Today
Collision Detection and EEPROM

This Week's Lab
Without the operating system, write a program that does line tracking and collision detection simultaneously.

Collision Detection
There are two easy ways to implement collision detection:

1. **Use infrared light.**
    a. Black absorbs color. White reflects. An infrared LED is turned on. The photons proceed out conically and reflect off the surface toward the component that emitted them.
2. **Use ultrasonic ranger**
    a. Bursts of sound energy higher than what humans can hear (20Hz – 20KHz) are emitted.

The robot has the infrared solution.

Darlington configuration is a special form of a BJT which provides higher current gain and faster switching. It is basically a current controlled switch.

An oscillating NAND gate turns an infrared led on and off at a rate of 38KHz. Then, the infrared demodulator will see that if you are close enough to the wall.

The demodulator gets a better response with bursts of input as opposed to a continuous signal.

When pulsing it, you get a heightened range. Since the only energy out there is the energy from your pulse,

Hold the robot up to a white surface, connected to the scope. Turn the software on, move it in toward the white surface, and tune the potentiometer on the bottom of the robot until you start to see the waveform come back on the scope channel.

Recommended Solution
Set up TOC2 to time = 0.
Set up TOCTL to toggle the TOC2 pin.

The infrared led should be in the front of the robot.

The collision detection should override the line tracking and at least stop.

Putting the Program on the EEPROM
There is no direct drop-in replacement for our memory chip.

Using an EEPROM means we cannot do STAA from $8000 and up. Therefore, any variable storage must be done on the internal RAM (0x0000 to 0x01FF). Don't write to a variable unless you have to.