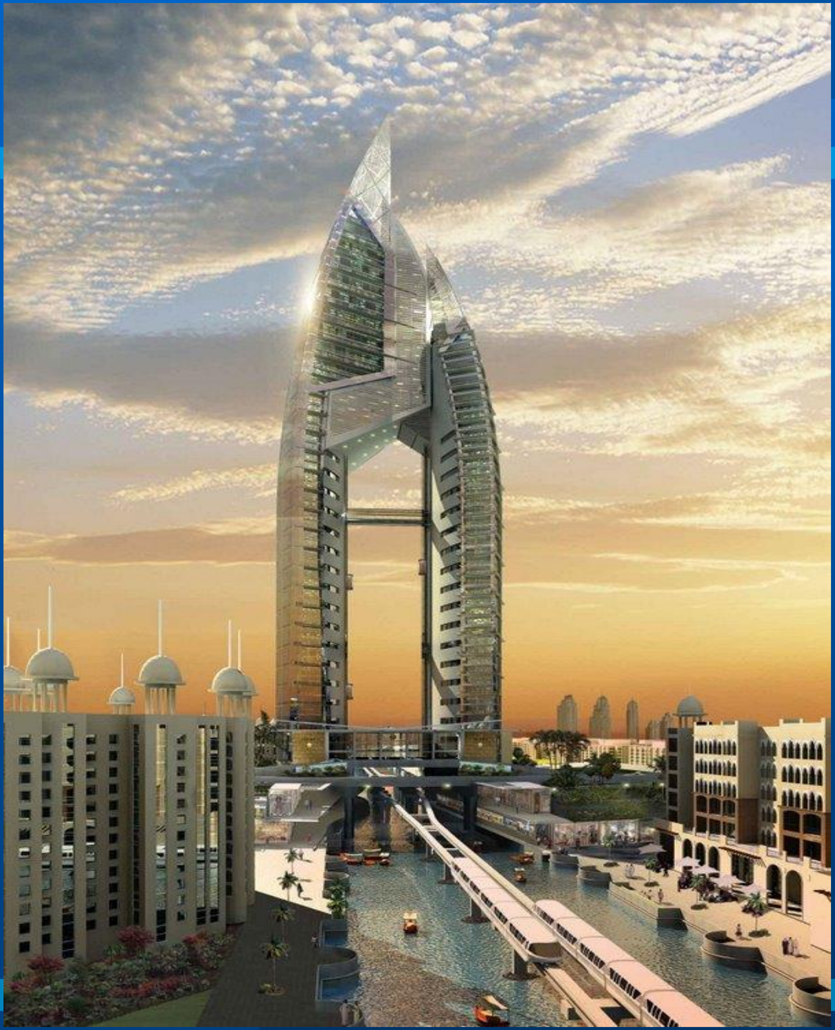
Constructed in Germany
Final Assembly on site



Will be 40% taller than the current record holder

Will be the tallest building
1200 meters high

Previous building is only
800 meters high



Trump International Hotel

ME135 Spring 2009 Summary:

- 60 Students
- 16 Groups
- Crane Game – Robot, Vision
- Folding Robot – Robot
- Music Writer – Sound
- Music Transcriber – Sound
- Chess Playing Robot – Robot, Vision
- Tic-Tac-Toe – Robot, Vision
- Hangman – Robot
- Putting Robot – Robot, Vision
- CNC Mill – Milling, Vision
- Musical Glass – Motion

Demo Day : May 9th 10:00 AM -- ??

Next week: Lecture Time we will meet in lab for one on one consultation if necessary, and debugging sessions

Lab will be open this weekend and next weekend, and evenings all next week

LMD18200 H-Bridge:

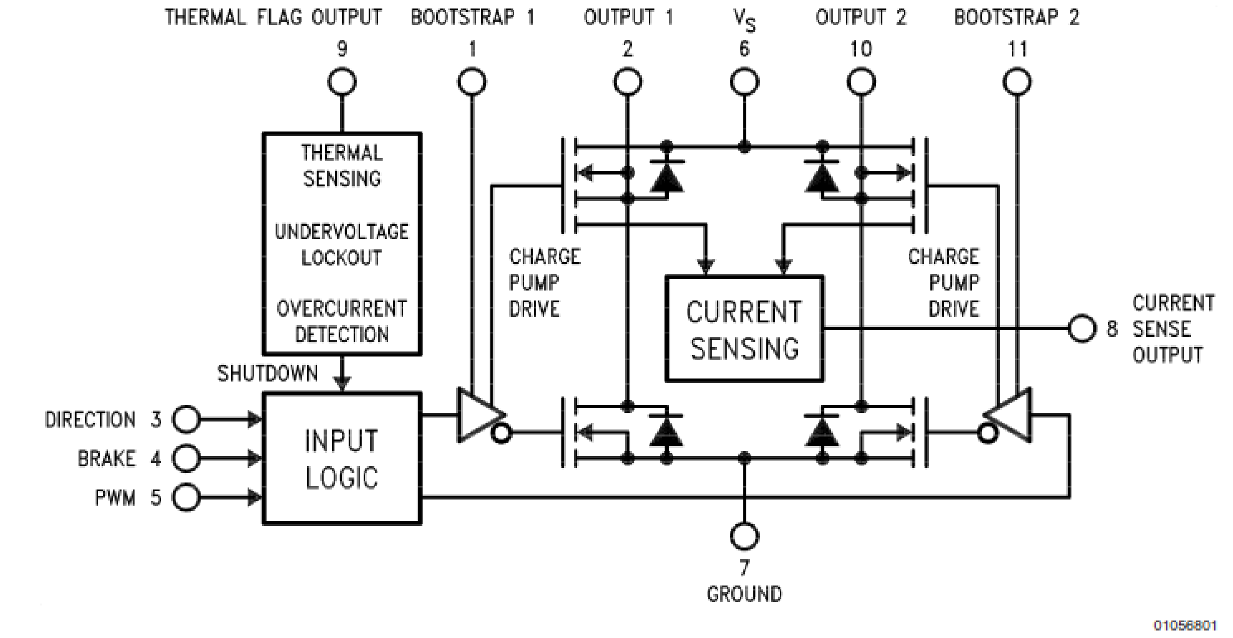
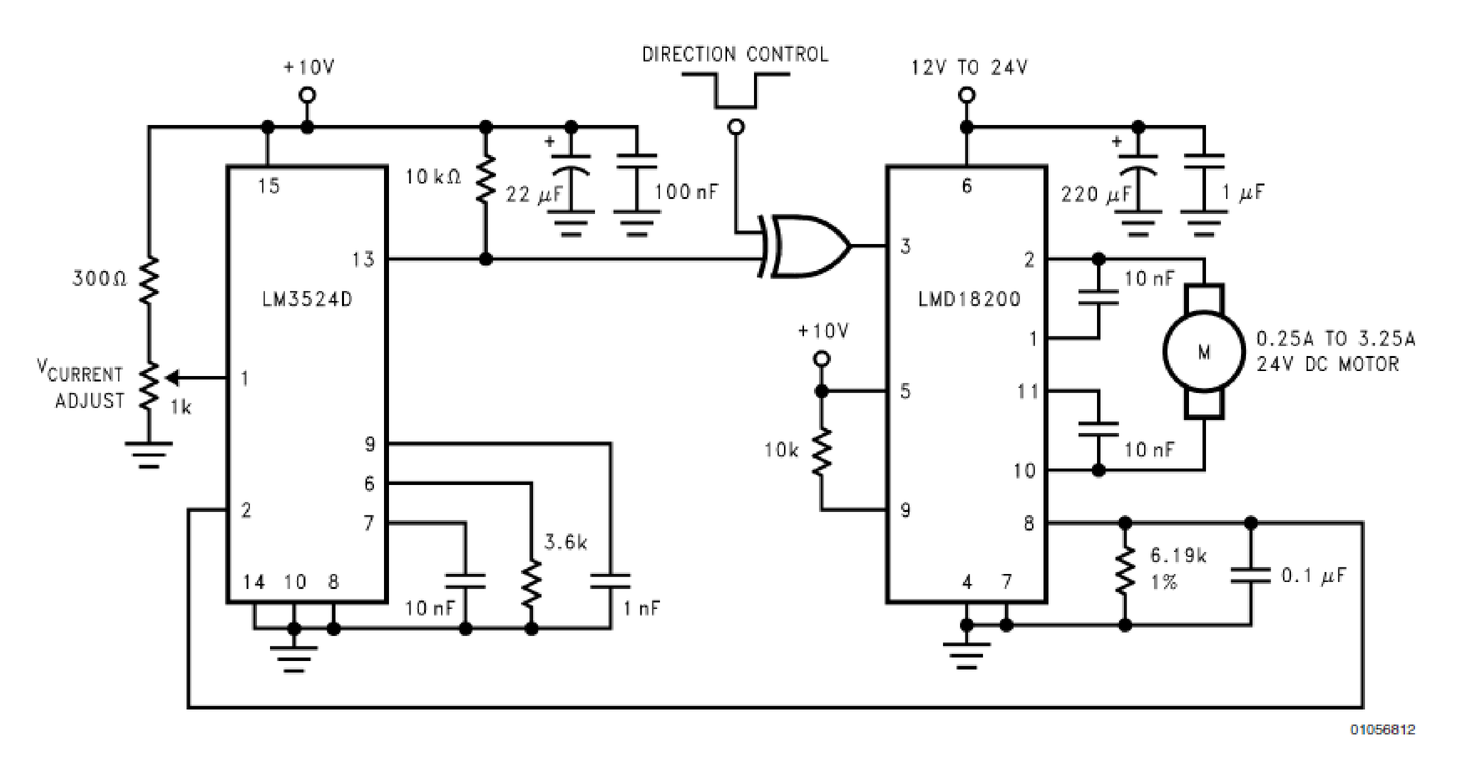


FIGURE 1. Functional Block Diagram of LMD18200

Motor with Current (Psuedo-Torque) Feedback:



Common Motion Trajectory Generation Methods:

- Trapezoidal Trajectory :

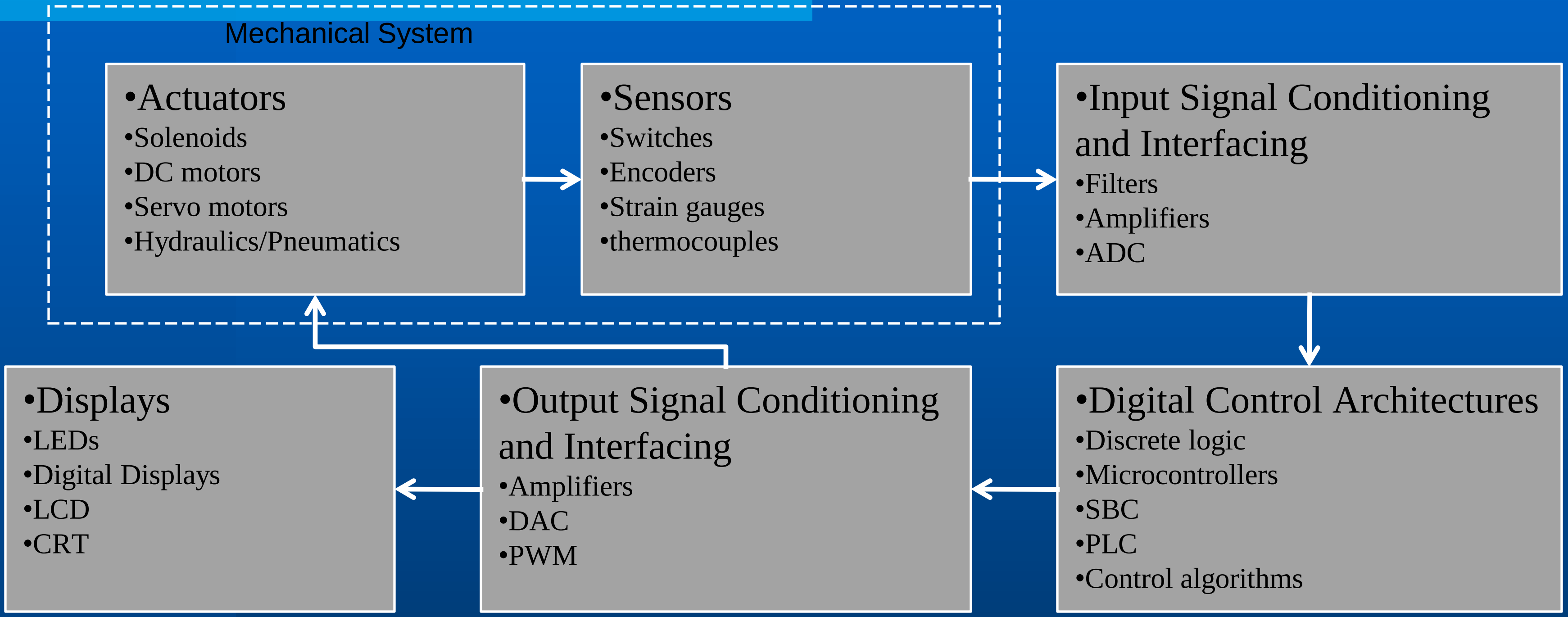
Constant Acceleration

Acc(k) = C;
Vel(k) = Vel(k-1) + Acc(k);
Pos(k) = Pos(k-1) +Vel(k);

- Decision points:

- Vel(k) > $V_{cruising}$, Pos_{switch} = Pos(k)
- $Pos_{switch2}$ = $Pos_{destination}$ - Pos_{switch}

Mechatronic system components:



Mechanical System

- Actuators
  - Solenoids
  - DC motors
  - Servo motors
  - Hydraulics/Pneumatics

- Sensors
  - Switches
  - Encoders
  - Strain gauges
  - thermocouples

- Input Signal Conditioning and Interfacing
  - Filters
  - Amplifiers
  - ADC

- Digital Control Architectures
  - Discrete logic
  - Microcontrollers
  - SBC
  - PLC
  - Control algorithms

- Output Signal Conditioning and Interfacing
  - Amplifiers
  - DAC
  - PWM

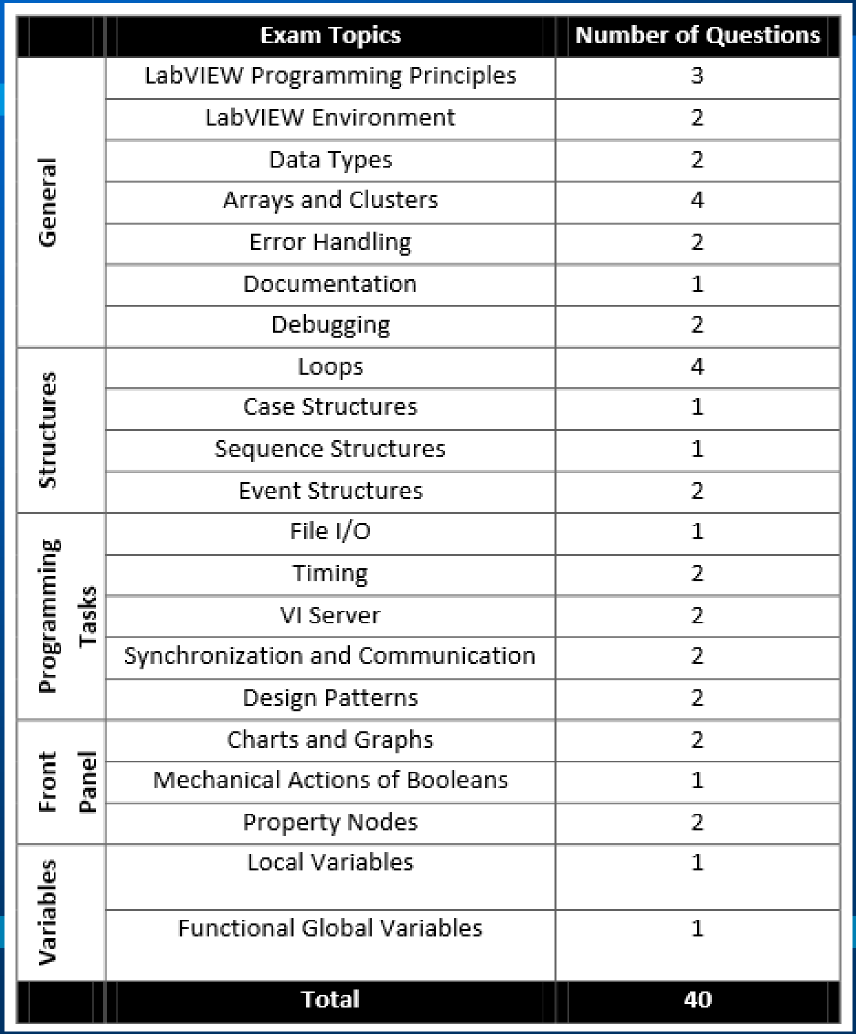
- Displays
  - LEDs
  - Digital Displays
  - LCD
  - CRT

Demo Day: May 8th
Final Report : By the end of the final (May 15th)

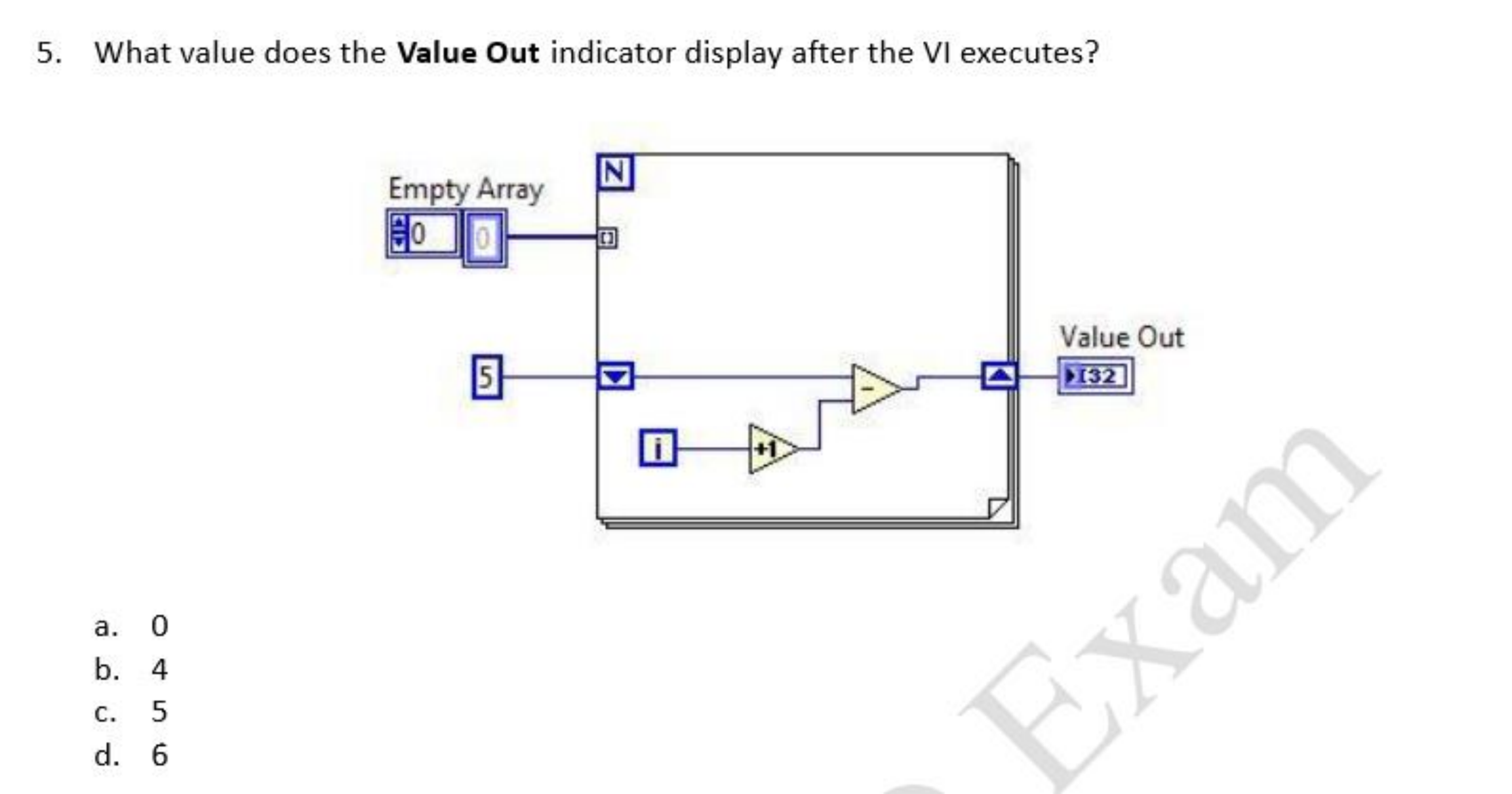
What to include in the report:

- Background and Motivation
- Technical Challenges prior to proposal
- Actual technical challenges (highlight code solution to challenges)
- Now that you know what you know---
What would you do different if you had access to a time machine
- Your code
- Journal (if any)

CLAD Exam Topics:



	Exam Topics	Number of Questions
General	LabVIEW Programming Principles	3
	LabVIEW Environment	2
	Data Types	2
	Arrays and Clusters	4
	Error Handling	2
	Documentation	1
	Debugging	2
Structures	Loops	4
	Case Structures	1
	Sequence Structures	1
	Event Structures	2
Programming Tasks	File I/O	1
	Timing	2
	VI Server	2
	Synchronization and Communication	2
	Design Patterns	2
Front Panel	Charts and Graphs	2
	Mechanical Actions of Booleans	1
	Property Nodes	2
Variables	Local Variables	1
	Functional Global Variables	1
	Total	40



5. What value does the **Value Out** indicator display after the VI executes?

a. 0
b. 4
c. 5
d. 6