

EE445M/EE360L.12 Embedded and Real-Time Systems/ Real-Time Operating Systems

Lecture 9: Sensing & Acting, Input Capture, PWM, Motors

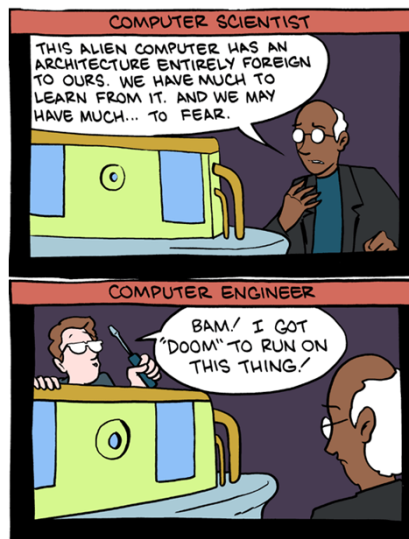
Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

1

EE445M vs. EE461S/CS372

THE DIFFERENCE:



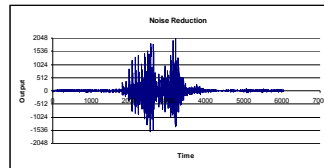
<http://www.smbc-comics.com/index.php?db=comics&id=2158#comic>

Lecture 9

2

Class Agenda

- Recap: RTOS Kernel
 - Multi-tasking, context switch, scheduling
 - Synchronization, communication, semaphores
 - File system, memory management
- Outlook: Applications of RTOS
 - Lab 6: Robot interfaces
 - Sensors, Motors
 - Networking
 - Lab 7: Robot control



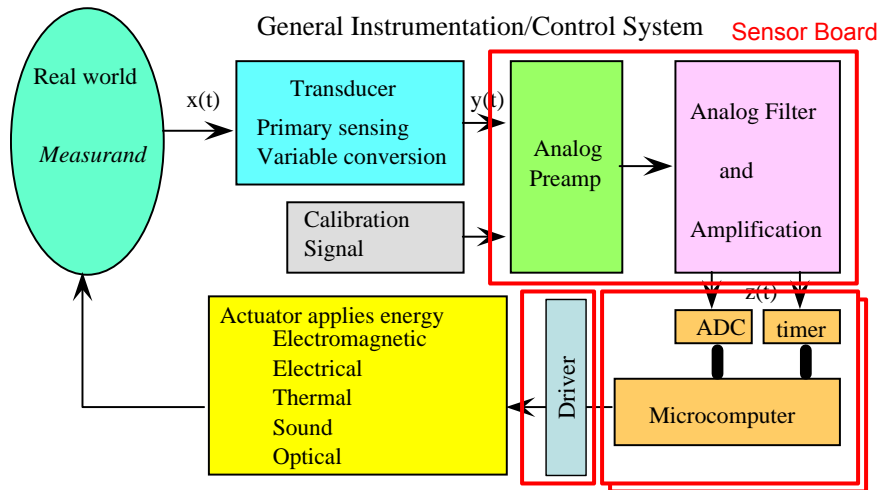
Reference book, Chapter 10

Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

3

Instrumentation & Control



Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

Motor Board
2x LaunchPad
connected via CAN 4

Sensor Board

- **Reference material**
 - Schematic: http://www.ece.utexas.edu/~gerst/ee445m_s16/resources/Robot_Sensor_v3.pdf
 - PCB layout: http://www.ece.utexas.edu/~gerst/ee445m_s16/resources/sensor_top3.png

Lecture 9 J. Valvano, A. Gerstlauer
EE445M/EE380L.12 5

Sharp GP2Y0A21YK

- Infrared distance sensor
 - Distance to analog voltage
 - Powered by 5V
 - 10 uF or larger +5V to Gnd cap for each sensor
 - Needs analog LPF
 - Reduces noise
 - Analog input protection
 - Needs digital median filter
 - Needs calibration



Sharp GP2Y0A21YK

- Noise => Filtering
- Accuracy => Calibration
- Precision => Resolution



$$\text{ADC} = 6707/d + 40$$

$$d = 6707/(\text{ADC} - 40)$$

$$d (0.01\text{cm}) = 6706700/(\text{ADC} - 40)$$

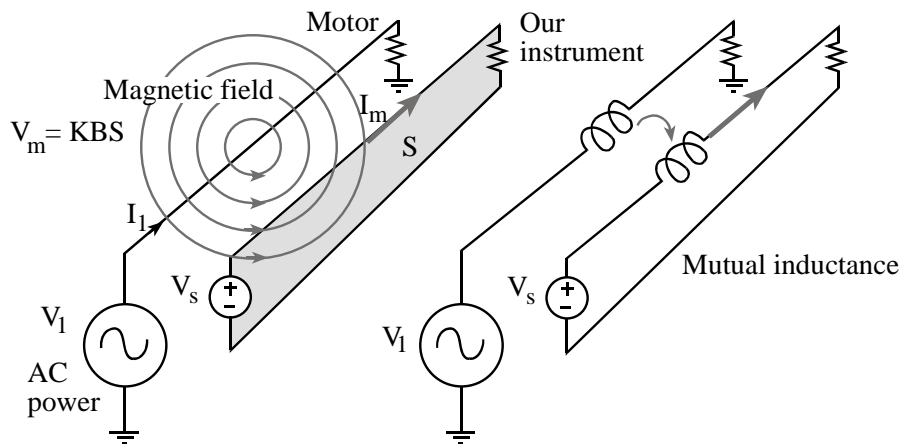


Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

7

Magnetic Field Noise

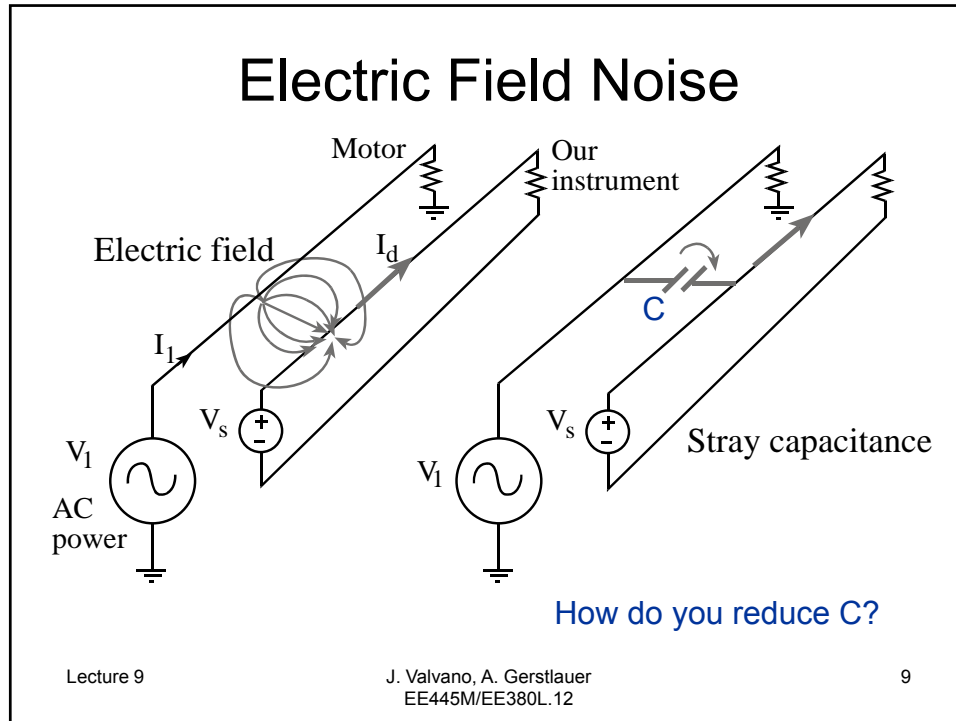


How do you reduce S?

Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

8



Noise Measurement

- Digital volt meter (AC mode)
 - Most accurate quantitative noise measure
- Oscilloscope (line trigger)
 - Shape
- Spectrum Analyzer
 - Classify type

White or 1/f noise

Periodic noise

60Hz noise

1/f noise

white noise

bandwidth limits

Noise Reduction (1)

1. Reducing noise from the source

- Shielding
 - Enclose noisy sources in a grounded metal box
- Filter noisy signals
- Limit the rise/fall times of noisy signals.
- Limiting the di/dt in the coil.



Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

11

Noise Reduction (2)

2. Limiting the coupling between the noise source and instrument

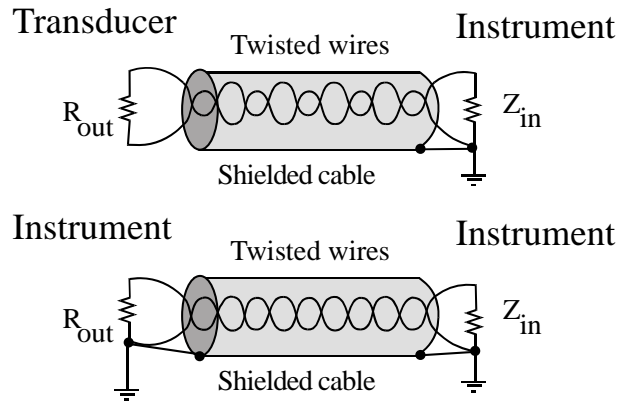
- Maximize the distance from source to instrument
- Cables with noisy signals should be twisted together
- Cables should also be shielded.
- For high frequency signals, use coaxial
- Reduce the length of a cable
- Place the delicate electronics in a grounded case
- Optical or transformer isolation circuits

Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

12

Limiting the Coupling



Lecture 9

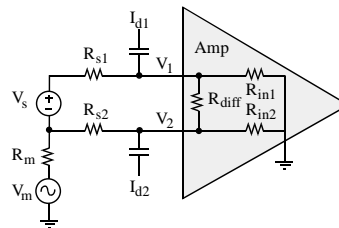
J. Valvano, A. Gerstlauer
EE445M/EE380L.12

13

Noise Reduction (3)

3. Reduce the noise at the receiver

- Bandwidth should be as small as possible
- Add frequency-reject filters
- Use power supply decoupling capacitors on each
- Twisted wires then I_{d1} should equal I_{d2}
 - $V_1 - V_2 = R_{s1} I_{d1} - R_{s2} I_{d2}$



Henry Ott, *Noise Reduction Techniques in Electronic Systems*, Wiley, 1988.
Ralph Morrison, *Grounding and Shielding Techniques*, Wiley, 1998.

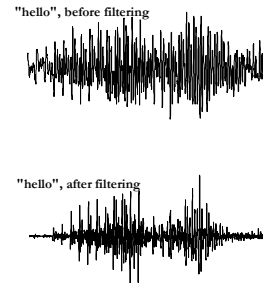
Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

14

Filter Types

- Analog
 - Low pass filter (LPF)
 - High pass filter (HPF)
 - Band pass filter (BPF)
- Digital
 - Extremely flexible
 - But only available after sampling



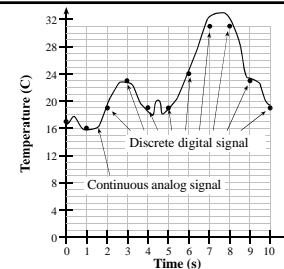
Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

15

Sampling

- Time & value quantizing
 - Precision $n_z = 2^n$
- Nyquist theory
 - If sampled at f_s , digital samples only contain frequency components from 0 to $\frac{1}{2}f_s$
 - If analog signal contains frequency components larger than $\frac{1}{2}f_s$, **aliasing** error
- System design
 - Choice of sampling rate: $f_s > 2 f_{\max}$
 - Low pass analog filter to remove frequency components above $0.5f_s$
 - A digital filter can not be used to remove aliasing



Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

16

Analog Filters

- Prevent aliasing
 - No signal $> 0.5f_s$
- Band-pass filter

frequency $\frac{1}{2}f_s$

Gain $G_3 = |H_3(s)|$
 Pass $f_{min} \leq f \leq f_{max}$
 Min. error seen by ADC

Frequency

Lecture 9

17

Butterworth Filters

$$\frac{V_{out}}{V_{in}} = \sqrt{\frac{1}{1 + (\frac{f}{f_c})^4}}$$

- 2-pole Butterworth low-pass filter (LPF)
 1. Select the cutoff frequency, f_c
 2. Divide the two capacitors by $2\pi f_c$
 - $C_{1A} = 141.4\mu F / 2\pi f_c$
 - $C_{2A} = 70.7\mu F / 2\pi f_c$
 3. Locate two standard value capacitors (with 2/1 ratio) in the same order of magnitude as the desired values
 - $C_{1B} = C_{1A}/x$
 - $C_{2B} = C_{2A}/x$
 4. Adjust the resistors to maintain the cutoff frequency
 - $R = 10k\Omega \cdot x$

See Sensor Board schematic

Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

18

Digital Filters

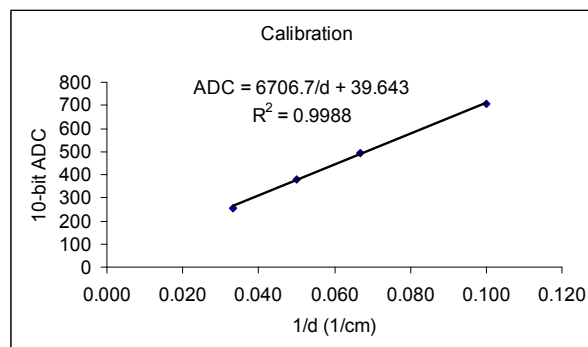
- Finite/Infinite Impulse Response (FIR/IIR)
 - Linear: HPF, LPF, BPF/notch
- Median filter
 - Non-linear: preserves edges, removes spikes

```
// Non-recursive, 3-point median filter
uint8_t Median(uint8_t u1,uint8_t u2,uint8_t u3) {
    if(u1>u2)
        if(u2>u3) return u2;    // u1>u2,u2>u3    u1>u2>u3
        if(u1>u3) return u3;    // u1>u2,u3>u2,u1>u3 u1>u3>u2
        return u1;              // u1>u2,u3>u2,u3>u1 u3>u1>u2
    else
        if(u3>u2) return u2;    // u2>u1,u3>u2    u3>u2>u1
        if(u1>u3) return u1;    // u2>u1,u2>u3,u1>u3 u2>u1>u3
        return u3;              // u2>u1,u2>u3,u3>u1 u2>u3>u1
}
```

[Reference book, Chapter 6](#)

IR Sensor Calibration

d (cm)	1/d	ADC
10	0.100	703
15	0.067	484
20	0.050	380
30	0.033	260



Lecture 9

Ping Distance Sensor

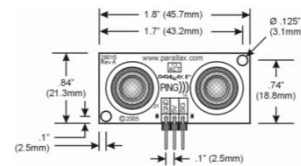
- Ultrasound transducers to measure distance

- Ping)))

- One **SIG** pin for both input & output

- HCSR04

- Two signals:
Trig output and **Echo** input



- Need 5V to power

- Use 5V tolerant input (PD4, PD5, PB0 and PB1 must be <3.6V)

Lecture 9

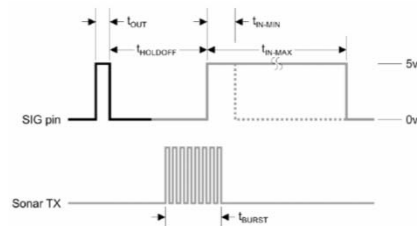
J. Valvano, A. Gerstlauer
EE445M/EE380L.12

21

Ping))) Sensor

- Sample 10 times a second

- 1) Disable interrupts
- 2) Make the **SIG** pin an output
- 3) Issue a $5\mu\text{s}$ output pulse (causing a sound pulse)
- 4) Switch the **SIG** pin back to an input
- 5) Enable interrupts
- 6) Measure time until the echo is received
 - Busy-wait if foreground, interrupt if background



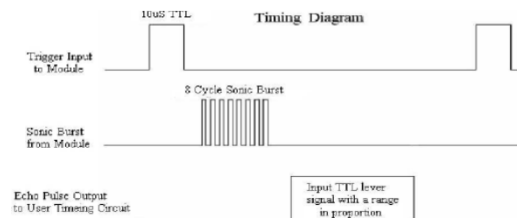
Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

22

HCSR04 Sensor

- Sample 10 times a second
 - 1) Disable interrupts
 - 2) Issue a 10 μ s output pulse (causing a sound pulse)
 - 3) Enable interrupts
 - 4) Measure time until the echo is received
 - Busy-wait if foreground, interrupt if background



Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

23

Input Capture

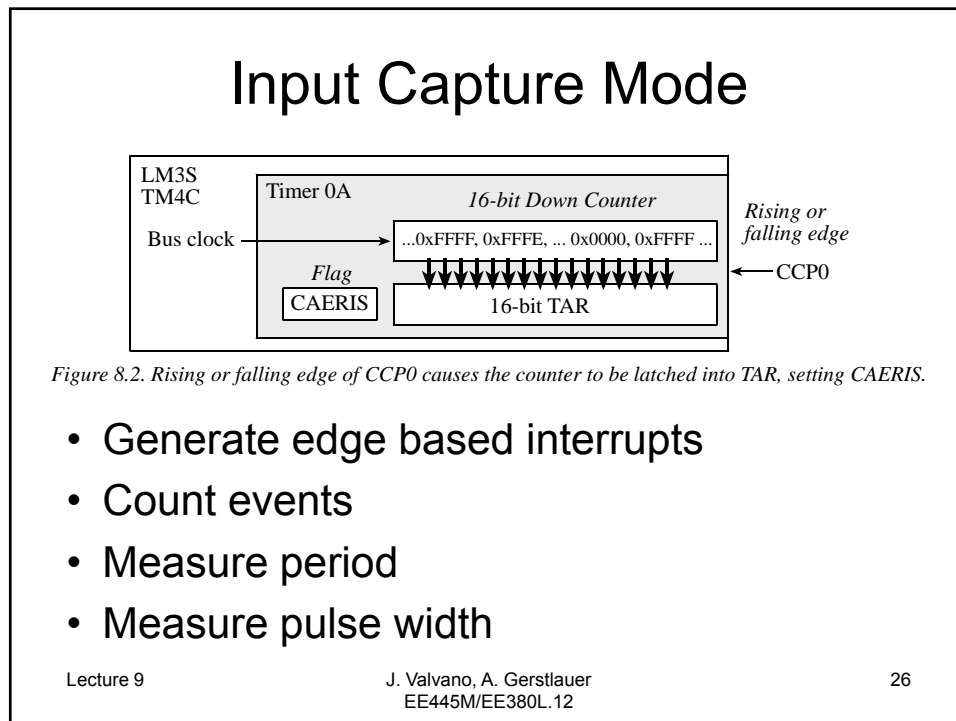
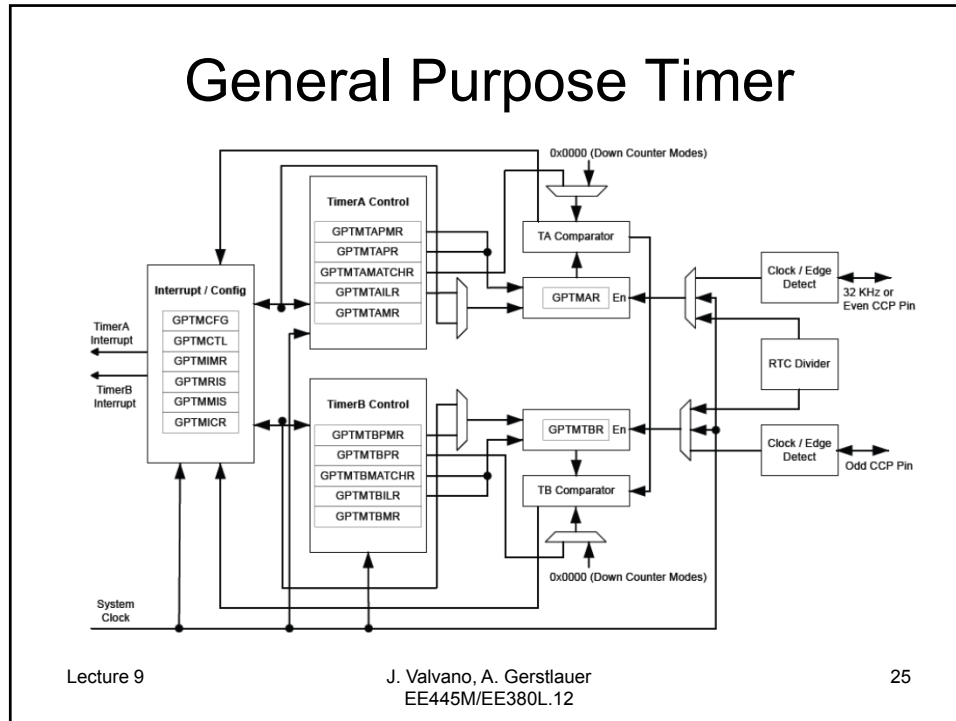
- General purpose timers
 - TM4C123: 6 GP timers (Timer 0...Timer 5)
 - CCPx pins used for input capture
- Input edge time (input capture) mode
 - Detect rising/falling input edges
 - Make time measurements on input signals

[See book Section 8.1](#)

Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

24



Event Counting

- Count wheel turns (tachometer)

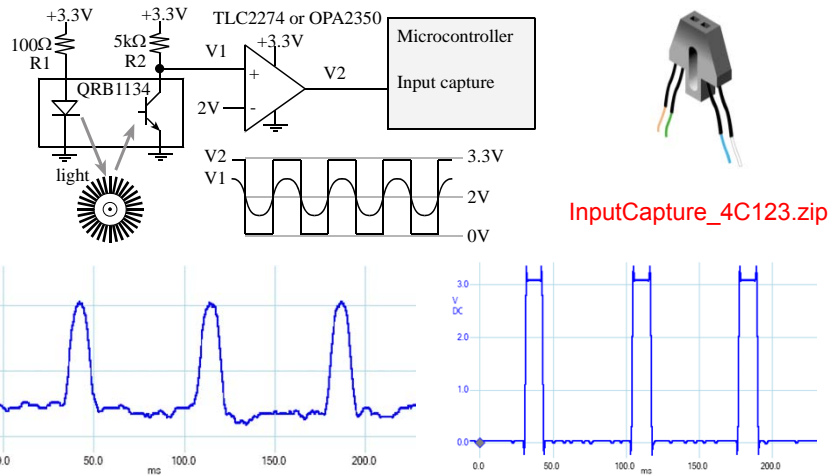


Figure 8.4. Measured V1 and V2

Period Measurement

- Init
 - Select clock period, Δt (measurement resolution)
 - TIMER0_TAILR_R = 0xFFFF (reload=wraparound)
 - Choose edge (rise or fall)
 - Arm interrupt on capture
- ISR
 - Poll to see which channel (if needed)
 - Now = captured time (TIMER0_TAR_R)
 - Period = Last - Now
 - Last = Now
 - Acknowledge interrupt
 - Save/process period

PeriodMeasure_4C123.zip

Resolution, Precision, Range

- How to choose the resolution?
 - Determine minimum & maximum robot speed
 - Convert speed to tachometer period

Period	7100	
	4	holes/rotation
Resolution	10	μsec
Speed	3.521127	rps
Speed	211.2676	RPM

- How to detect speed too slow (period too large)?
 - Clear a counter on each tachometer edge
 - AddPeriodicThread
 - Increment the counter on each rollover 0000 to FFFF
 - If counter ≥ 2 , then wheel is stopped

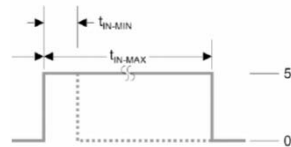
Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

29

Ping Distance Measurement

- Input pulse width
 - Time t_{IN} for sound to travel back and forth
 - $t_{IN} = 2 d/c$ (c : speed of sound)
- Measure using input capture
 - Rising edge: record TAR
 - Falling edge: calculate distance $d = c * t_{IN}/2$



Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

30

Motor Board

- **Reference material**
 - Schematic: http://www.ece.utexas.edu/~gerstl/ee445m_s18/resources/Robot_Motor_v6.pdf
 - PCB layout: http://www.ece.utexas.edu/~gerstl/ee445m_s18/resources/Robot_Motor_v6_top.png

Lecture 9 J. Valvano, A. Gerstlauer 31
 EE445M/EE380L.12

Motor Interfacing

- Motor physics
- Transistor-level interface

Lecture 9 J. Valvano, A. Gerstlauer 32
 EE445M/EE380L.12

Motor Physics

Lecture 9 J. Valvano, A. Gerstlauer 33
 EE445M/EE380L.12

Digital Interfacing

V_{OL} is defined as the voltage at maximum I_{OL}

Family	Example	I_{OH}	I_{OL}	I_{IH}	I_{IL}	fan out
Standard TTL	7404	0.4 mA	16 mA	40 μ A	1.6 mA	10
Schottky TTL	74S04	1 mA	20 mA	50 μ A	2 mA	10
Low Power Schottky	74LS04	0.4 mA	4 mA	20 μ A	0.4 mA	10
High speed CMOS	74HC04	4 mA	4 mA	1 μ A	1 μ A	
LM3S/LM4F 2mA-drive	LM3S811	2 mA	2 mA	2 μ A	2 μ A	
LM3S/LM4F 4mA-drive	LM3S811	4 mA	4 mA	2 μ A	2 μ A	
LM3S/LM4F 8mA-drive	LM3S811	8 mA	8 mA	2 μ A	2 μ A	

Electrical specifications

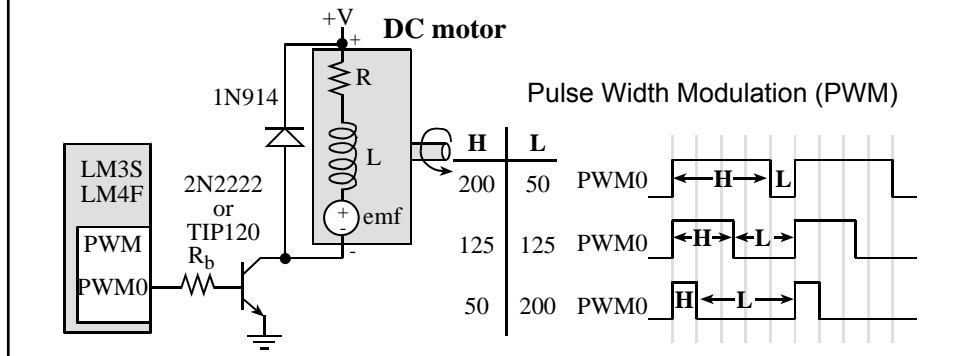
- See Chapter 24 of TM4C123
- 5V tolerant?
- PD0, PD1 \Leftrightarrow PB7, PB6

All GPIO signals are 5-V tolerant when configured as inputs except for PD4, PD5, PB0 and PB1, which are limited to 3.6 V.

Lecture 9 J. Valvano, A. Gerstlauer 34
 EE445M/EE380L.12

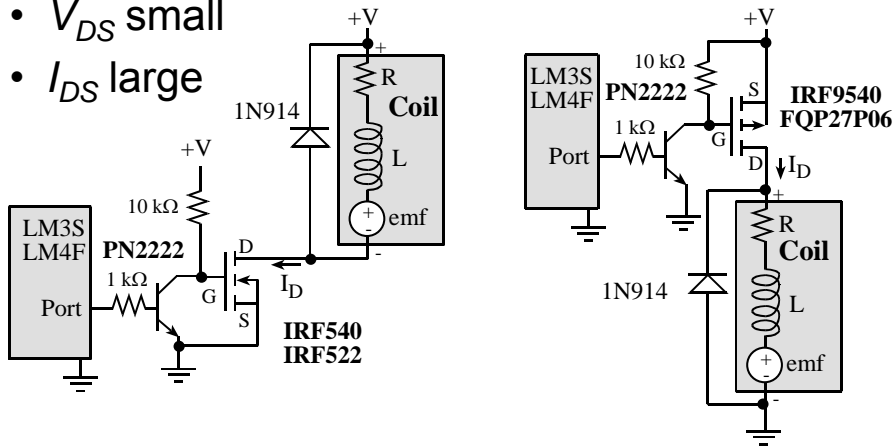
Motor Interface

- Darlington transistor
 - TIP120 (NPN) $I_b = I_{coil}/h_{fe} = 1A/1000 = 1mA$
 - $h_{fe} = 1000$ $R_b \leq (V_{OH} - V_{be})/I_b = (3-2.5)/1mA = 0.5 k\Omega$
 - $I_{ce} = 3A$ $R_b = 100 \Omega$
 - V_{CE} depends on current



MOSFET Interface

- V_{GS} turns on
- V_{DS} small
- I_{DS} large



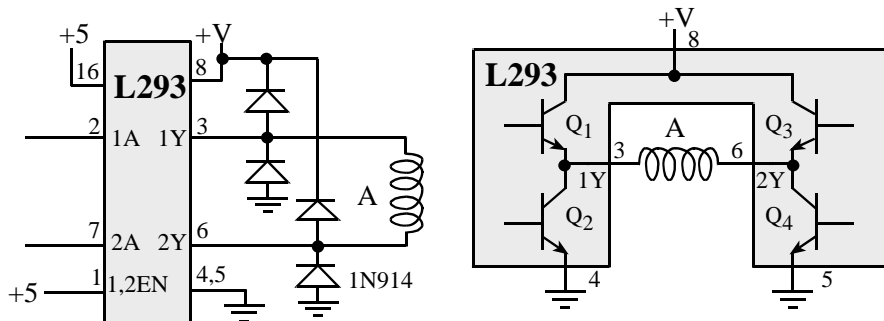
Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

36

H-bridge Interface

- Both directions (forward & backward)
- $V_{OH} = +V - 1.4$, $V_{OL} = 1.2$



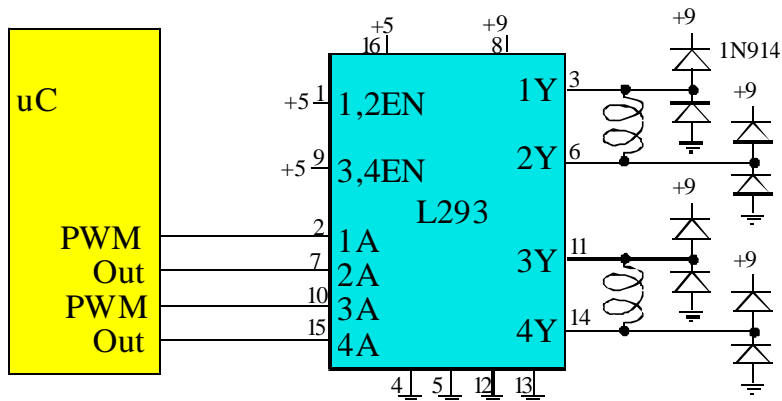
Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

37

H-bridge Interface (V1)

- PWM controls power
- Out controls direction



Lecture 9

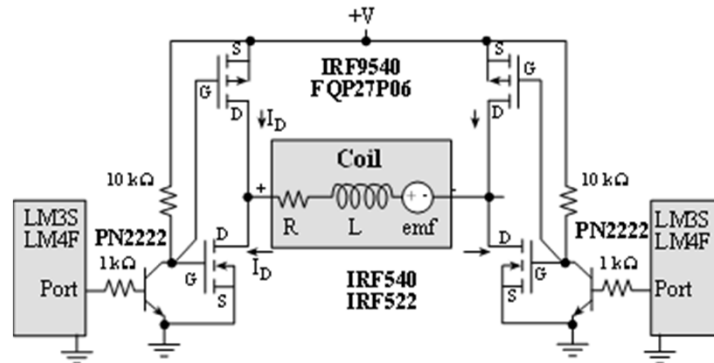
J. Valvano, A. Gerstlauer
EE445M/EE380L.12

See L6203.pdf

38

H-bridge Interface (V2)

- One Port is PWM controlling power
- Other port controls direction



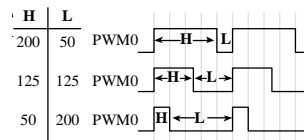
Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

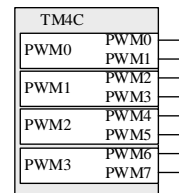
39

Pulse Width Modulation (PWM)

- Generate output waveform
 - Period = High + Low
 - Duty cycle = High / Period



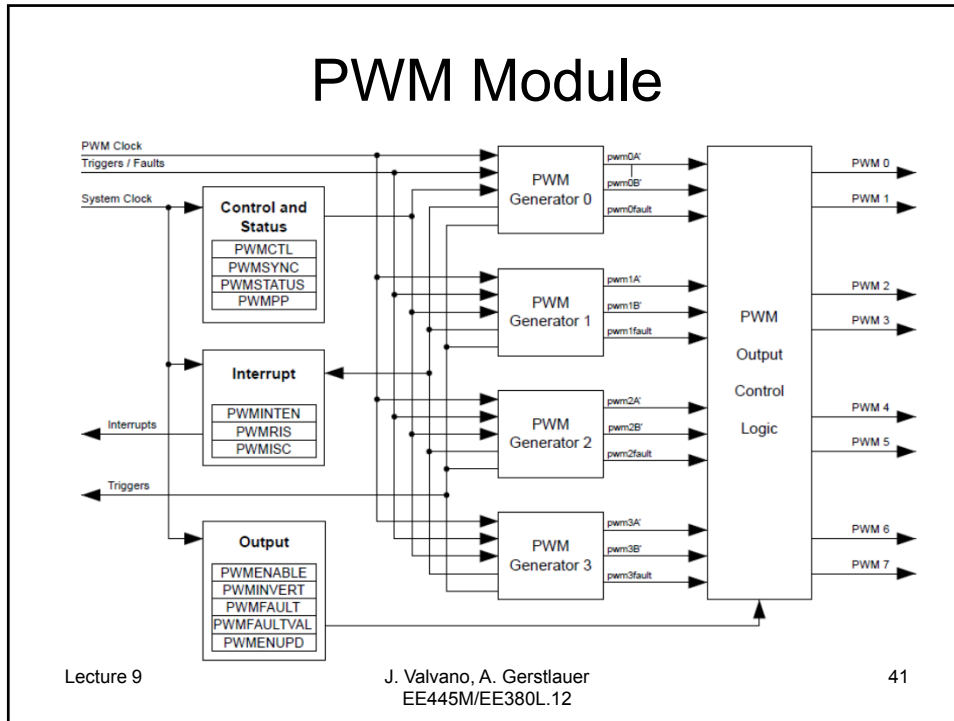
- PWM generators
 - TM4C123: 2 modules
 - 4 generators per module
 - 2 PWM signals per generator



Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

40



TM4C123 Alternate Function

IO	Ain	0	1	2	3	4	5	6	7	8	9	14
PA0		Port	U0Rx									
PA1		Port	U0Tx									
PA2		Port		SSI0Clk								
PA3		Port		SSI0Fss								
PA4		Port		SSI0Rx								
PA5		Port		SSI0Tx								
PA6		Port			LC1SCL		M1PWM2					
PA7		Port			LC1SDA		M1PWM3					
PB0	USB0ID	Port	U1Rx							T2CCP0		
PB1	USB0VBUS	Port	U1Tx							T2CCP1		
PB2		Port			LC0SCL					T3CCP0		
PB3		Port			LC0SDA					T3CCP1		
PB4	Ain10	Port		SSI2Clk		M0PWM2				T1CCP0	CAN0Rx	
PB5	Ain11	Port		SSI2Fss		M0PWM3				T1CCP1	CAN0Tx	
PB6		Port		SSI2Rx		M0PWM0				T0CCP0		
PB7		Port		SSI2Tx		M0PWM1				T0CCP1		
PC4	C1-	Port	U4Rx	U1Rx		M0PWM6		IDX1	WT0CCP0	U1RTS		
PC5	C1+	Port	U4Tx	U1Tx		M0PWM7		PhA1	WT0CCP1	U1CTS		
PC6	C0+	Port	U3Rx					PhB1	WT1CCP0	USB0open		
PC7	C0-	Port	U3Tx						WT1CCP1	USB0pft		
PD0	Ain7	Port	SSI3Clk	SSI1Clk	LC3SCL	M0PWM6	M1PWM0			WT2CCP0		
PD1	Ain6	Port	SSI3Fss	SSI1Fss	LC3SDA	M0PWM7	M1PWM1			WT2CCP1		
PD2	Ain5	Port	SSI3Rx	SSI1Rx						WT3CCP0	USB0open	
PD3	Ain4	Port	SSI3Tx	SSI1Tx		M0Fault0			IDX0	WT3CCP1	USB0pft	
PD4	USB0DM	Port	U6Rx							WT4CCP0		
PD5	USB0DP	Port	U6Tx							WT4CCP1		
PD6		Port	U2Rx			M0Fault0		PhA0	WT5CCP0			
PD7		Port	U2Tx					PhB0	WT5CCP1	NMI		
PE0	Ain3	Port	U7Rx									
PE1	Ain2	Port	U7Tx									
PE2	Ain1	Port										
PE3	Ain0	Port										
PE4	Ain9	Port	U5Rx		LC2SCL	M0PWM4	M1PWM2				CAN0Rx	
PE5	Ain8	Port	U5Tx		LC2SDA	M0PWM5	M1PWM3				CAN0Tx	
PF0		Port	U1RTS	SSI1Rx	CAN0Rx		M1PWM4	PhA0	T0CCP0	NMI	C0o	TRD1
PF1		Port	U1CTS	SSI1Tx			M1PWM5	PhB0	T0CCP1		C1o	TRD1
PF2		Port		SSI1Clk		M0Fault0	M1PWM6			T1CCP0		TRD0
PF3		Port		SSI1Fss	CAN0Tx		M1PWM7			T1CCP1		TRCLK
PF4		Port					M1Fault0	IDX0	T2CCP0	USB0open		

PWM Channels

- Use PWM channel
 - Choose PWM outputs
 - Runs at 16-bit precision
 - Fix the period (10 times faster than time constant)
 - Prescaled clock determines resolution
 - high+low sets the precision
 - Choose as large as possible (prescale as low as possible)
- Example
 - 1 ms period, bus clock = 80 MHz
 - Prescale divide by 2, so clocks at 40 MHz, i.e. 25ns
 - high+low= 40000
 - Precision is 40000 alternatives or 16 bits
 - Duty cycle range is 0 to 100%
 - Duty cycle resolution is $100\%/40000 = 0.0025\%$

Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

43

16-Bit PWM Output

```
// period is 16-bit number of PWM clock cycles in one period (3<=period)
// duty is number of PWM clock cycles output is high (2<=duty<=period-1)
// PWM clock rate = processor clock rate/SYSCTL_RCC_PWMDIV
// = BusClock/2 (in this example)
void PWM0_Init(uint16_t period, uint16_t duty){
    volatile uint32_t delay;
    SYSCTL_RCGCPWM_R |= 0x0001; // 1)activate PWM
    SYSCTL_RCGCGPIO_R |= 0x0020; // 2)activate port F
    delay = SYSCTL_RCGCGPIO_R; // allow time to finish activating
    GPIO_PORTF_AFSEL_R |= 0x01; // enable alt funct on PF0
    SYSCTL_RCC_R |= SYSCTL_RCC_USEPWMDIV; // 3) use PWM divider
    SYSCTL_RCC_R &= ~SYSCTL_RCC_PWMDIV_M; // clear PWM divider field
    SYSCTL_RCC_R |= SYSCTL_RCC_PWMDIV_2; // configure for /2 divider
    PWM0_CTL_R = 0; // 4) re-loading mode
    PWM0_GENA_R = (PWM_X_GENA_ACTCMPAD_ONE|PWM_X_GENA_ACTLOAD_ZERO);
    PWM0_LOAD_R = period - 1; // 5) cycles needed to count down to 0
    PWM0_CMPA_R = duty - 1; // 6) count value when output rises
    PWM0_CTL_R |= PWM_X_CTL_ENABLE; // 7) start PWM0
    PWM_ENABLE_R |= PWM_ENABLE_PWM0EN; // enable PWM0
}
void PWM0_Duty(uint16_t duty){
    PWM0_CMPA_R = duty - 1; // 6) count value when output rises
}

```

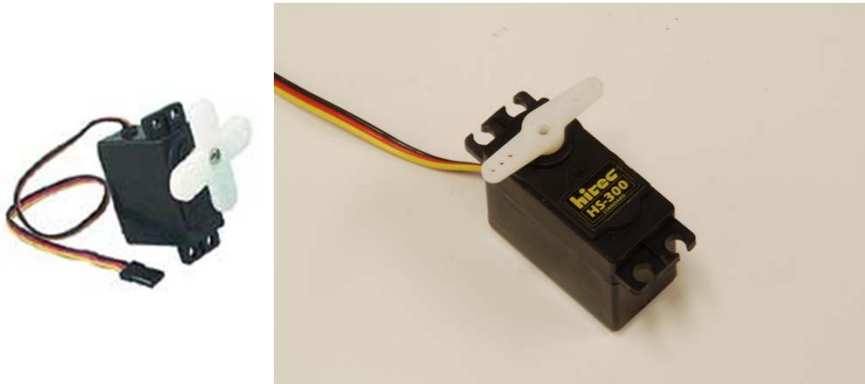
Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12PWM_4C123.zip
PWMDual_4C123.zip

44

Servo Motor

- Simple digital interface (built in controller)
- Duty cycle controls angle



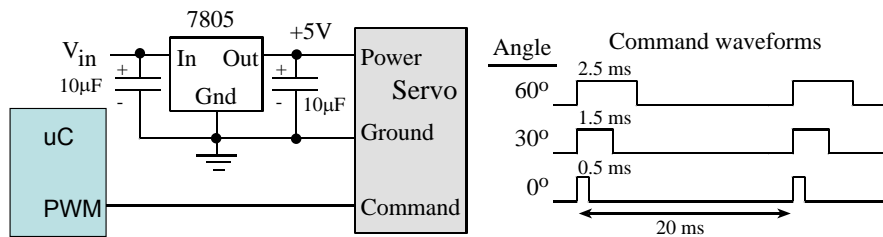
Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

45

Servo Interface

- Needs its own +5V regulator
- Duty cycle controls angle



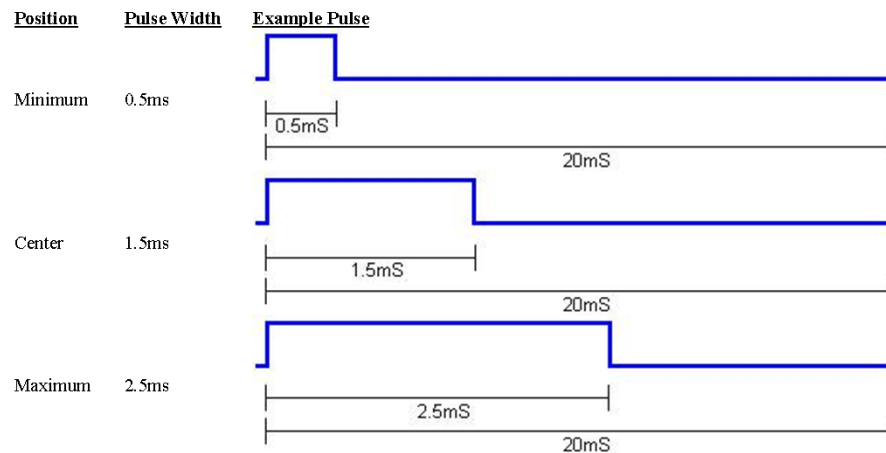
Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

46

Servo Software

- Duty cycle controls angle



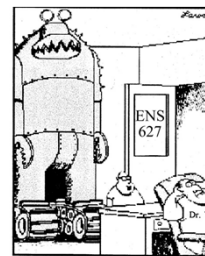
Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

47

Robot Interfacing (Lab 6)

- Sensor board
 - 4x IR sensors
 - ADC input w/ analog filter
 - 4x Distance sensors
 - Timer input, 3- or 4-pin headers
 - 2x Switches (bumper)
 - Digital input, pull-down
- Motor board
 - 2x Discrete H bridge or integrated driver chip
 - PWM output, 2 motor connectors (0.156in header)
 - 2x Servo
 - PWM output



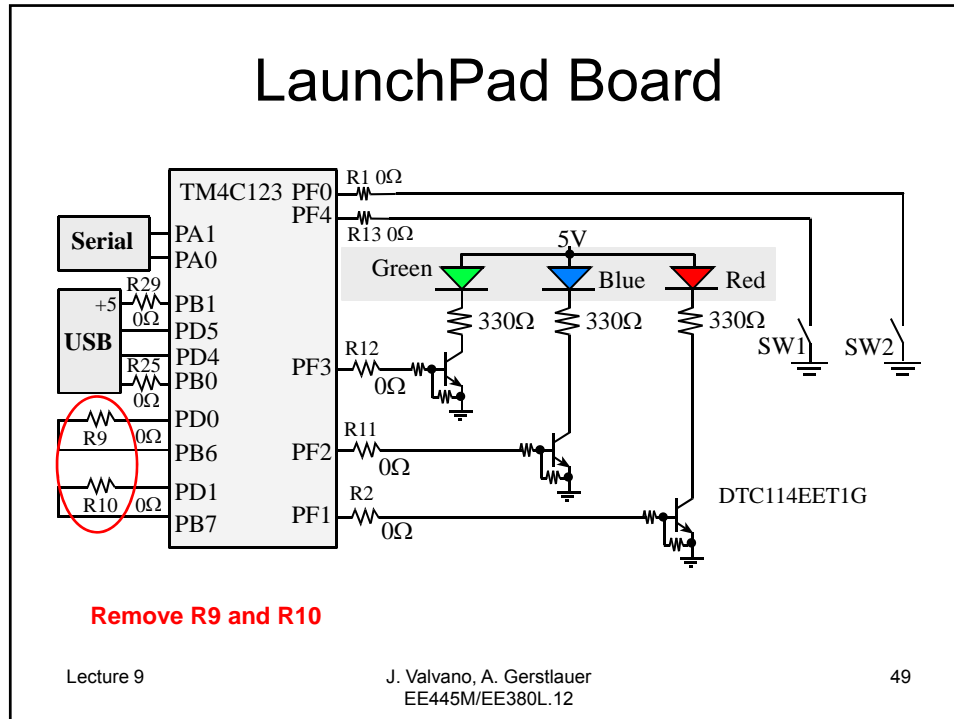
"My project's ready for grading, Dr. Big Nose...
Hey! ... I'm talking to you, squid brain!"

[Robot_Sensor_v3.pdf](#)
[Robot_Motor_v6.pdf](#)

Lecture 9

J. Valvano, A. Gerstlauer
EE445M/EE380L.12

48



Summary

- Be careful of the currents
- Sensors are noisy
- Time lag makes it unstable
- Component testing
- Visualization and control