

# ECE445M/ECE380L.12 Embedded and Real-Time Systems/ Real-Time Operating Systems

## Lecture 8: Sensing & Acting, Input Capture, PWM, Motors

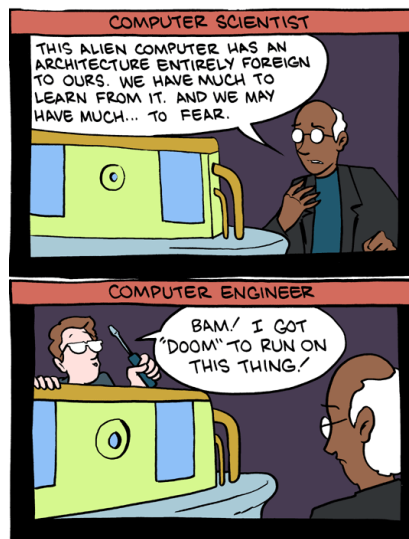
Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

1

# ECE445M vs. ECE461S/CS372

THE DIFFERENCE:



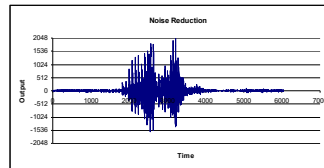
<http://www.smbc-comics.com/index.php?db=comics&id=2158#comic>

Lecture 8

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



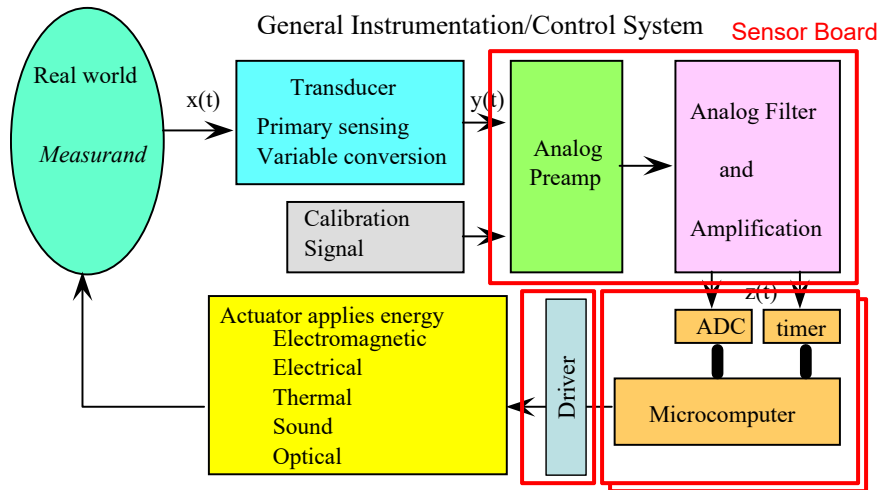
Reference book, Chapter 10

Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

3

## Instrumentation & Control



Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

Motor Board  
2x LaunchPad  
connected via CAN 4

## Sensor Board

TI OPT3101 time-of-flight sensor

PB3/SDA  
PB2/SDC  
PC4/GP1  
PC5/RST

WiFi power

ESP8266 WiFi header

Ping))) Ultrasonic sensors (3- or 4-pin)

Sharp GP2Y0A21YK analog IR sensors

Testpoints

3.3V power

5V power

Switch input (e.g. bumper)

Ping))) Ultrasonic sensors (3- or 4-pin)

5V power from motor board (vs. USB)

CAN to/from motor board

- **Reference material**
  - Schematic: [http://www.ece.utexas.edu/~gerstl/ee445m\\_s16/resources/Robot\\_Sensor\\_v3.pdf](http://www.ece.utexas.edu/~gerstl/ee445m_s16/resources/Robot_Sensor_v3.pdf)
  - PCB layout: [http://www.ece.utexas.edu/~gerstl/ee445m\\_s16/resources/sensor\\_top3.png](http://www.ece.utexas.edu/~gerstl/ee445m_s16/resources/sensor_top3.png)

Lecture 8 J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.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

Lecture 8 J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12 6

## Sharp GP2Y0A21YK

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



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

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

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



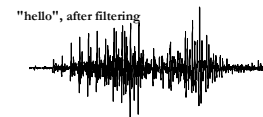
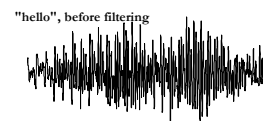
Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

7

## Filter Types

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

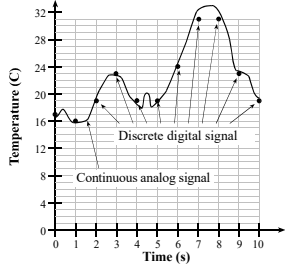


Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

8

## Sampling

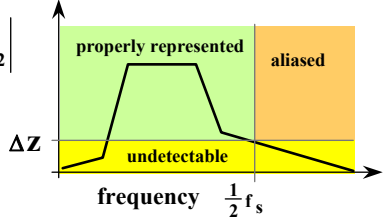


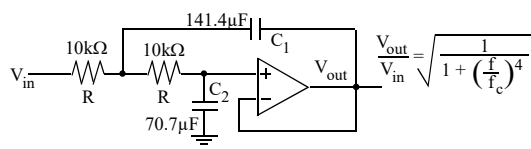
- Time & value quantizing
  - Precision  $n_z = 2^n$
- Nyquist theory
  - If sampled at  $f_s$ , digital samples only contain frequency components from 0 to  $\frac{1}{2}f_s$
  - If analog signal contains frequency components larger than  $\frac{1}{2}f_s$ , **aliasing** error
- System design
  - Choice 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 8 J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12 9

## Analog Filters

- Prevent aliasing
  - No signal  $> 0.5f_s$
- Butterworth filter
  - 2-pole low-pass filter (LPF)





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

See Sensor Board schematic

Lecture 8 7

## Digital Filters

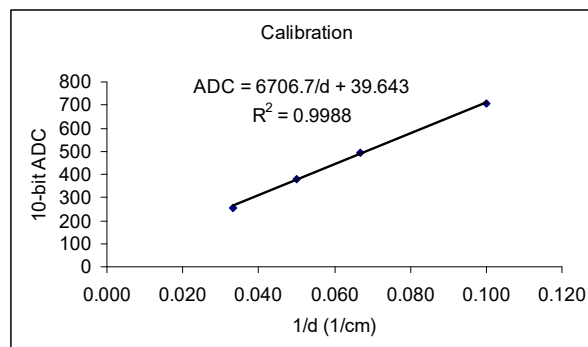
- Finite/Infinite Impulse Response (FIR/IIR)
  - Linear: HPF, LPF, BPF/notch
- Median filter
  - Non-linear: preserves edges, removes spikes

```
// 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 8

## TI OPT3101 via I<sup>2</sup>C

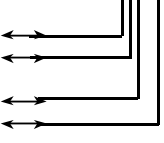
Time-of-flight (ToF)  
IR ranging



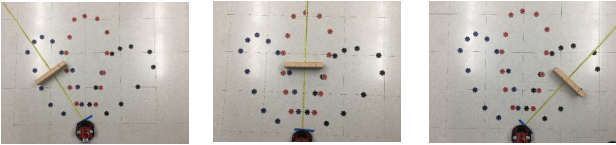
TM4C123

I2C0 PB3/I2C0SDA  
PB2/I2C0SCL

GPIO PC4  
PC5



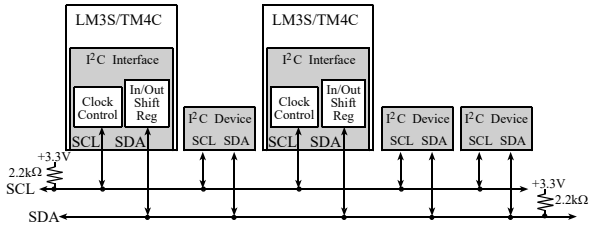
Gnd = 0V  
Vin = +3.3V  
SDA data  
SCL clock  
GP1 data\_rdy  
RST reset



The sensor performs simultaneous measurements on 3 fields of view (FOV) ±30° left/center/right beams, around 2 meters distance  
Polling- or interrupt-based on top of OS, needs digital filtering and calibration

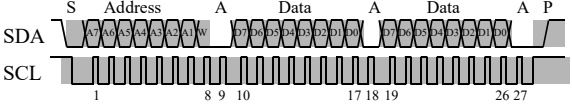
Lecture 8
J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12
13

## I<sup>2</sup>C Bus and Protocol



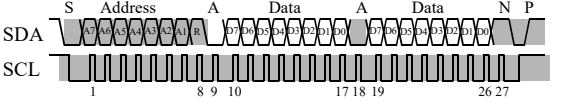
**Write data from master to slave**

S Address A Data A Data A P



**Read data from slave to master**

S Address A Data A Data N P



Lecture 8
J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12
14

# Ping Distance Sensor

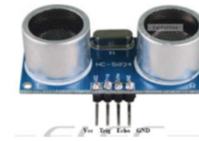
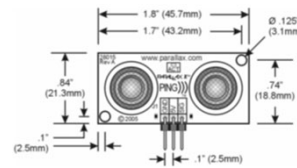
- Ultrasound transducers to measure distance

- 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 8

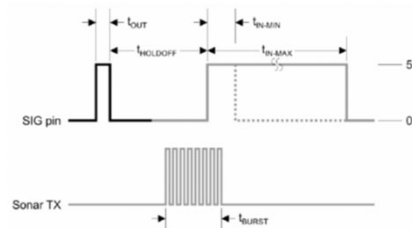
J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

15

# Ping))) Sensor

- Sample 10 times a second

- 1) Disable interrupts
- 2) Make the **SIG** pin an output
- 3) Issue a 5 $\mu$ 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 8

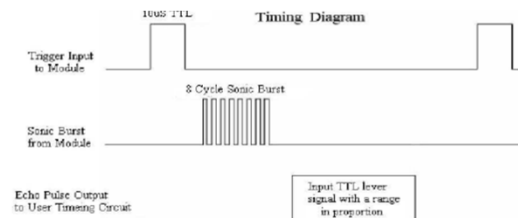
J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

16



## HCSR04 Sensor

- Sample 10 times a second
  - 1) Disable interrupts
  - 2) Issue a 10 $\mu$ 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 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

17

## Input Capture

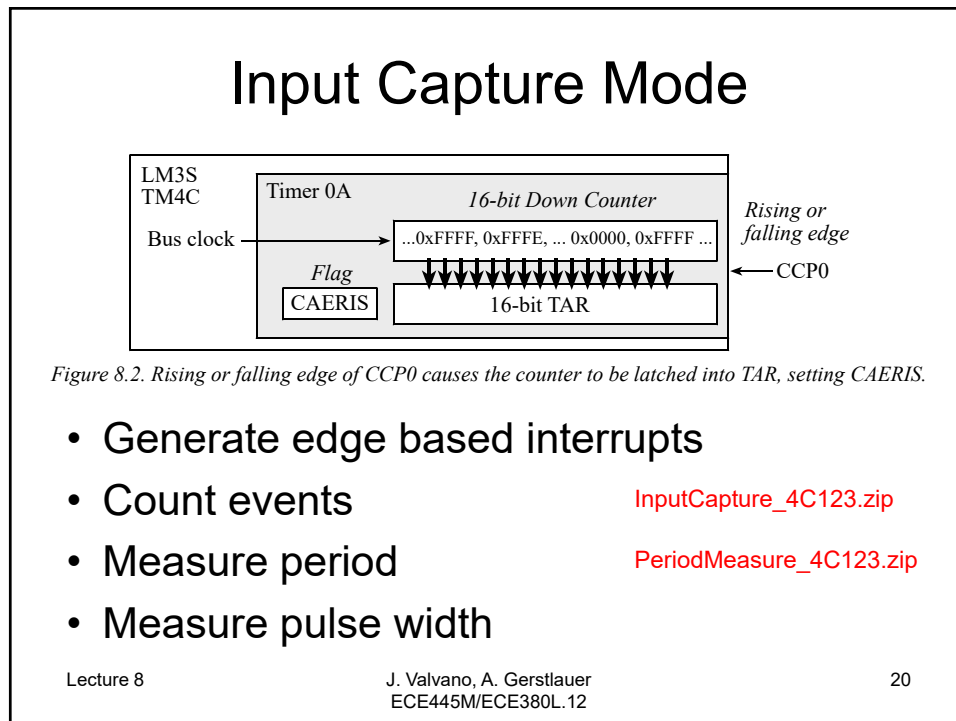
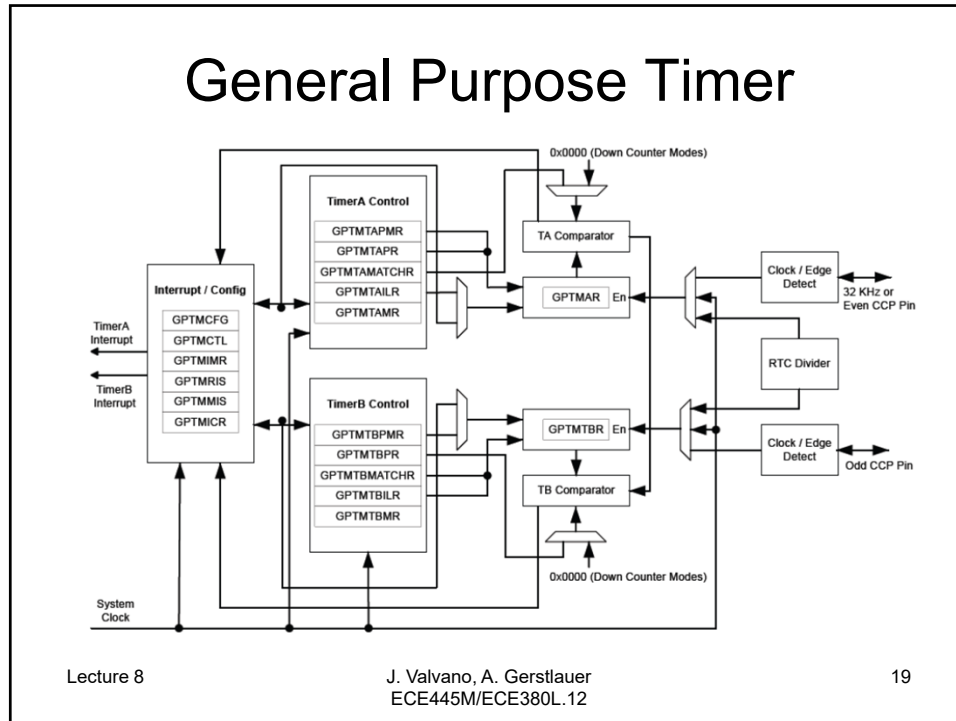
- General purpose timers
  - TM4C123: 6 GP timers (Timer 0...Timer 5)
  - CCPx pins used for input capture
- Input edge time (input capture) mode
  - Detect rising/falling input edges
  - Make time measurements on input signals

[See book Section 8.1](#)

Lecture 8

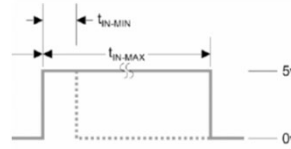
J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

18



## 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  $d = c * t_{IN}/2$

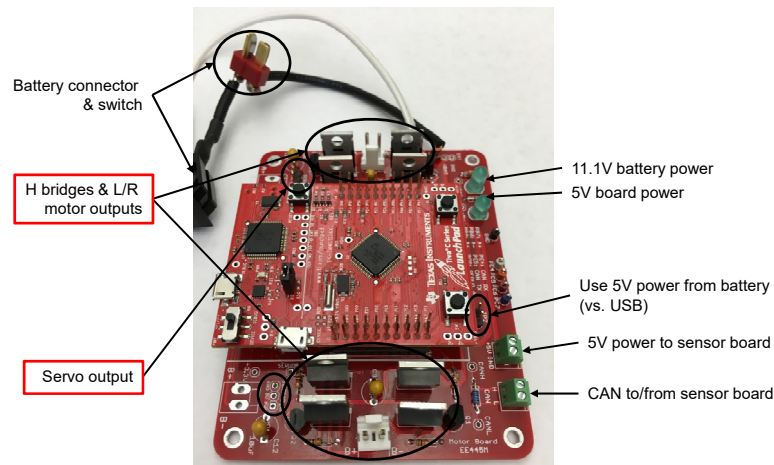


Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

21

## Motor Board



- Reference material
  - Schematic: [http://www.ece.utexas.edu/~gerst/ee445m\\_s18/resources/Robot\\_Motor\\_v6.pdf](http://www.ece.utexas.edu/~gerst/ee445m_s18/resources/Robot_Motor_v6.pdf)
  - PCB layout: [http://www.ece.utexas.edu/~gerst/ee445m\\_s18/resources/Robot\\_Motor\\_v6\\_top.png](http://www.ece.utexas.edu/~gerst/ee445m_s18/resources/Robot_Motor_v6_top.png)

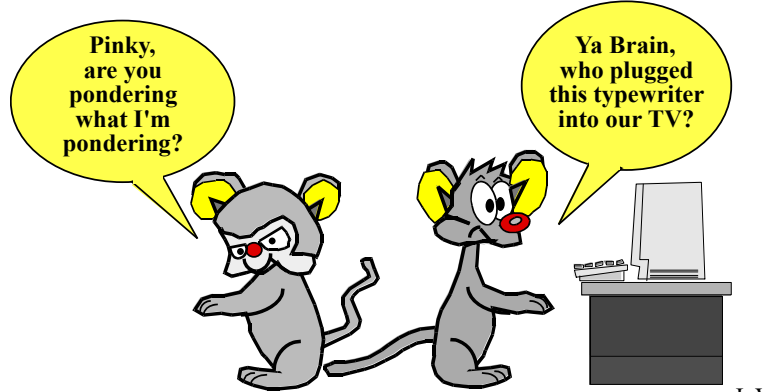
Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

22

# Motor Interfacing

- Motor physics
- Transistor-level interface



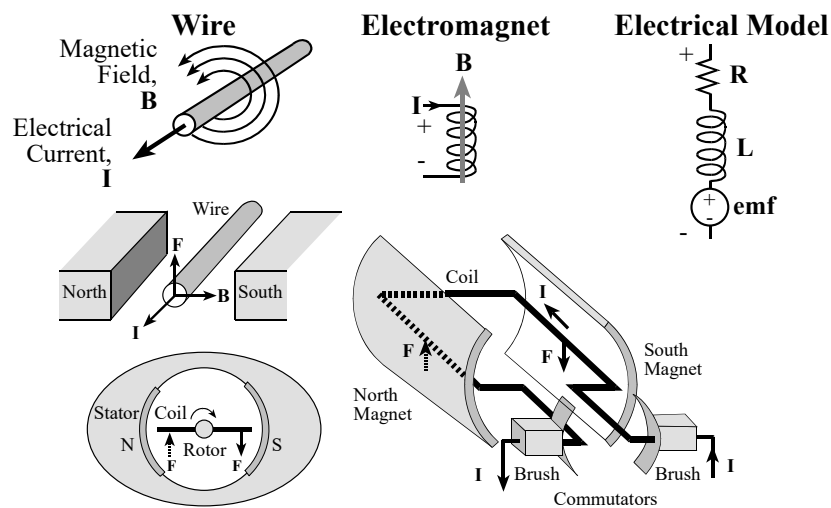
Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

JWV

23

# Motor Physics



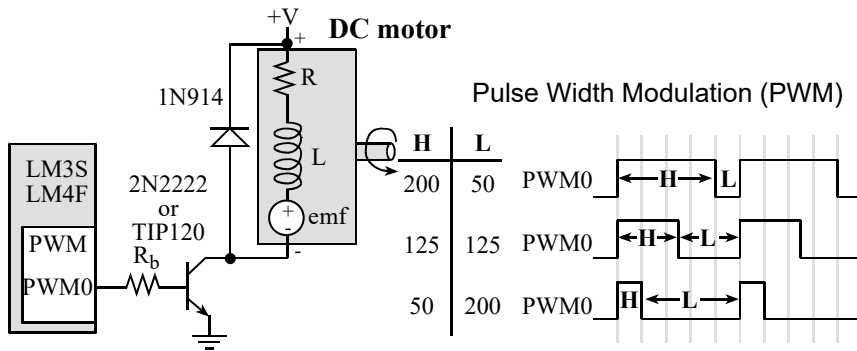
Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

24

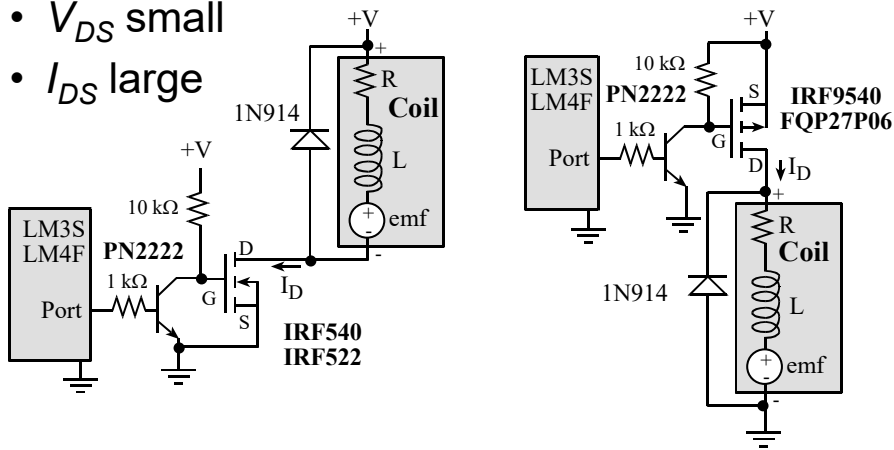
# Motor Interface

- 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



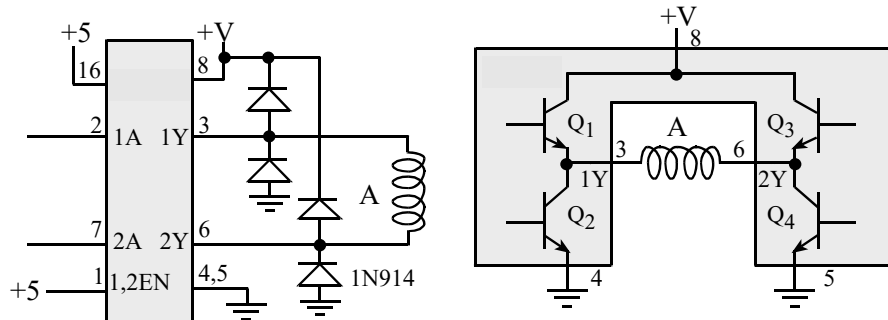
Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

26

# H-Bridge Interface

- Both directions (forward & backward)
  - One port is PWM controlling power
  - Other port controls direction



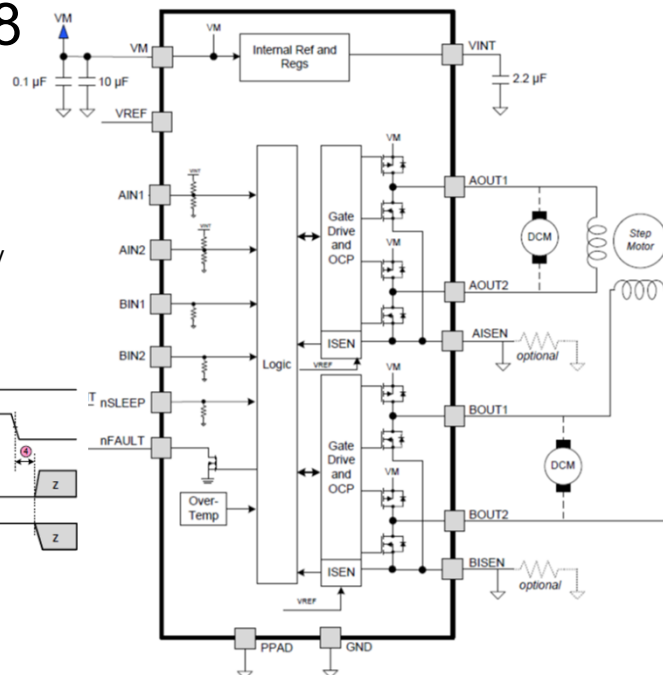
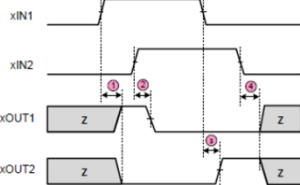
Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

27

# DRV8838

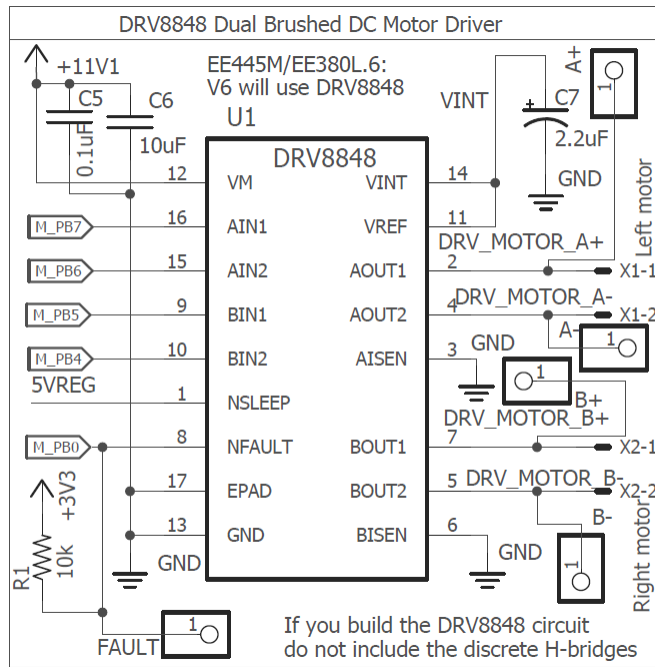
IN1	IN2	Mode
0	0	Coast
1	0	One way
0	1	Other way
1	1	Stop



Lecture 8

# H Bridge

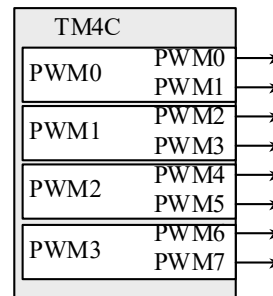
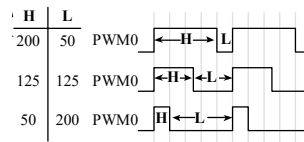
- PWM
- Direction



Lecture 8

# Pulse Width Modulation (PWM)

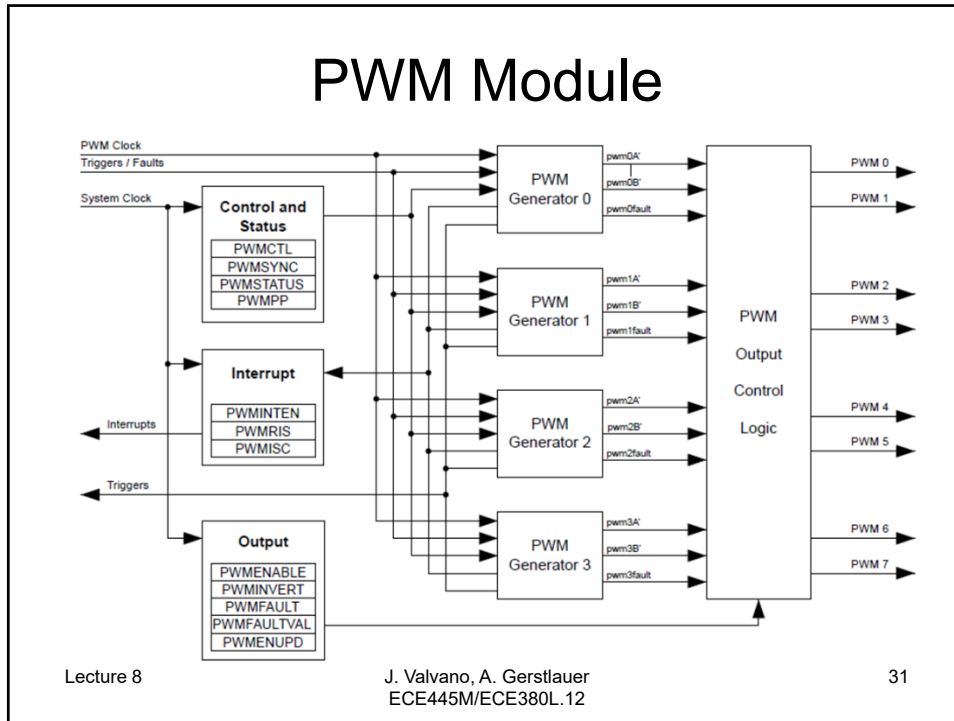
- Generate output waveform
  - Period = High + Low
  - Duty cycle = High / Period
- PWM generators
  - TM4C123: 2 modules
    - 4 generators per module
    - 2 PWM signals per generator



Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

30



## TM4C123 Alternate Function

IO	Ain	0	1	2	3	4	5	6	7	8	9	14
PA0		Port	U0Rx									CAN1Rx
PA1		Port	U0Tx									CAN1Tx
PA2		Port		SSI0Clk								
PA3		Port		SSI0Fss								
PA4		Port		SSI0Rx								
PA5		Port		SSI0Tx								
PA6		Port			I <sub>C</sub> 1SCL		M1PWM2					
PA7		Port			I <sub>C</sub> 1SDA		M1PWM3					
PB0	USB0ID	Port	U1Rx						T2CCP0			
PB1	USB0VBUS	Port	U1Tx						T2CCP1			
PB2		Port			I <sub>C</sub> 0SCL				T3CCP0			
PB3		Port			I <sub>C</sub> 0SDA				T3CCP1			
PB4	Ain10	Port		SSI2Clk		M0PWM2				CAN0Rx		
PB5	Ain11	Port		SSI2Fss		M0PWM3				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	I <sub>C</sub> 3SCL	M0PWM6	M1PWM0			WT2CCP0		
PD1	Ain6	Port	SSI3Fss	SSI1Fss	I <sub>C</sub> 3SDA	M0PWM7	M1PWM1			WT2CCP1		
PD2	Ain5	Port	SSI3Rx	SSI1Rx		M0Fault0				WT3CCP0	USB0open	
PD3	Ain4	Port	SSI3Tx	SSI1Tx				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		I <sub>C</sub> 2SCL	M0PWM4	M1PWM2			CAN0Rx		
PE5	Ain8	Port	U5Tx		I <sub>C</sub> 2SDA	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 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

33

## 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_PWMMDIV
// = 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_USEPWMMDIV; // 3) use PWM divider
    SYSCTL_RCC_R &= ~SYSCTL_RCC_PWMMDIV_M; // clear PWM divider field
    SYSCTL_RCC_R += SYSCTL_RCC_PWMMDIV_2; // configure for /2 divider
    PWM_0_CTL_R = 0; // 4) re-loading mode
    PWM_0_GENA_R = (PWM_X_GENA_ACTCMPAD_ONE|PWM_X_GENA_ACTLOAD_ZERO);
    PWM_0_LOAD_R = period - 1; // 5) cycles needed to count down to 0
    PWM_0_CMPA_R = duty - 1; // 6) count value when output rises
    PWM_0_CTL_R |= PWM_X_CTL_ENABLE; // 7) start PWM0
    PWM_ENABLE_R |= PWM_ENABLE_PWM0EN; // enable PWM0
}
void PWM0_Duty(uint16_t duty){
    PWM_0_CMPA_R = duty - 1; // 6) count value when output rises
}

```

Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12PWM\_4C123.zip  
PWMDual\_4C123.zip

34

## Motor Driver Software

```
// change duty cycle of right motor (PB7,PB6)
// Inputs: period was set in call to Right_InitDRV8848
//   duty is in 800 ns units
// duty is number of PWM clock cycles output is high (2<=duty<=period-1)
// Must call Right_InitDRV8848
// - once initially before calling Right_DutyDRV8848
// - to change direction
// Motor board version 6 with DRV8848 dual H-bridge
void DRV8848_RightDuty(uint16_t duty){
  if(RightDirection == FORWARD){
    Right_Duty(RightPeriod-duty, 1); // PB6 is negative logic duty, PB7 is 1
  }else{
    Right_DutyB(RightPeriod-duty, 1); // PB7 is negative logic duty, PB6 is 1
  }
}
```

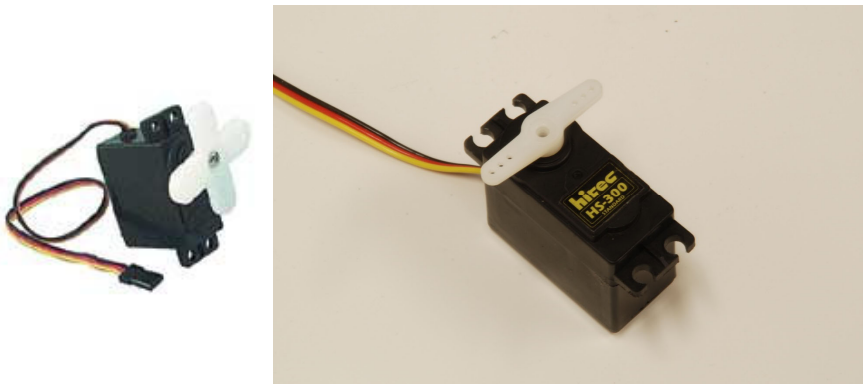
Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

35

## Servo Motor

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



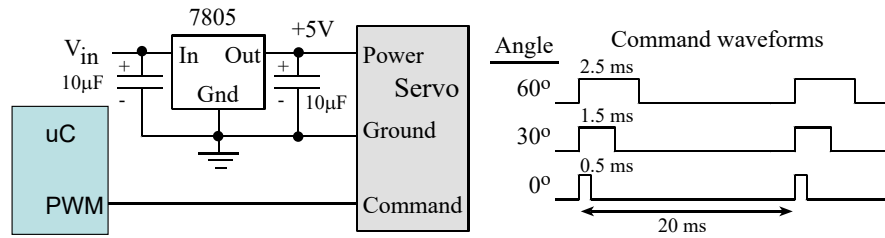
Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

36

## Servo Interface

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



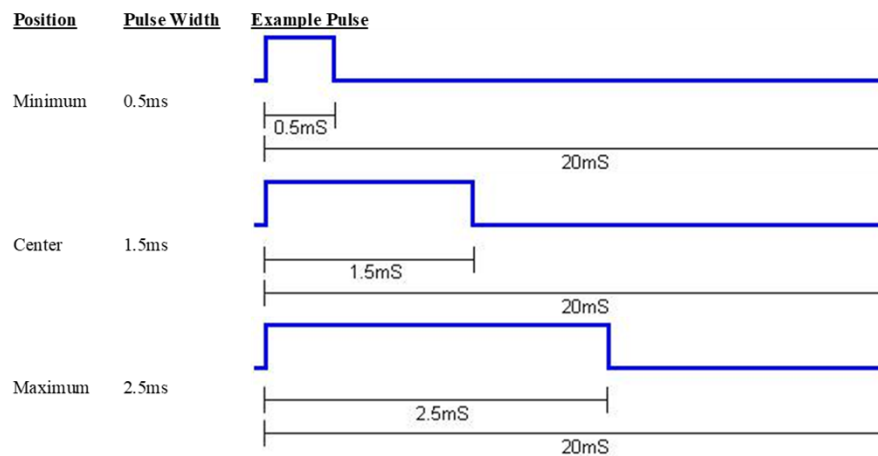
Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

37

## Servo Software

- Duty cycle controls angle



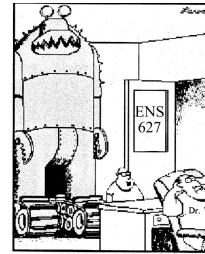
Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

38

# Robot Interfacing (Lab 6)

- Sensor board
  - 4x IR sensors
    - ADC input w/ analog filter
  - 4x Ultrasonic sensors
    - Timer input, 3- or 4-pin headers
  - 1x Time-of-flight (ToF) sensor
    - I<sup>2</sup>C port, male header pins
  - 2x Switches (bumper)
    - Digital input, pull-down
- Motor board
  - 2x Integrated H-bridge driver chip (DRV8838)
    - 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!"

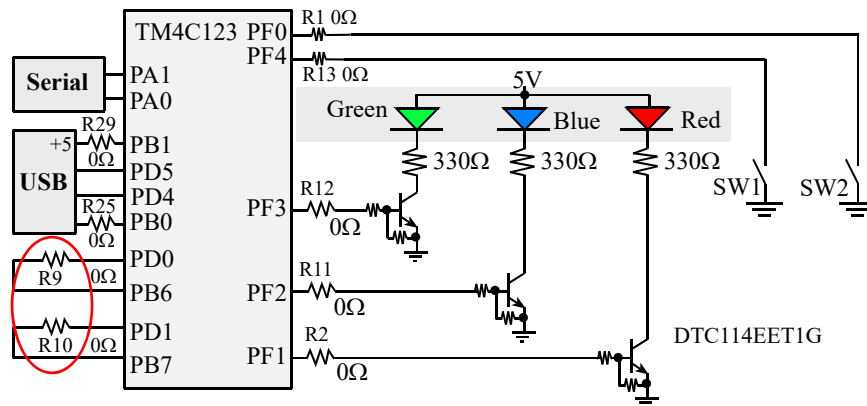
Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

[Robot\\_Sensor\\_v3.pdf](#)  
[Robot\\_Motor\\_v6.pdf](#)

39

# LaunchPad Board



- > **Remove R9 and R10**
  - PB6/PB7 needed for ultrasound J9 (Timer T0) or motor driver (PWM0/1)
  - PD0/PD1 needed for IR sensor J7/J8 (Ain6/7) or servo A/B (PWM6/7)
- > **Can not use ToF with 3<sup>rd</sup> ultrasound sensor**
  - PB2/3 used by ultrasound J11 (Timer T3) and ToF sensor (I2C0)

Lecture 8

J. Valvano, A. Gerstlauer  
ECE445M/ECE380L.12

40

## Summary

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