ECED3901 Design Methods II

LECTURE #8: SENSOR CHARACTERISTICS

Colin O'Flynn

What are we covering?

- What are sensors?
- Examples of Robot Sensors
  - Distance
    - Ultrasonic
    - Reflective (angle)
    - Reflective (time-of-flight)
  - Optical Sensors
  - Positional Sensors
    - Wheel Encoders
    - Inertial Navigation
- Sensors in Your Kit

Sensing our World

Star-Nosed Mole: https://www.youtube.com/watch?v=fio1NUxszhY

What is a Sensor

Sensor: A device which provides <u>information</u> about the physical world.

Transducers

- Convert of energy from one form to another.
- e.g. microphone and speakers convert *between* acoustic and electrical energy.
- Many transducers are bidirectional.
- Some sensors *are* transducers; almost all involve transduction.

Types of Sensors

Direct Transduction: e.g., electric and magnetic fields, mechanical strain, temperature, electromagnetic energy...

Derived quantities: e.g. distance, human presence, heading, air flow, air pressure, color...

Sensors for Robots

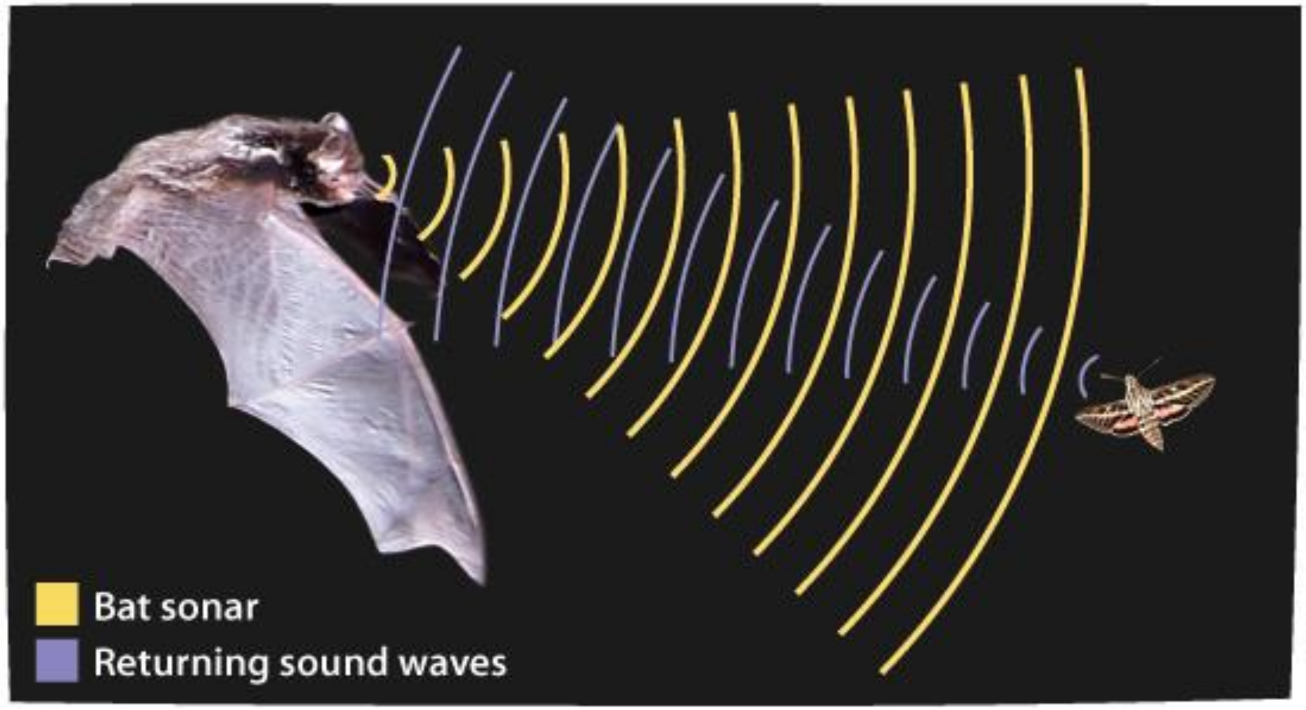
Distance Sensors

Objective – answer question about how far away object is?

Methods of achieving this:

- Reflective
  - Sound
  - Light
  - Electromagnetic
- Visual

Echo Location



…very simple version

Simple Idea…

Tx

Rx

$$d = (v * t) / 2$$

Speed of Sound

343 m/s

…but varies with temperature

Example Calculation

t = 10mS

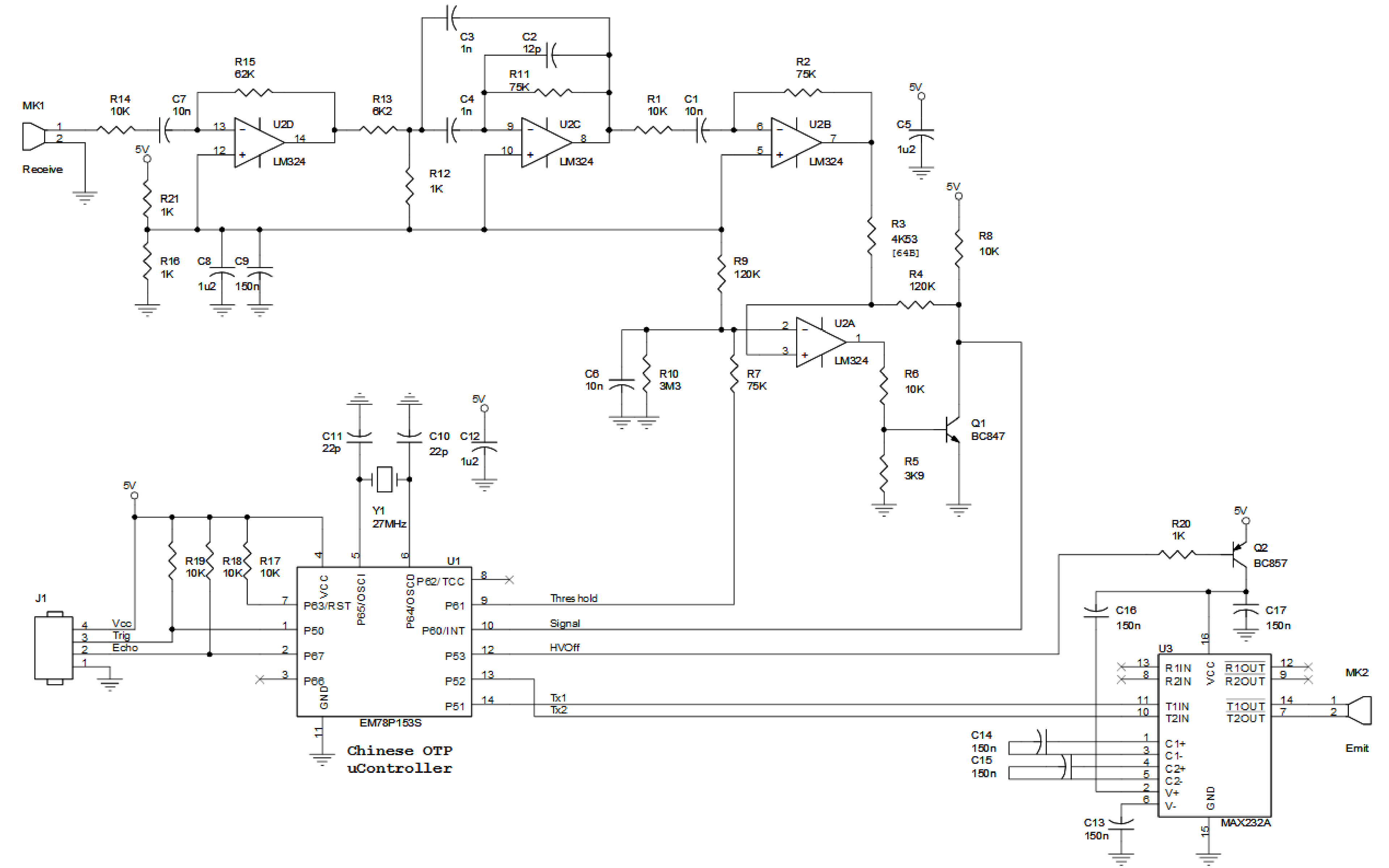
v = 343 m/S

d = 343 * 0.01 / 2 = 1.715 m

…assuming 20C. What if actually at 40C, where v=354.7 m/S

d = 354.7 * 0.01 / 2 = 1.774 m

Error = ~6cm



http://uglyduck.ath.cx/ep/archive/2014/01/Making_a_better_HC_SR04_Echo_Locator.html

HC-SR04

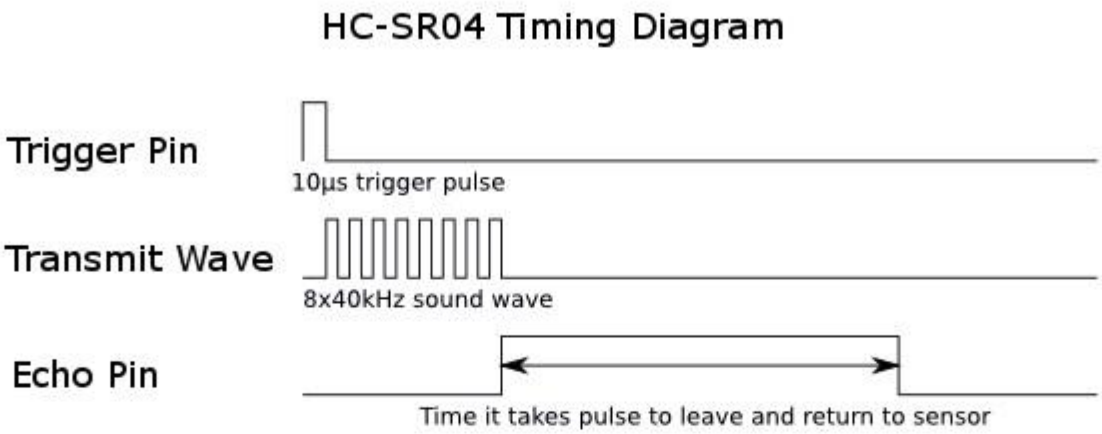
Vcc: 5V Operating Voltage pin

Trig: transmit signal pin

Echo: echo signal pin

Gnd: Ground pin

Timing Diagram



Interfacing to AVR

NOTE: HC-SR04 is *very* popular, so should be able to find lots of resources online…

For example:

1) https://electrosome.com/hc-sr04-ultrasonic-sensor-pic/

2) http://www.instructables.com/id/Talking-to-Ultrasonic-Distance-Sensor-HC-SR04-usin/

3) http://extremeelectronics.co.in/avr-tutorials/ultrasonic-rangefinder-hc-sr04-interfacing-with-atmega8/

…probably some better tutorials too!

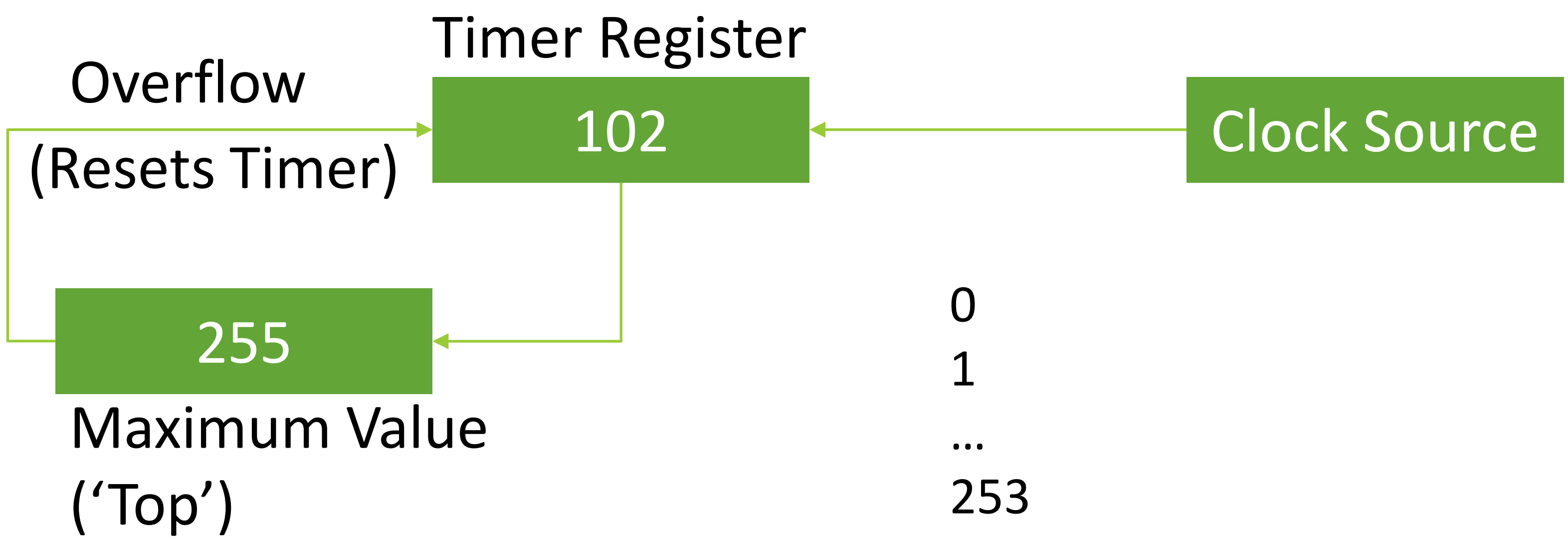
Sidenote #1: AVR Timers

What the heck are timers?

What can we use them for?

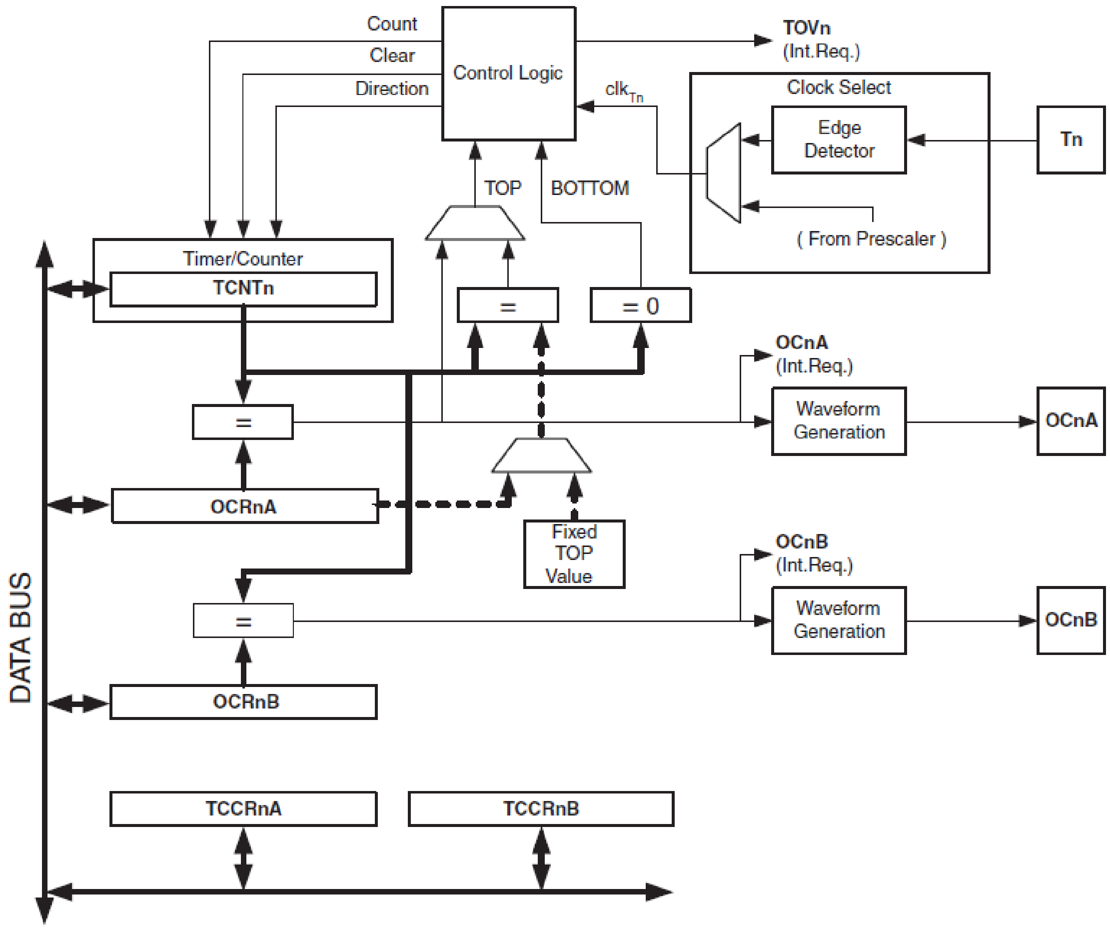
- Running code at very precise intervals
- Measuring events/time
- Generating a Pulse Width Modulation (PWM) signal
- Measuring width of external pulse

Basic Timer Architecture

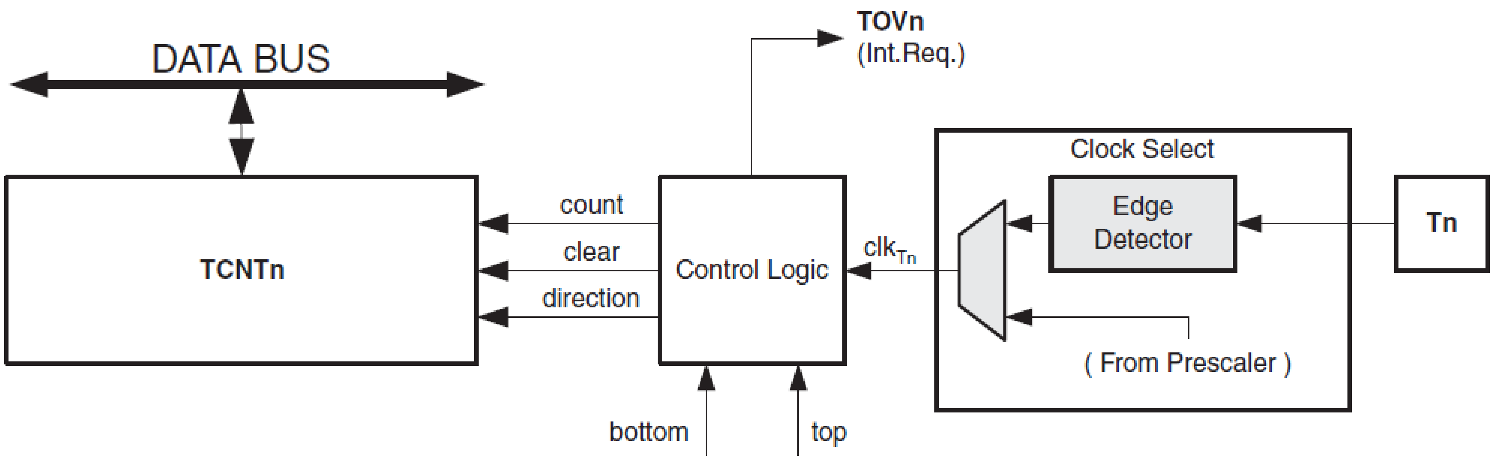


0
1
…
253
254
255
0
1
…

AVR Timer Architecture

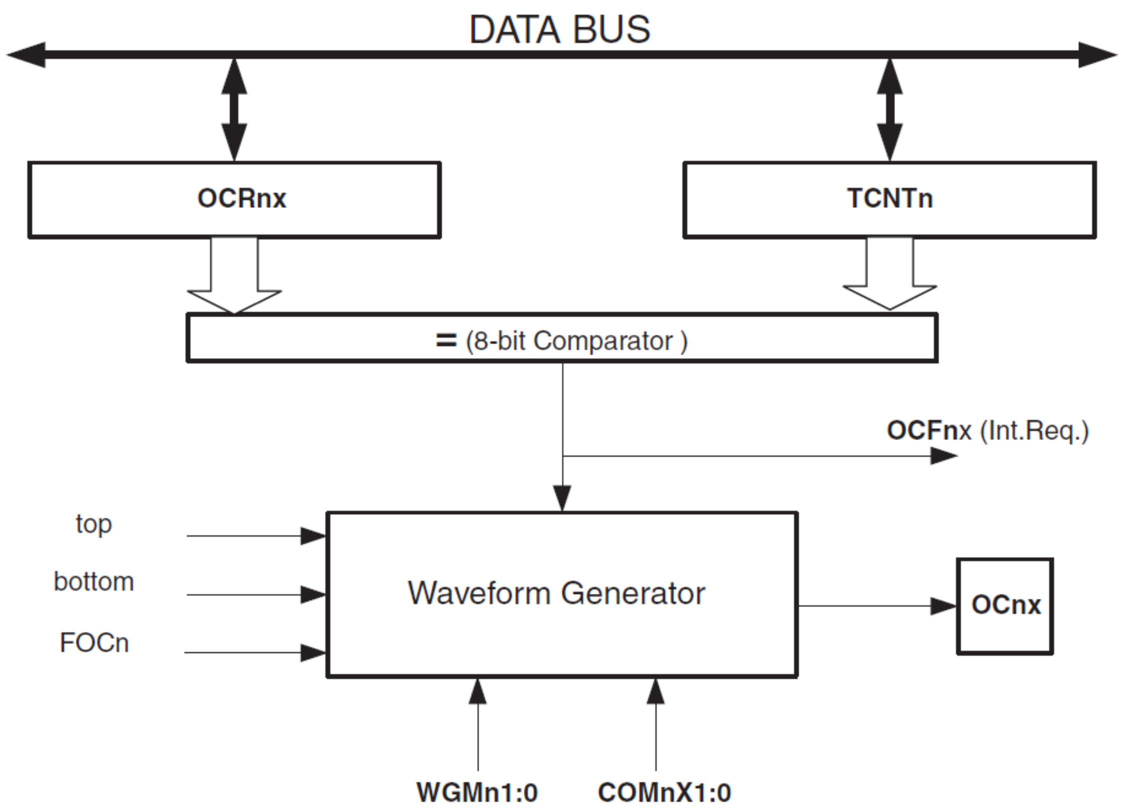


Timer Details



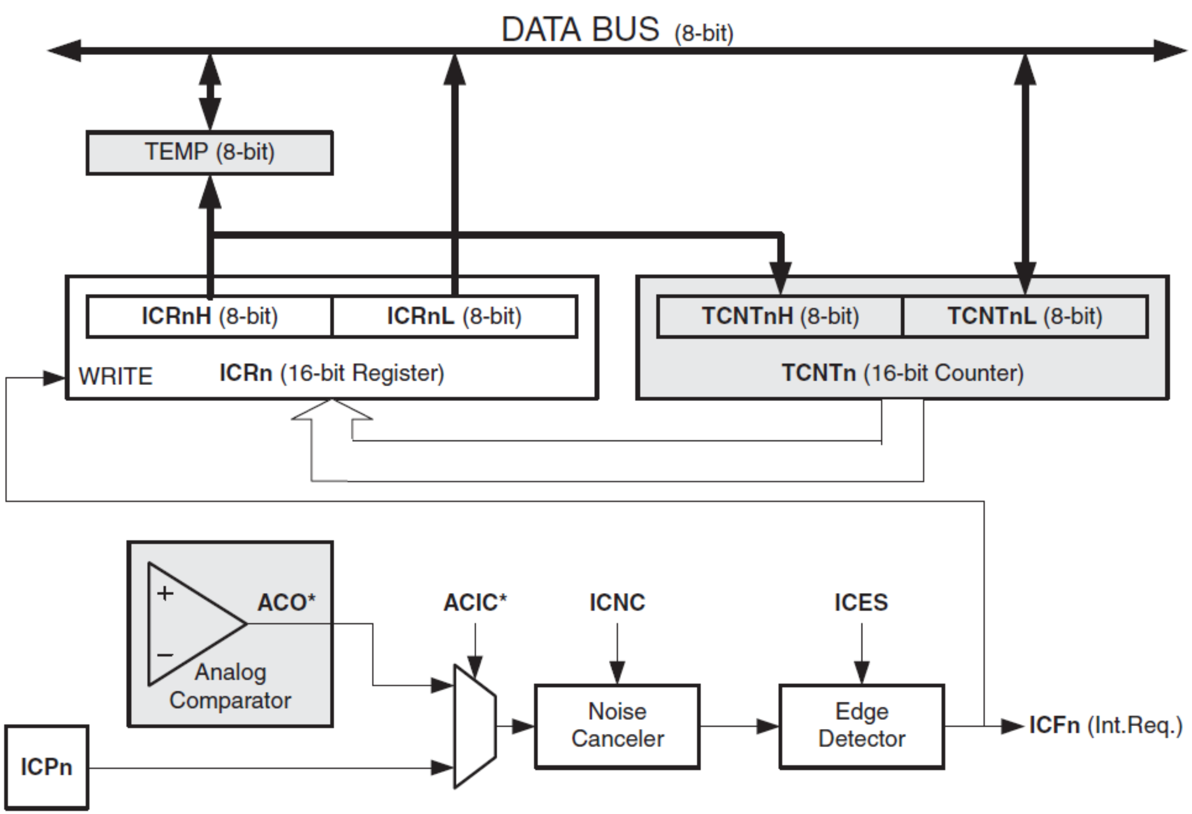
Output Compare Registers

Figure 12-3. Output Compare Unit, Block Diagram



Input Compare (*not* in Timer0)

Figure 13-3. Input Capture Unit Block Diagram



Measuring Width #1

```
TCNT1 = 0;
loop_until_bit_is_high(PINB, 1);
start_timer1();
loop_until_bit_is_low(PINB, 1);
stop_timer1();
```

delay = TCNT1;

Measuring Width #2

```
volatile uint16_t delay;

ISR(SomePinChangeInterrupt)
{
if (PINB & _BV(1))
{
    TCNT1 = 0;
    start_timer1();
}
else
{
    stop_timer1();
    delay = TCNT1;
}



int main(void)
{
    setup_timer1();
    delay = 0;

    while(delay == 0){
        continue;
    }

    printf("Delay = %d\n", delay);
}
```

Measuring Width #3

```
ISR(TIMER1_CAPT_vect)
{
   if(current_edge == 0){
         //Save timestamp
         starting_cnt = ???;

         //Switch to rising edge
         TCCR1B |= ????

         current_edge = 1;

   } else if (current_edge == 1){
         //Save timestamp
         ending_cnt = ICR1;

         //Switch to falling edge
         TCCR1B &= ~(????);

         current_edge = 2;
```

Getting Used to Timers

I highly recommend seeing ECED3204 Lab #4 (https://github.com/colinoflynn/eced3204/blob/master/labs/lab4_timer/LAB4.pdf) for some experiments with timer module!

That lab will teach you two critical things:

- How to generate PWM signals (required for moving your robot)
- How to use ICP (can be used for ultrasonic rangefinder)

Sidenote #2: AVR Interrupts

Polling vs. Interrupts

Q: Need to monitor and respond to something. How do we do it?

A: Dumb way… polling:

- → Microcontroller continually checks in if event as occurred
- → Wastes a lot of time doing this checking

A: Smart way… interrupts:

- → Microcontroller tells system to "interrupt" it when event has occurred
- → Typically this means the **Interrupt Service Routine (ISR)** is called
- → Examples of typical interrupts you can enabled:
  - → New byte received on serial port
  - → State of data line has changed
  - → Certain amount of time has elapsed

How to use Interrupts?

```
#include <avr/interrupt.h>

ISR(VECTOR_NAME)
{
    //User Code goes here
    ;
}

int main(void)
{
    //This is REQUIRED to enable interrupts
    sei();

    while(1){
        //do a bunch of stuff
        ;
    }

}
```

What happened in our program?

- Finish executing current instruction.
- Store where the next instruction to be executed is.
- Jump to Interrupt Service Routine (ISR) to handle interrupt.
- Run code in ISR.
- Return back to instruction next being executed.
- Continue on with life…

ISR Vector Names in Datasheet

9.2 Interrupt Vectors in ATmega164P/324P/644P

Table 9-1. Reset and Interrupt Vectors

Vector No.	Program Address(2)	Source	Interrupt Definition
1	$0000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$0002	INT0	External Interrupt Request 0
3	$0004	INT1	External Interrupt Request 1
4	$0006	INT2	External Interrupt Request 2
5	$0008	PCINT0	Pin Change Interrupt Request 0
6	$000A	PCINT1	Pin Change Interrupt Request 1
7	$000C	PCINT2	Pin Change Interrupt Request 2
8	$000E	PCINT3	Pin Change Interrupt Request 3
9	$0010	WDT	Watchdog Time-out Interrupt
10	$0012	TIMER2_COMPA	Timer/Counter2 Compare Match A
11	$0014	TIMER2_COMPB	Timer/Counter2 Compare Match B

Colin O’Flynn

ISR Vector names in AVR-Libc

http://www.nongnu.org/avr-libc/user-manual/group__avr__interrupts.html

INT1_vect	SIG_INTERRUPT1	External Interrupt Request 1	AT90S2313, AT90S2333, AT90S4414, AT90S4433, AT90S4434, AT90S8515, AT90S8535, AT90PWM216, AT90PWM2B, AT90PWM316, AT90PWM3B, AT90PWM3, AT90PWM2, AT90PWM1, AT90CAN128, AT90CAN32, AT90CAN64, ATmega103, ATmega128, ATmega1284P, ATmega16, ATmega161, ATmega162, ATmega163, ATmega168P, ATmega32, ATmega323, ATmega328P, ATmega32HVB, ATmega406, ATmega48P, ATmega64, ATmega8, ATmega8515, ATmega8535, ATmega88P, ATmega168, ATmega48, ATmega88, ATmega640, ATmega1280, ATmega1281, ATmega2560, ATmega2561, ATmega324P, ATmega164P, ATmega644P, ATmega644, ATmega16HVA, ATtiny2313, ATtiny28, ATtiny48, ATtiny261, ATtiny461, ATtiny861, AT90USB162, AT90USB82, AT90USB1287, AT90USB1286, AT90USB647, AT90USB646

Be Careful!

TIMER0_OVF0_vect	SIG_OVERFLOW0	Timer/Counter0 Overflow	AT90S2313, AT90S2323, AT90S2343, ATtiny22, ATtiny26
TIMER0_OVF_vect	SIG_OVERFLOW0	Timer/Counter0 Overflow	AT90S1200, AT90S2333, AT90S4414, AT90S4433, AT90S4434, AT90S8515, AT90S8535, AT90PWM216, AT90PWM2B, AT90PWM316, AT90PWM3B, AT90PWM3, AT90PWM2, AT90PWM1, AT90CAN128, AT90CAN32, AT90CAN64, ATmega103, ATmega128, ATmega1284P, ATmega16, ATmega161, ATmega162, ATmega163, ATmega165, ATmega165P, ATmega168P, ATmega169, ATmega169P, ATmega32, ATmega323, ATmega325, ATmega3250, ATmega3250P, ATmega328P, ATmega329, ATmega3290, ATmega3290P, ATmega32HVB, ATmega48P, ATmega64, ATmega645, ATmega6450, ATmega649, ATmega6490, ATmega8, ATmega8515, ATmega8535, ATmega88P, ATmega168, ATmega48, ATmega88, ATmega640, ATmega1280, ATmega1281, ATmega2560, ATmega2561, ATmega324P,

More experimentation…

ECED3204 Lab #4 also includes interrupt stuff (https://github.com/colinoflynn/eced3204/blob/master/labs/lab4_timer/LAB4.pdf)!

Note on *volatile* Keyword

The *volatile* keyword used when variable changes outside of “regular” scope.. i.e.:

```
#include <avr/interrupt.h>

int itIsDone = 0;

ISR(VECTOR_NAME)
{
    itIsDone = 1;
}

int main(void)
{
    //This is REQUIRED to enable interrupts
    sei();

    while(1){
        if(itIsDone == 1){
            break;
        }
    }

    printf("It is done!\n");

}
```

Note on *volatile* Keyword

The *volatile* keyword used when variable changes outside of “regular” scope.. i.e.:

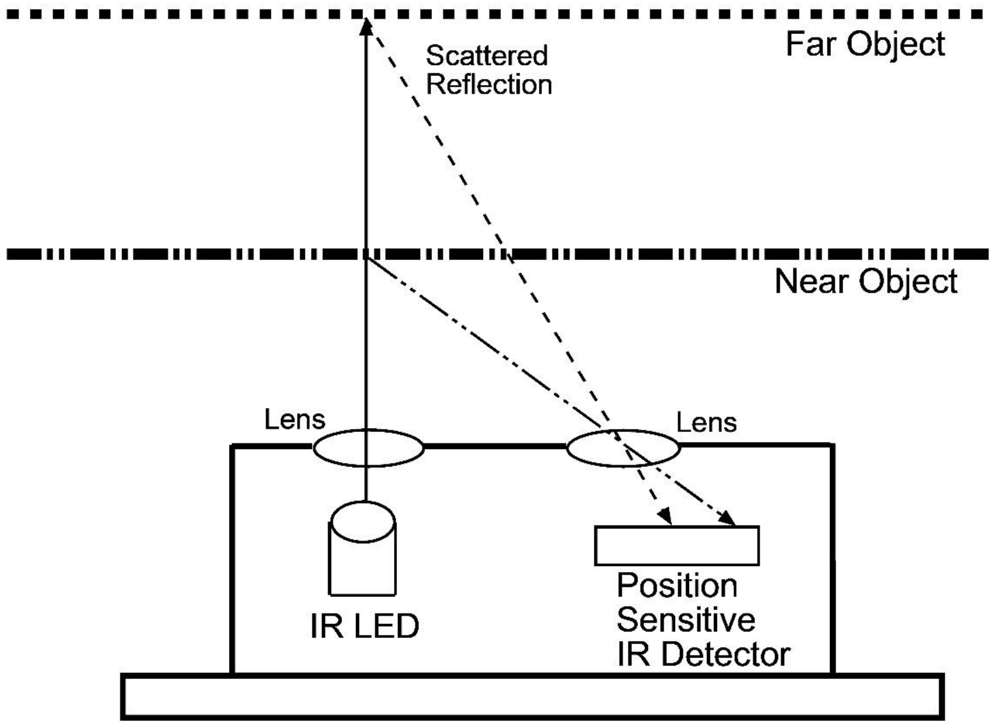
```
#include <avr/interrupt.h>

volatile int itIsDone = 0;

ISR(VECTOR_NAME)
{
    itIsDone = 1;
}

int main(void)
{
    //This is REQUIRED to enable interrupts
    sei();

    while(1){
        if(itIsDone == 1){
            break;
        }
    }

    printf("It is done!\n");

}
```

Interrupt Checklist

- ❑ Confirm I’ve enabled interrupt flags for peripheral
- ❑ Confirm interrupts globally enabled (with sei())
- ❑ Confirm I understand how to ‘clear’ interrupt flag… sometimes done automatically, sometimes not
  - This is required to avoid infinite interrupts!
- ❑ Any variables modified inside ISR that should be read by main loop are declared volatile
  - Variables inside ISR do not need to be volatile
- ❑ ISR isn’t too long!

Other Distance Sensors

IR – Reflective Angle

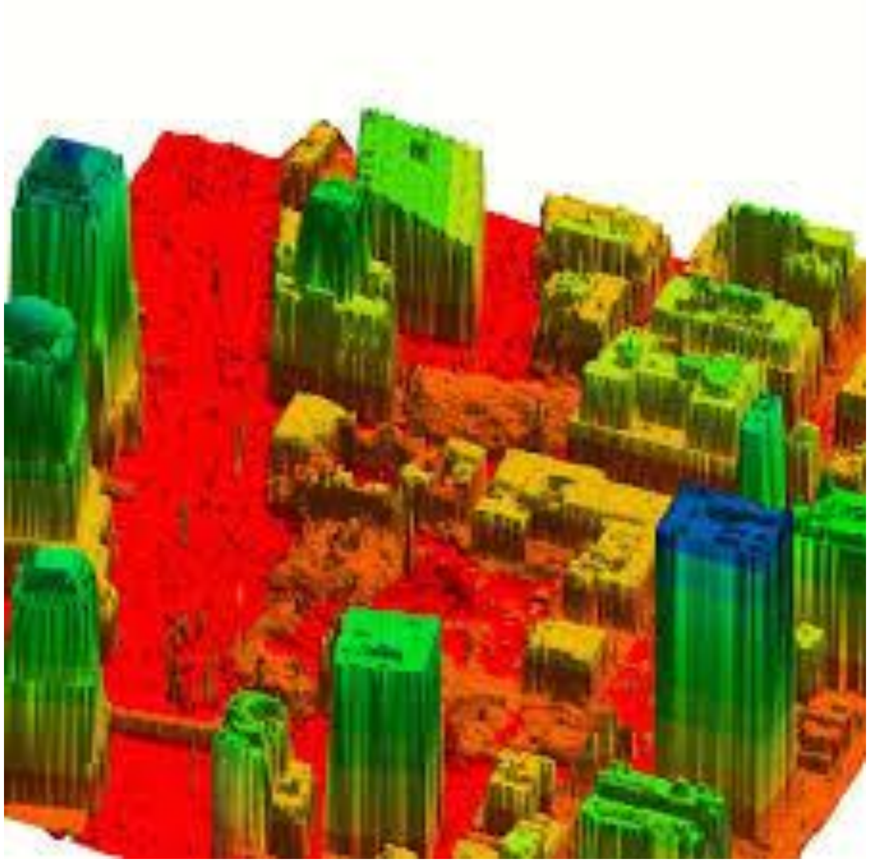


Time of Flight

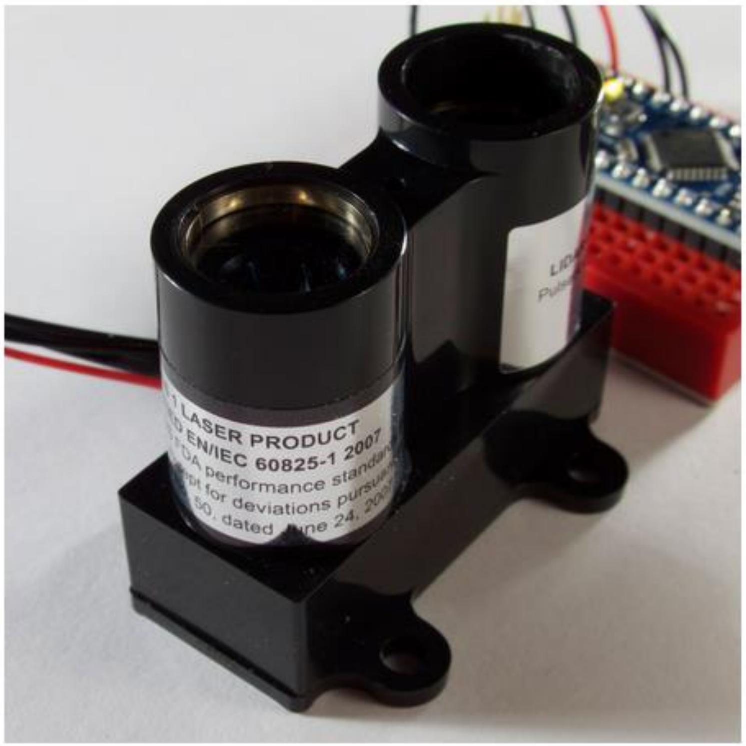
Problem with sound-based solution…

Use “Lasers”

LIDAR Maps



Cheaper LIDAR



LIDAR-Lite Laser Module

$ 89.00

QUANTITY 1

Ships within 1 week. Your card will be authorized, but not charged until we ship.

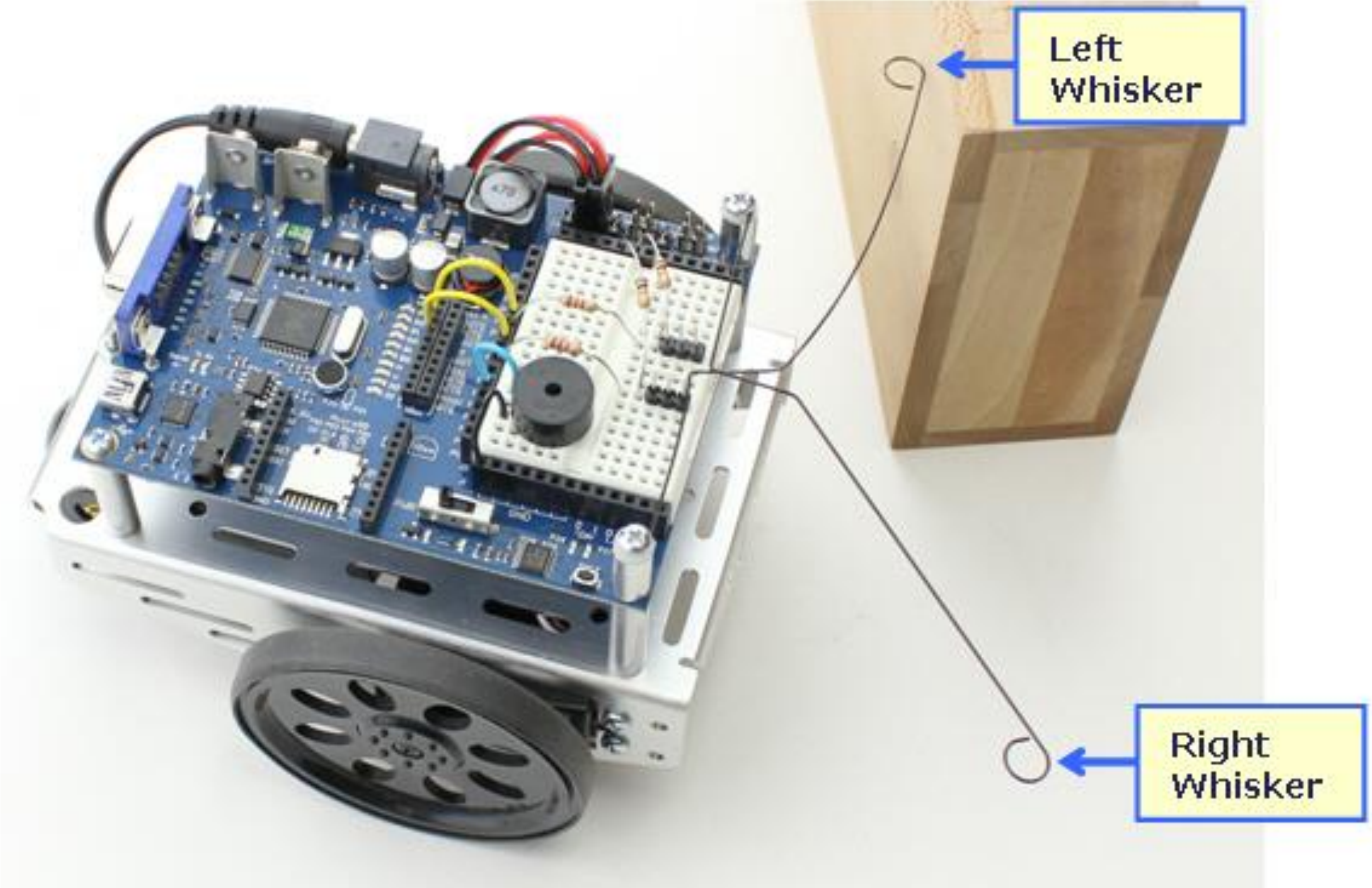
ADD TO CART

RobotShop sparkfun.

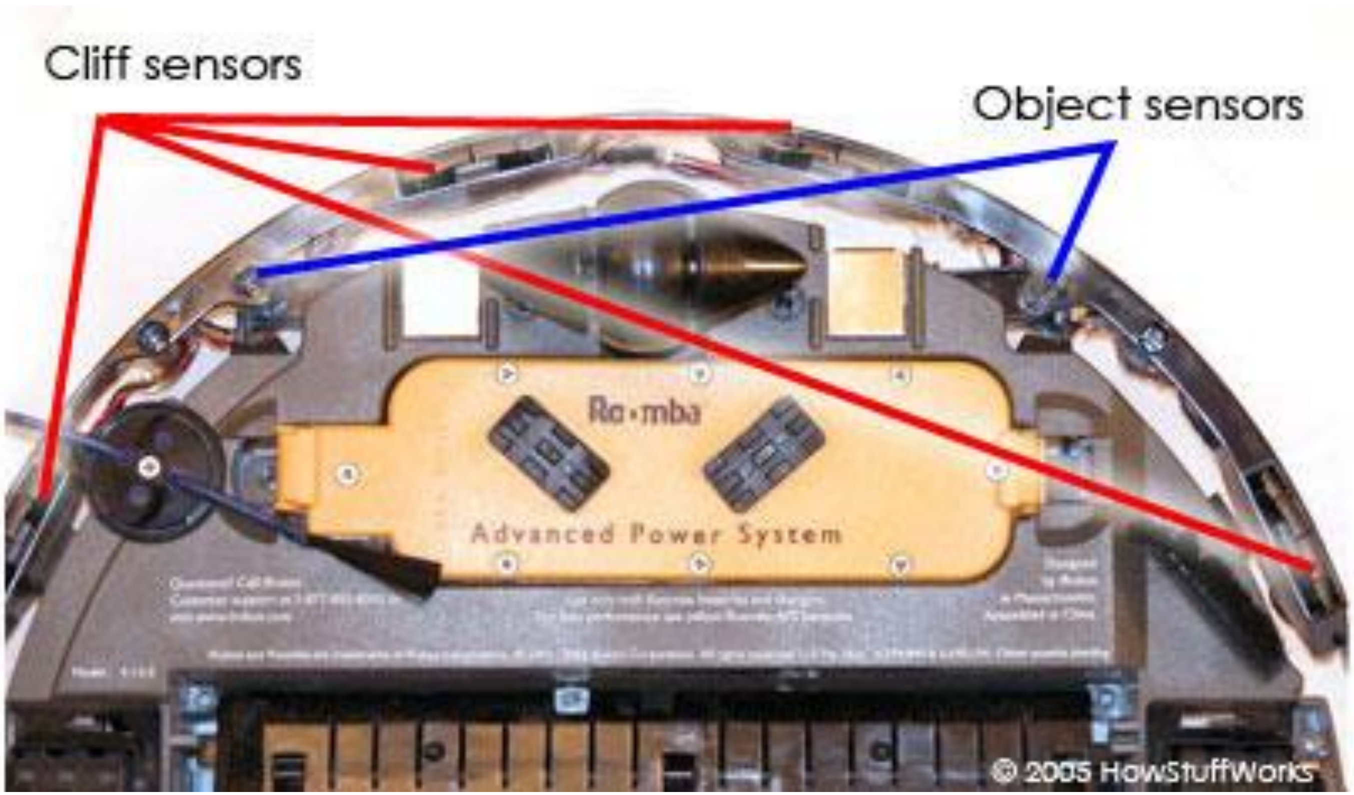
Physically Detecting Objects

Colin O'Flynn

Robot Whiskers



Robot Bump Sensors



Microswitches

Colin O'Flynn

Optical Sensors

Types of Optical Sensors

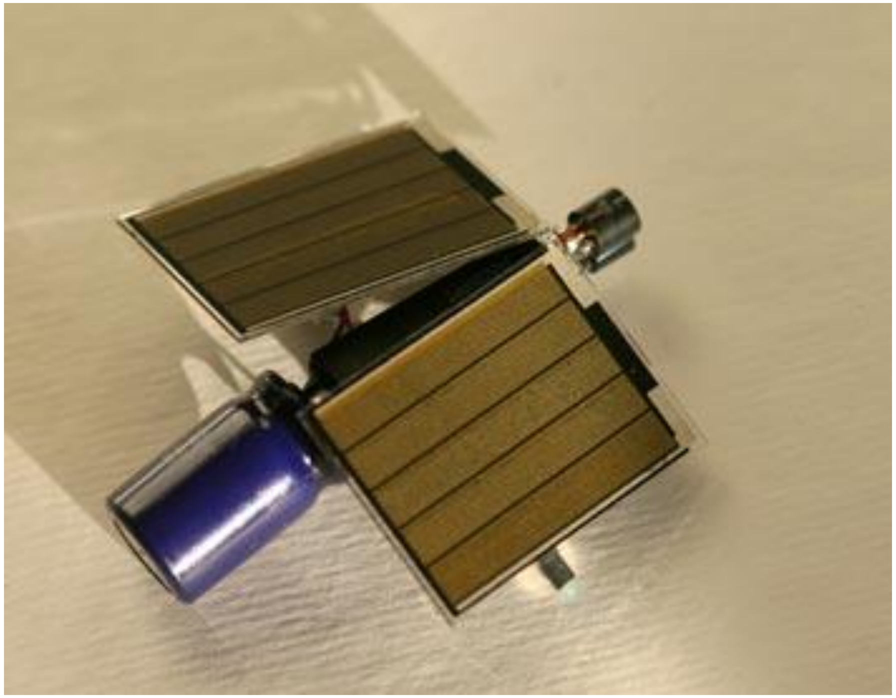
- Ambient light
- Reflective
- Transmissive
- Multi-Pixel

Ambient Light

- Detect light present in room, detect light source, etc.

Robot Applications

i.e., BEAM Robots



http://www.smfr.org/robots/

Ambient Light Sensor

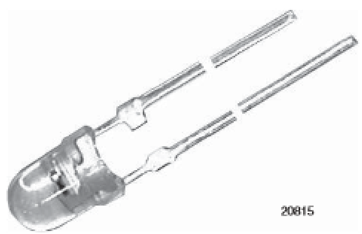
VISHAY

www.vishay.com

TEPT4400

Vishay Semiconductors

Ambient Light Sensor



FEATURES

- Package type: leaded
- Package form: T-1
- Dimensions (in mm): Ø 3
- High photo sensitivity
- Adapted to human eye responsivity
- Angle of half sensitivity: $\varphi = \pm 30°$
- Material categorization: for definitions of compliance please see www.vishay.com/doc?99912

Pb-free

e4

RoHS COMPLIANT

HALOGEN FREE

GREEN (5-2008)

DESCRIPTION

TEPT4400 ambient light sensor is a silicon NPN epitaxial planar phototransistor in a T-1 package. It is sensitive to visible light much like the human eye and has peak sensitivity at 570 nm.

APPLICATIONS

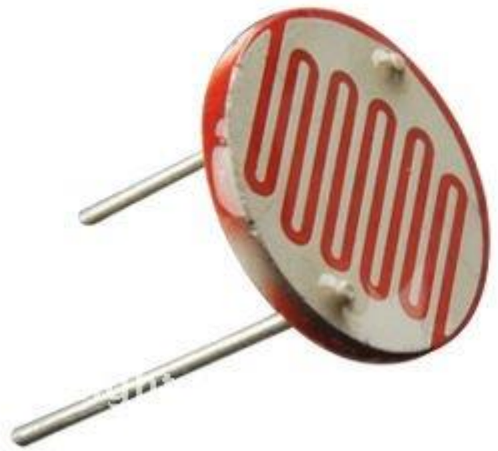
- Ambient light sensor for control of display backlight dimming in LCD displays and keypad backlighting of mobile devices and in industrial on / off-lighting operation
- Replacement of CdS photoresistors

PRODUCT SUMMARY			
COMPONENT	I_{PCE} (µA)	φ (deg)	$\lambda_{0.5}$ (nm)
TEPT4400	200	± 30	440 to 800

Note

- Test condition see table "Basic Characteristics"

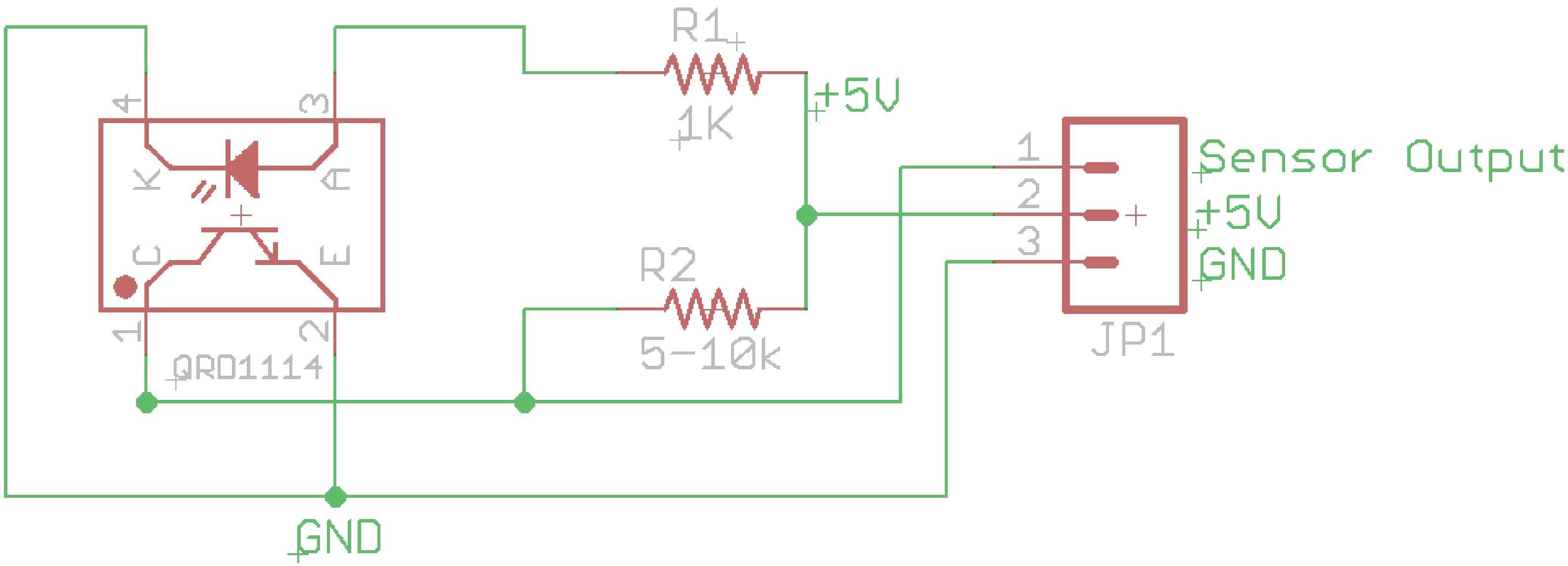
Other Ambient Light Sensors



Cadmium… not RoHS

Reflective Light Sensor

Using the QRD1114



Selecting R2?

- Need to ensure proper signal for your output circuit
- Be very careful of temperature variations (demo on Monday)

Performance Curve

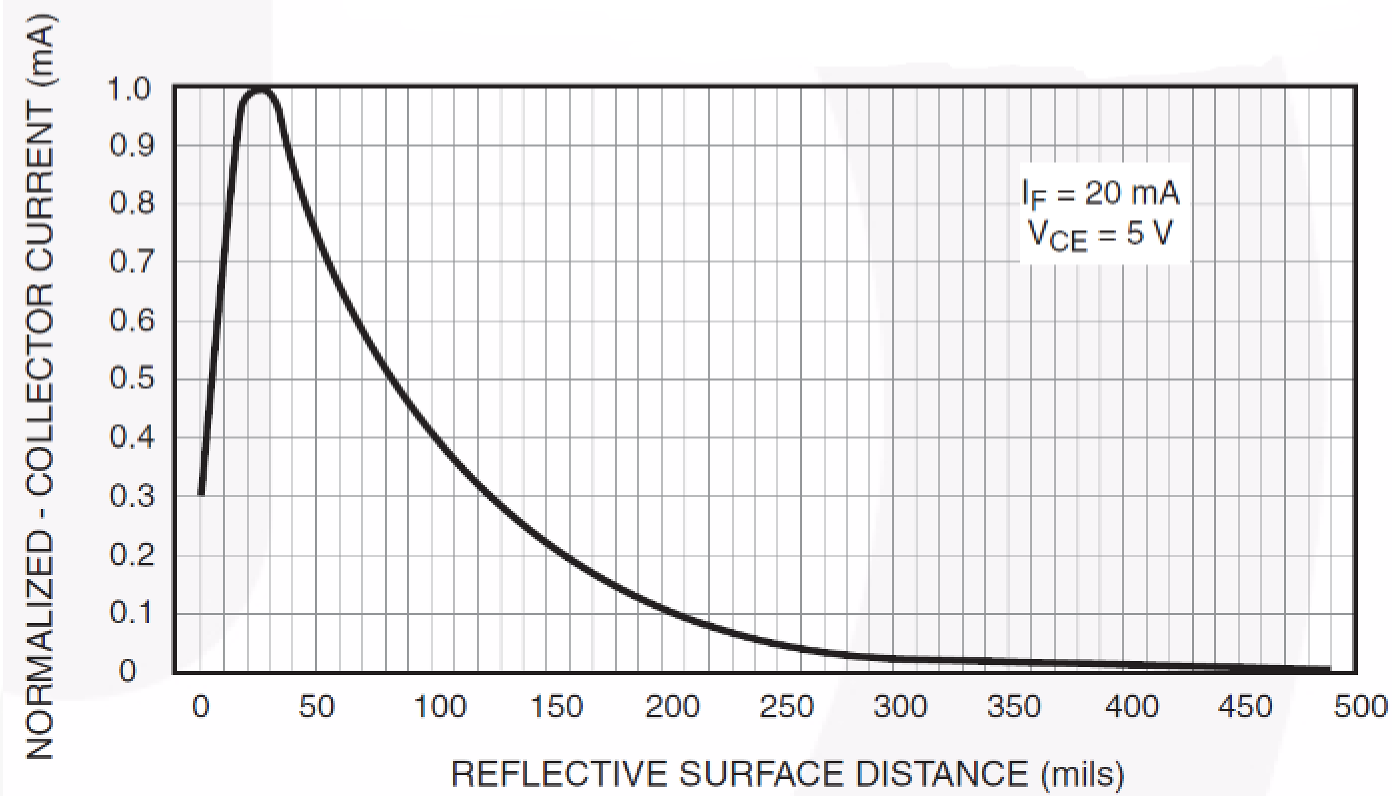
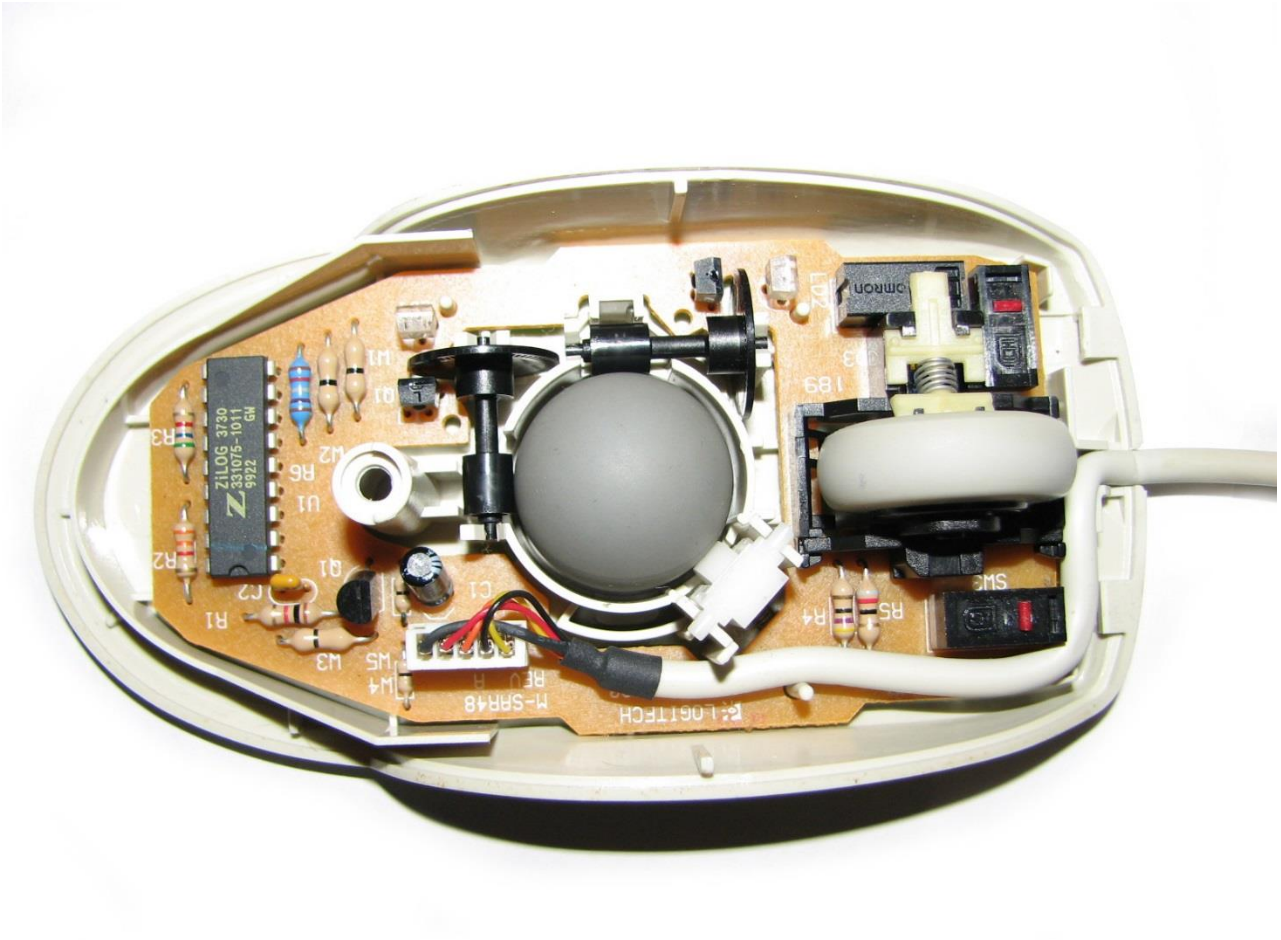


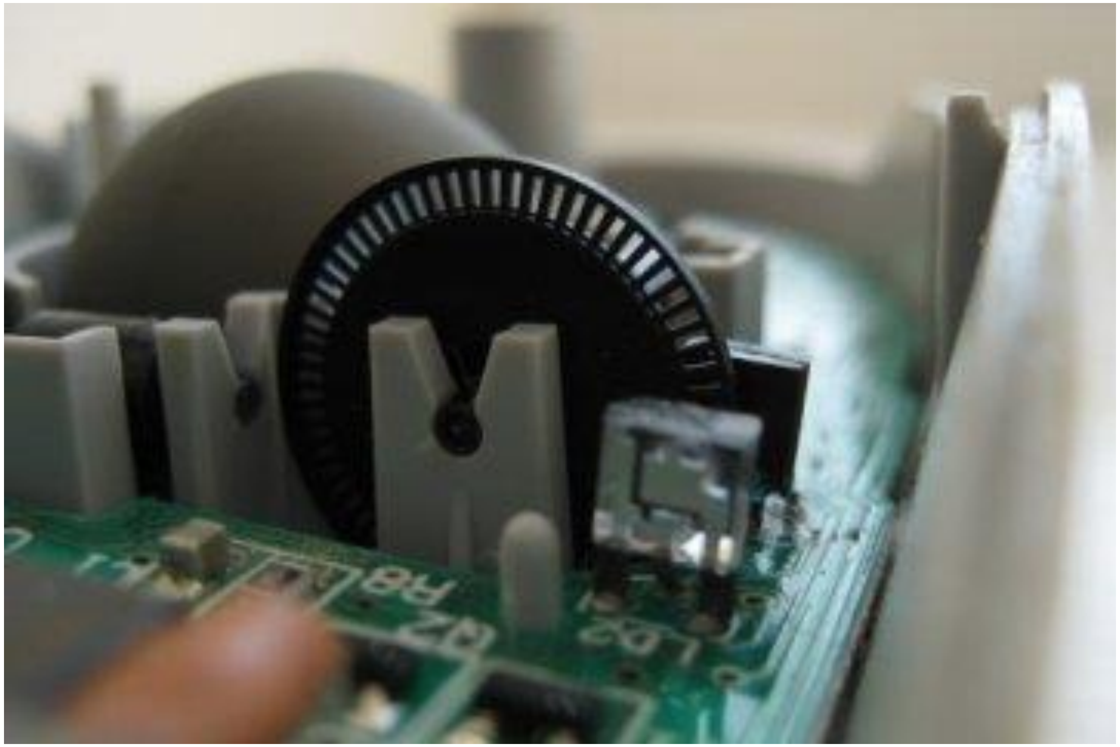
Figure 5. Normalized Collector Current vs. Distance

Reflective Tape…

Wheel Encoders



Wheel Encoders



…on your robot

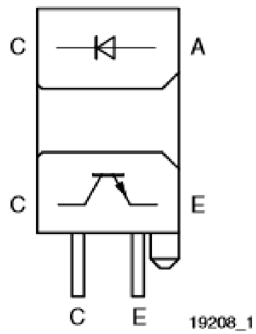
Transmissive Part

TCST5250

Vishay Semiconductors

Transmissive Optical Sensor with Phototransistor Output

21834



19208_1

DESCRIPTION

The TCST5250 is a transmissive sensor that includes an infrared emitter and a phototransistor, located face-to-face on the optical axes in a leaded package which blocks visible light.

FEATURES

- Package type: leaded
- Detector type: phototransistor
- Dimensions (L x W x H in mm): 14.3 x 6 x 9.5
- Gap (in mm): 2.7
- Aperture (in mm): 0.5
- Typical output current under test: I_C = 1.5 mA
- Daylight blocking filter
- Emitter wavelength: 950 nm
- Lead (Pb)-free soldering released
- Compliant to RoHS directive 2002/95/EC and in accordance to WEEE 2002/96/EC

Pb-free

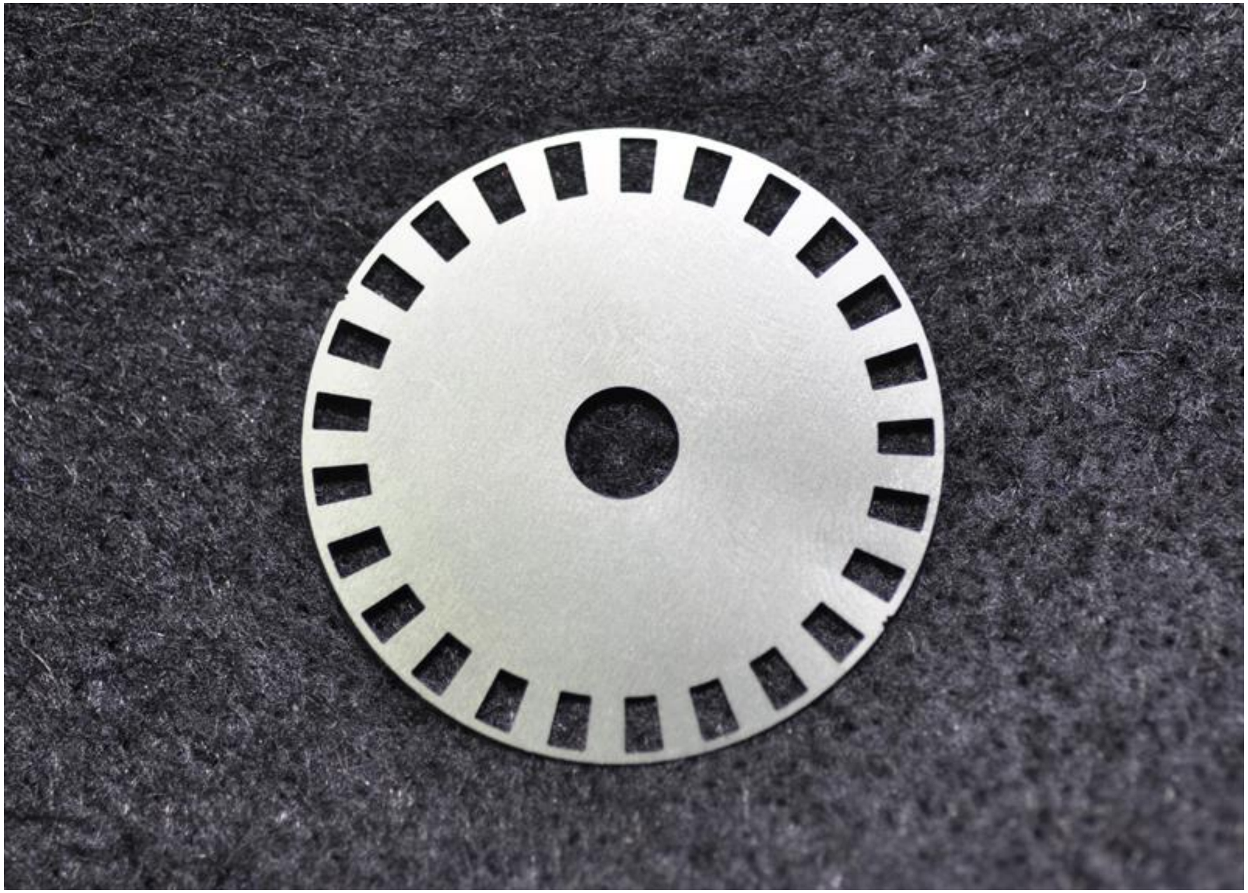
e2

RoHS
COMPLIANT

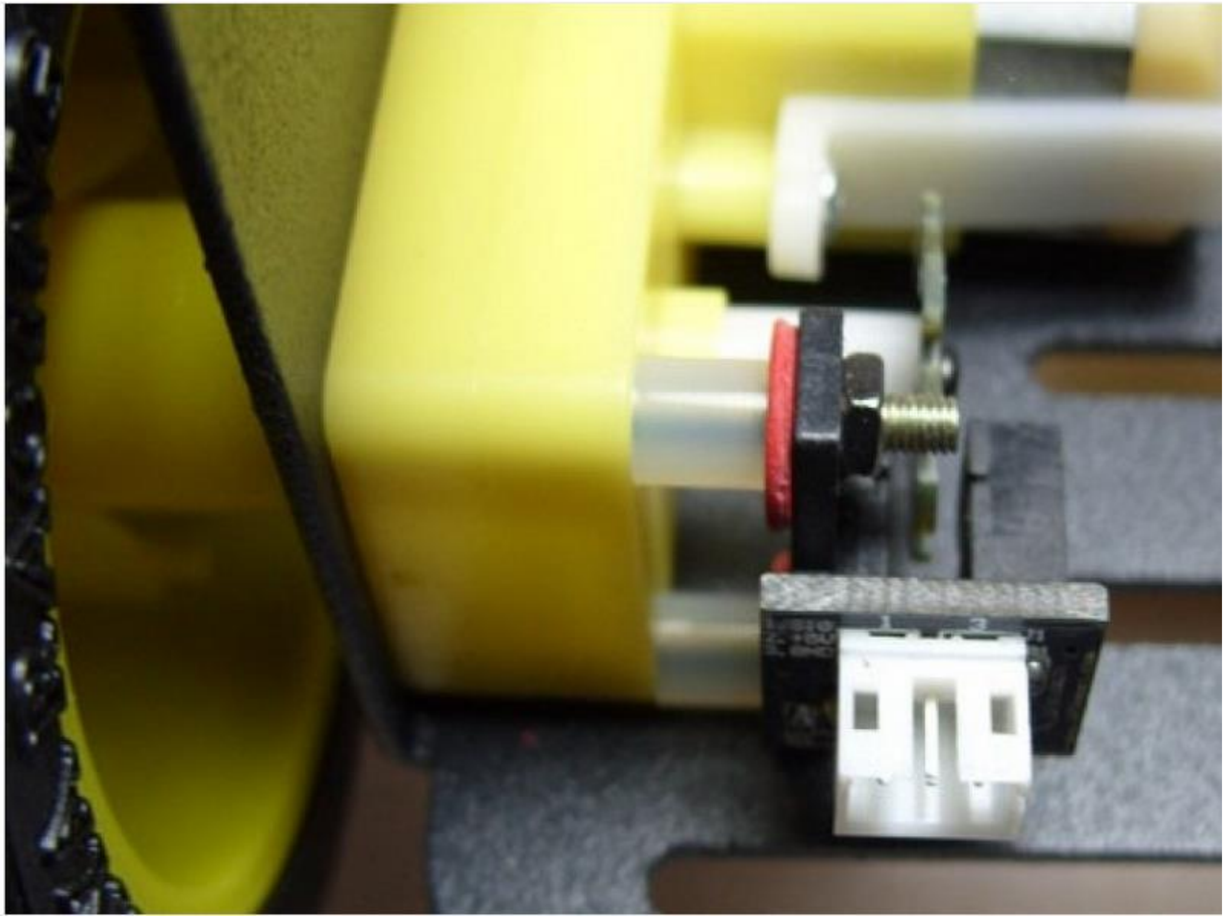
APPLICATIONS

- Optical switch
- Shaft encoder

Encoder Disk Design



Mounting Wheel Encoders



3D Printing Required

Problems with Wheel Encoders

..more details in the navigational lecture, but briefly:

- What happens if wheels slip?
- What happens if object holds robot back?

Metal Detectors

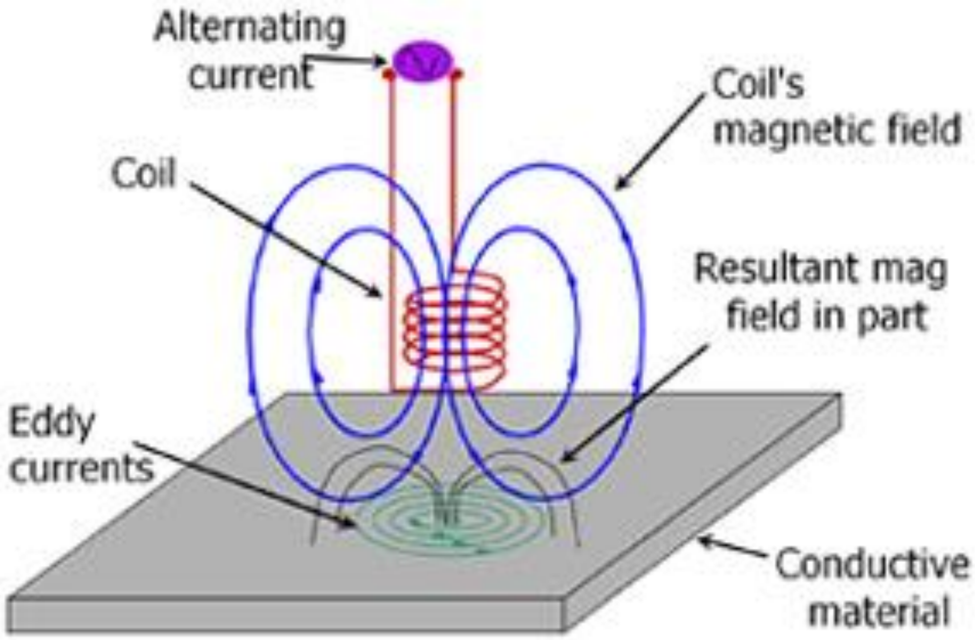
...more than just for vacations.

Basic Principle

Some Possible Methods

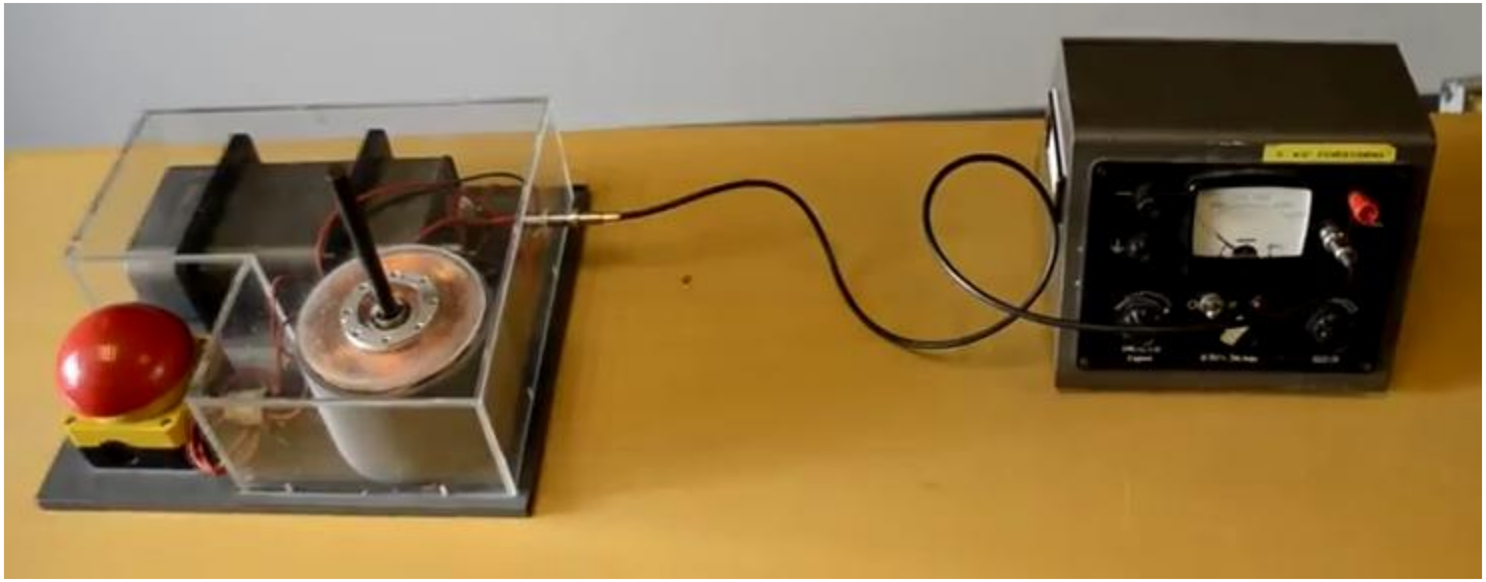
- Detecting de-tuning of tank circuit
- Detecting variations in Earth's magnetic field

Eddy Currents



Fun With Eddy’s

Also – Lenz’s Law tells us induced current will cause magnetic field that opposes inducing field

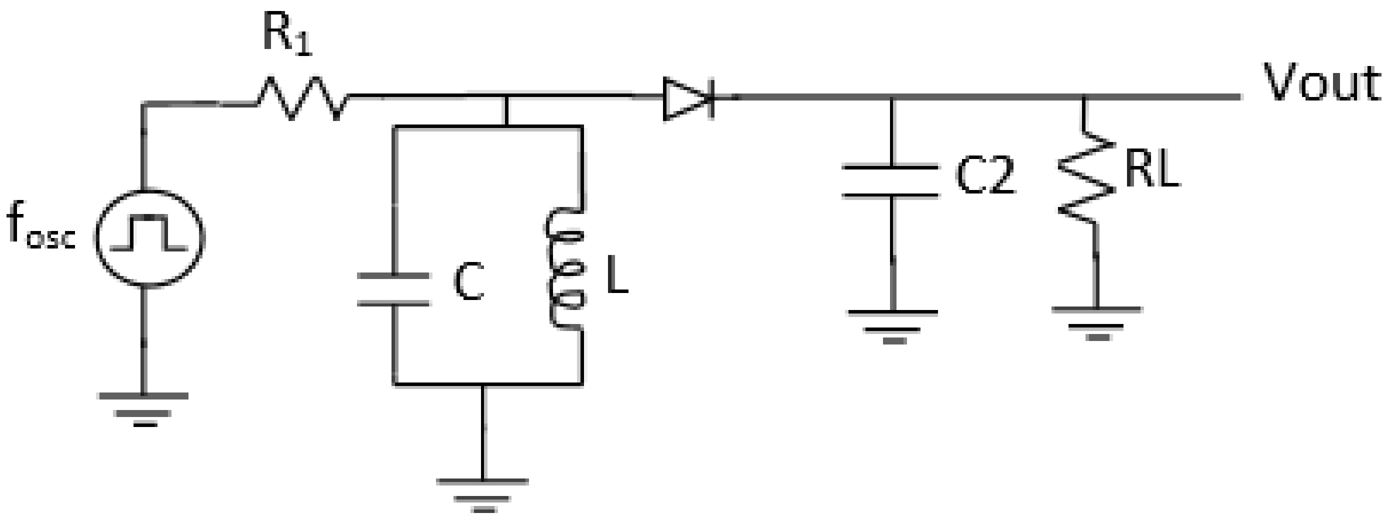


https://www.youtube.com/watch?v=0dU-t9nTbU4

Metal Detector

Colin O'Flynn

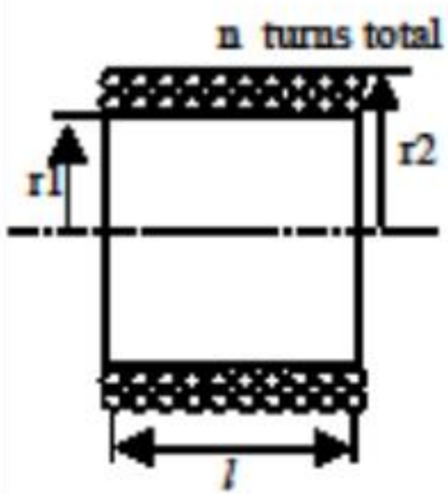
Simple Metal Detector



Items to Select in Design

1. R1
2. C
3. L
4. C2
5. RL

Calculating Inductance



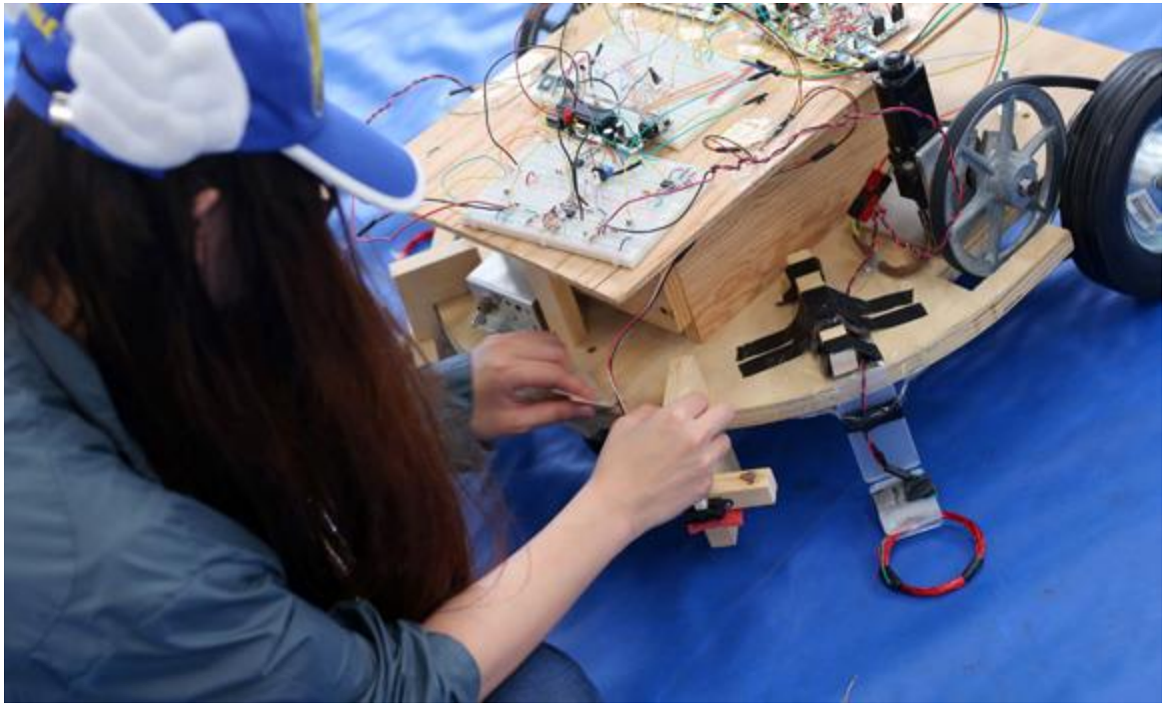
$$L = \frac{31.6 \cdot r1^2 \cdot n^2}{6 \cdot r1 + 9 \cdot l + 10 \cdot (r2 - r1)}$$

L = inductance in microhenries
r1 = inside coil radius in meters
r2 = outside coil radius in meters
l = coil length in meters
n = # of turns total

Multi-layer Air Core Solenoid

Building a Sense Coil

Mounting to Robot



Amplified Detector

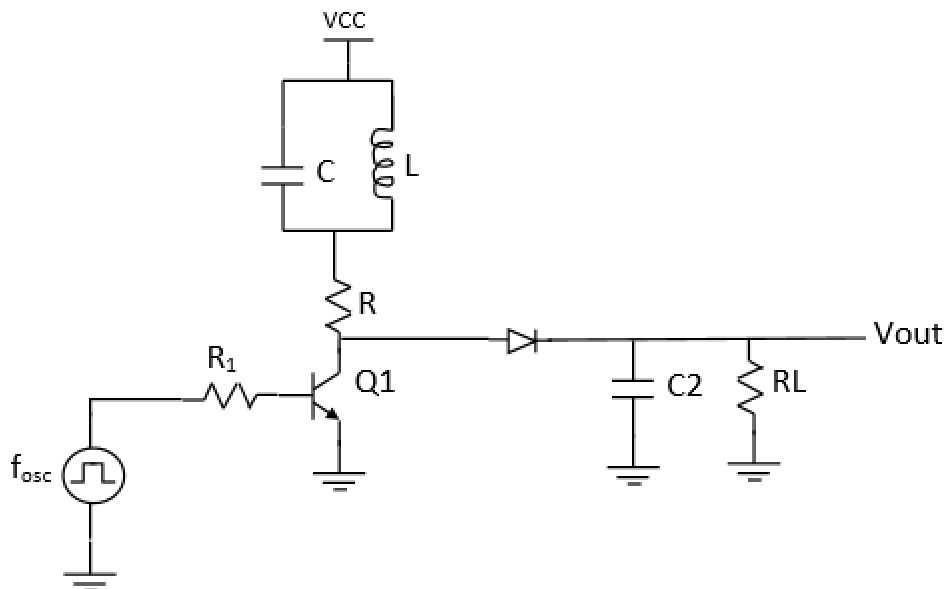


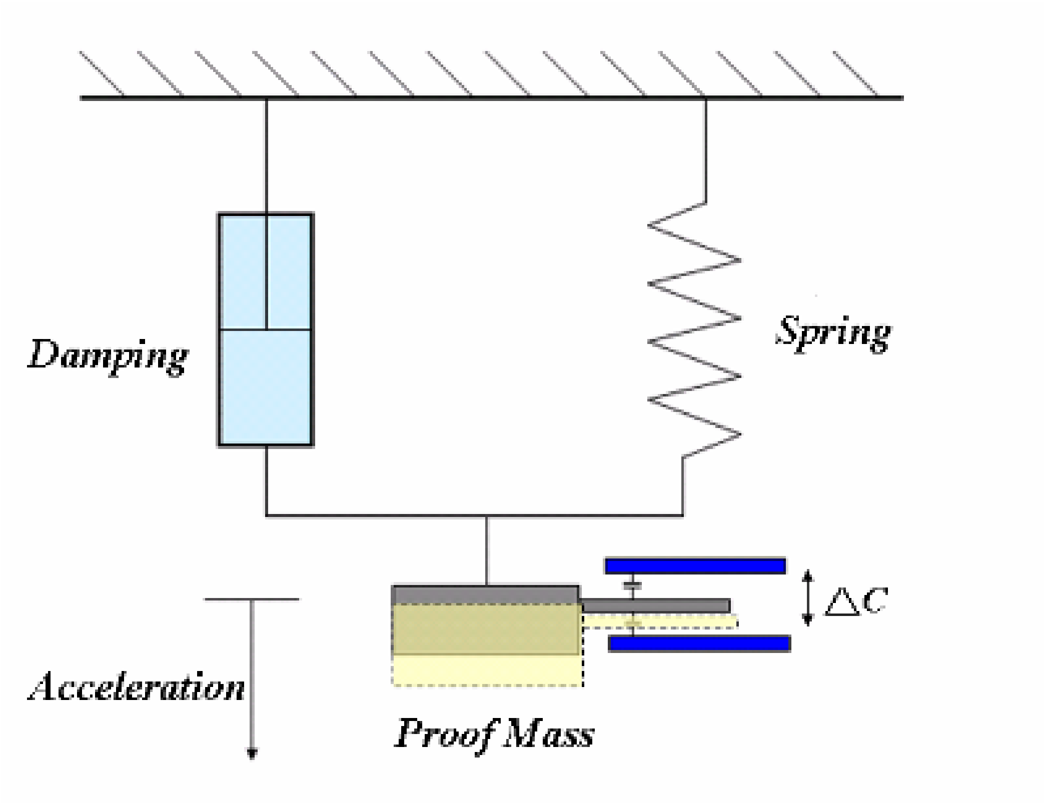
Figure 4

…more details later

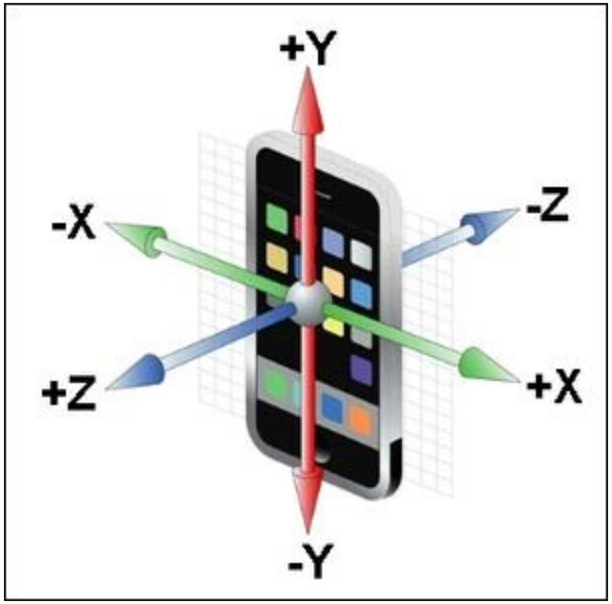
Will build a metal detector in Lab #5 – so more details to follow!

Acceleration

Basic Principle



3-Axis Accelerometers



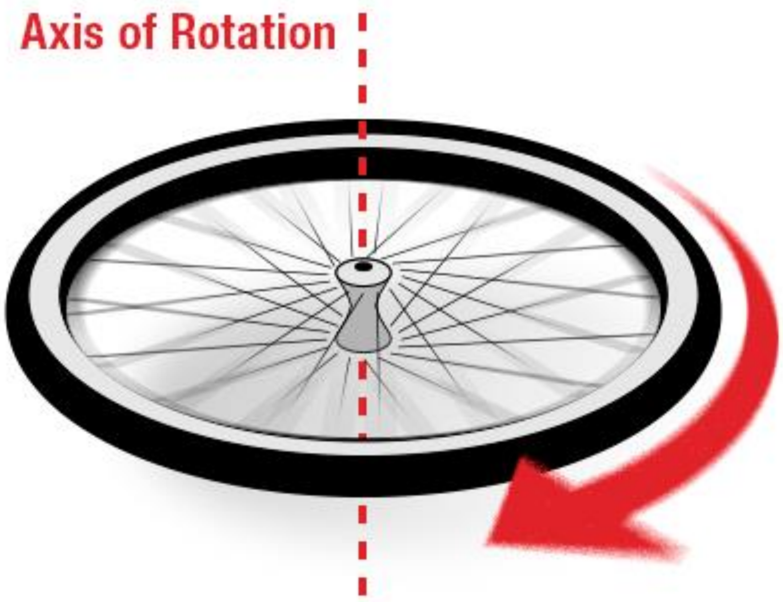
Examples of Uses

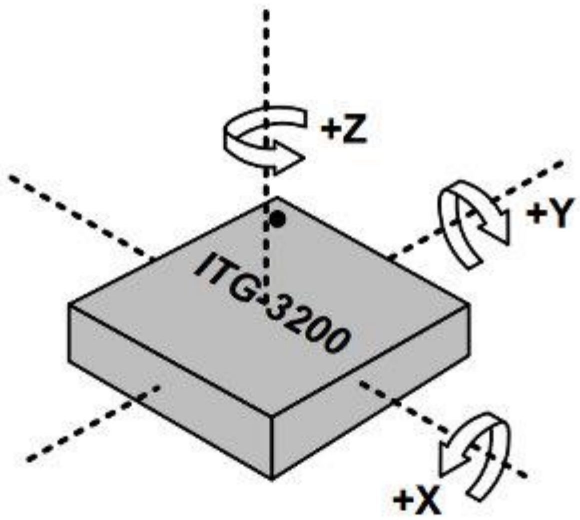
…position??

Gravity Vector

Gyros

Rotational Speed



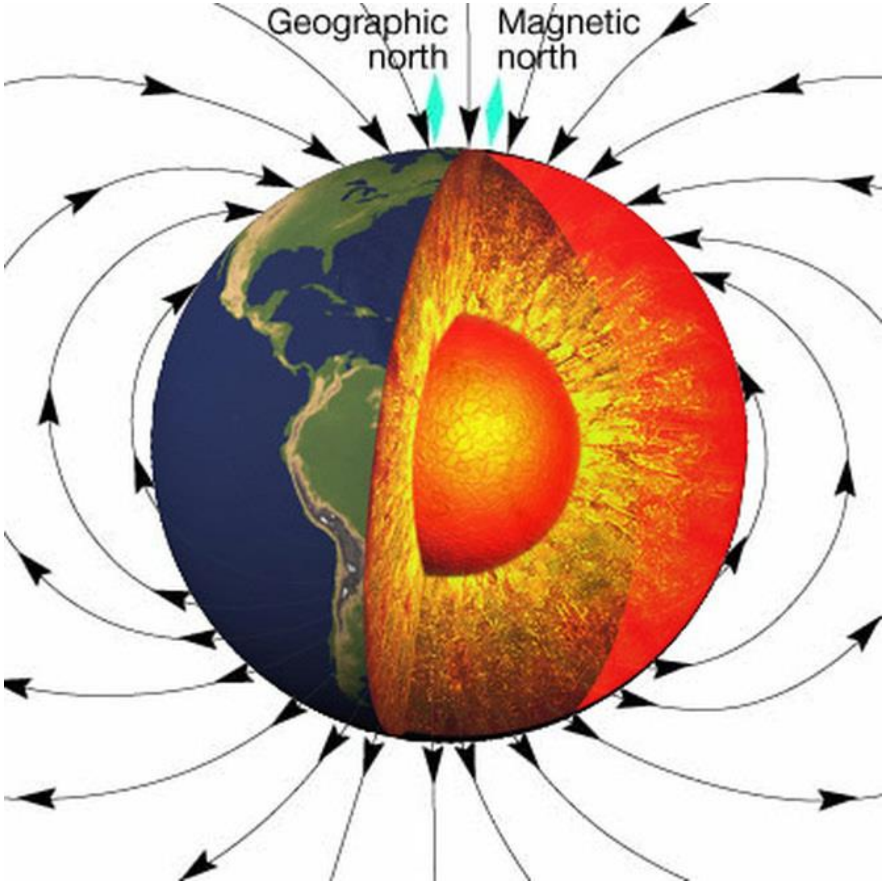

+Z
+Y
+X
ITG-3200

Problems with Drift

…Inertial Measurement Units (IMU)

Magnetometers

Earth’s Magnetic Field



Compass

Single Axis

Fixing it with Three Axis

Summary

- Many sensor options in your robot
- Plan to experiment a little (PoC)
- Covered how Timers & Interrupts work
- Plan on using wheel encoders to sense position
- Metal detectors to sense magnetic strips
- Accelerometers have many uses
- Can combine with gyros, magnetometers to form complete solution (more details in later lectures/labs)