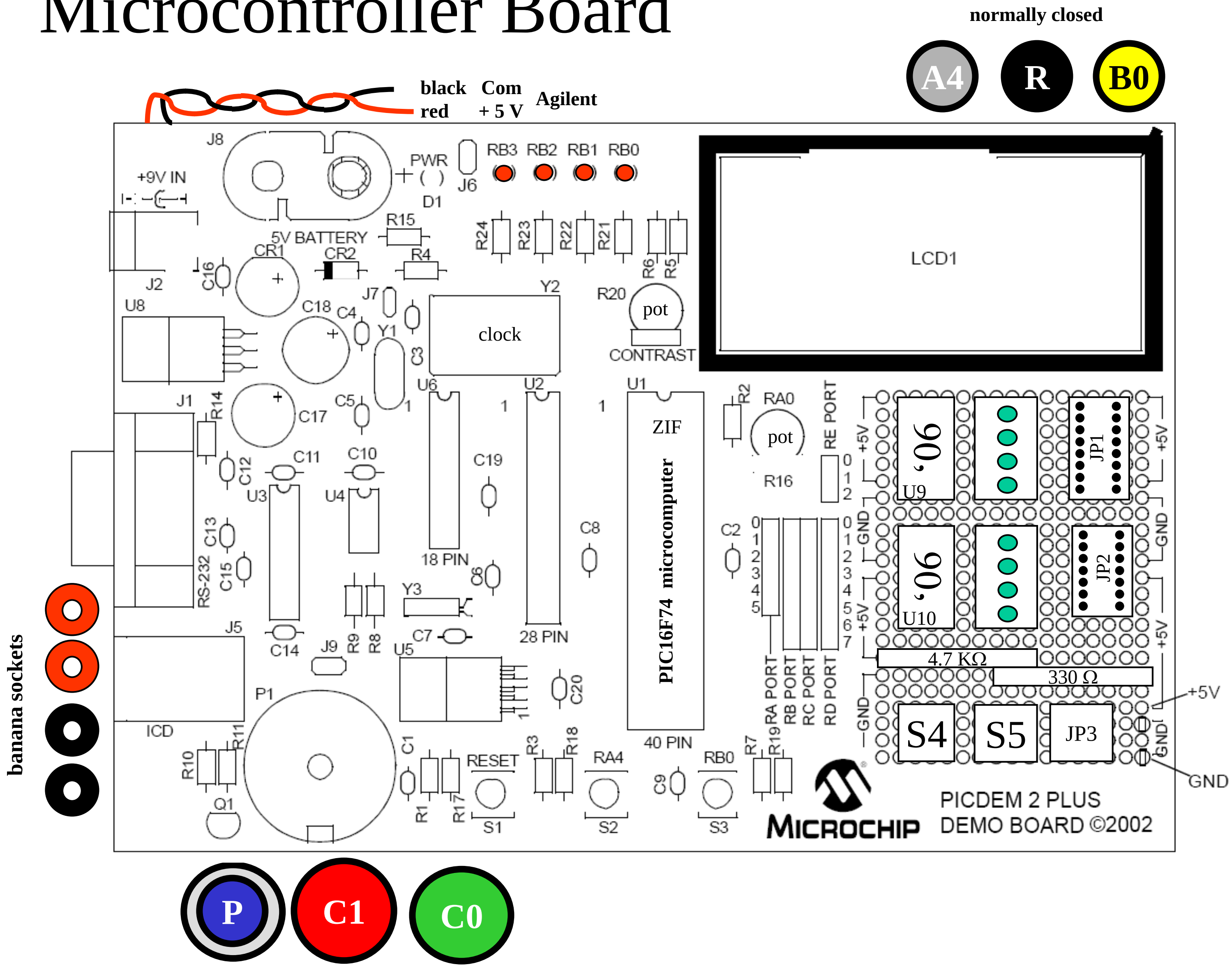
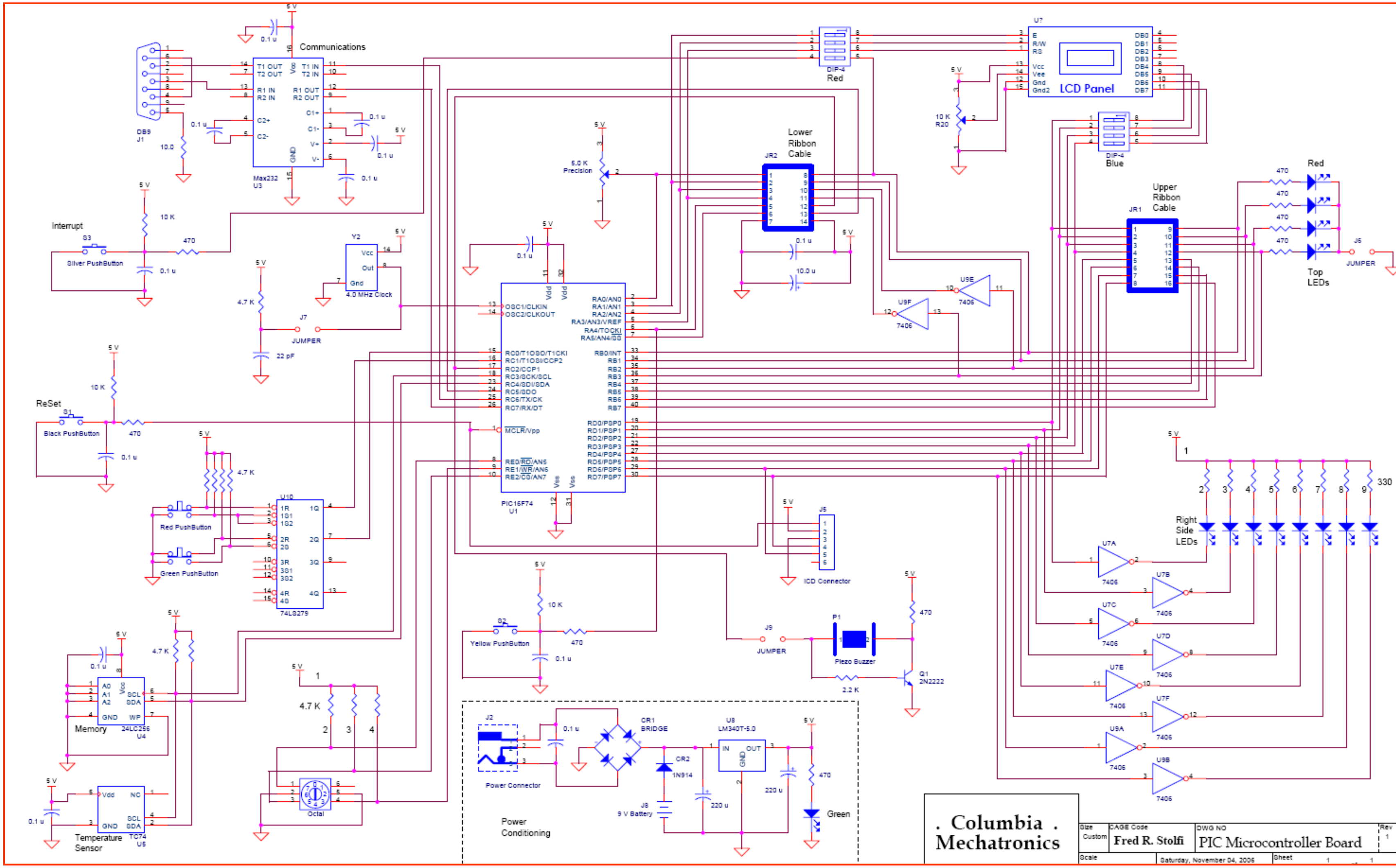
Introduction to On / Off Control

Microcontroller Board



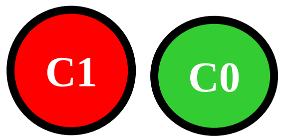
Microcontroller Board Schematic



Interfaces

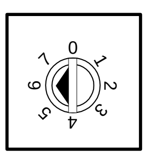
User Interface

Input



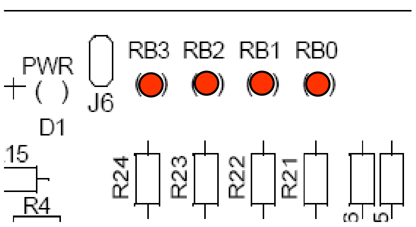
pushbutton switches

analog knob (potentiometer)



selector (octal switch)

Output



Idiot lights
(Port B LEDs)

Found on Microcontroller Board.

System Interface

driver for solenoid
(transistor)

circuit for optical sensor
(comparator)
(1 bit A/D converter)

To be fabricated on Proto Board.

Indicating an Error

Four LEDs on the microcomputer board

Bottom 4 bits of Port B

RB 3 RB 2 RB 1 RB 0

Indicates an error

on $\Rightarrow$ error

off $\Rightarrow$ OK

Indicates the mode in binary

0 through 7

For example

1 0 1 0 $\Rightarrow$ error in mode 2

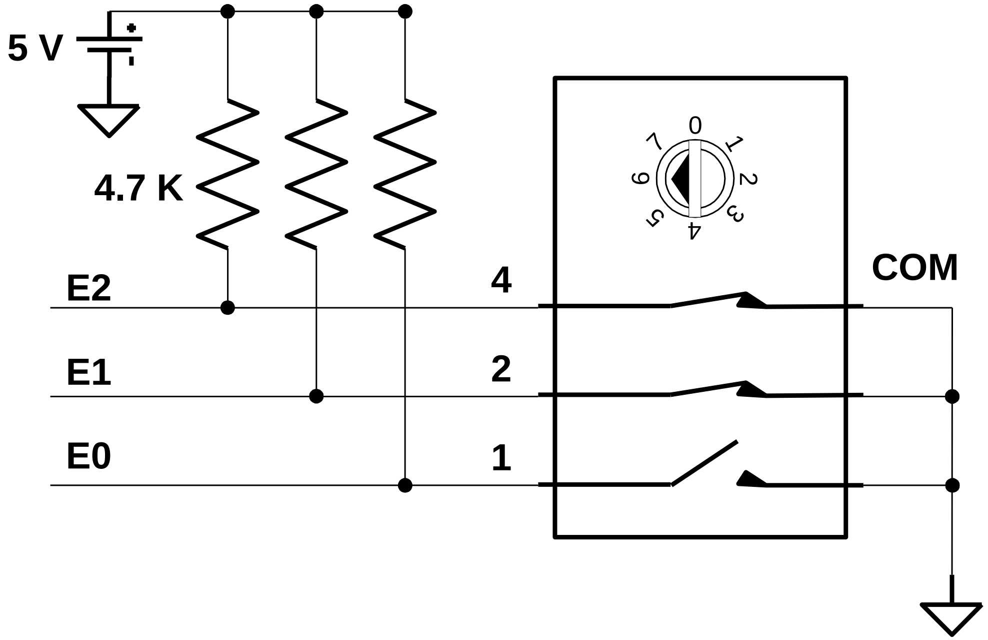
Assembly Include File

```
;==========================================================================
;
;       Register Definitions
;
;==========================================================================

W                            EQU   H'0000'
F                            EQU   H'0001'

;----- Register Files -----------------------------------------------------

;-----Bank0------------------
INDF                         EQU   H'0000'
TMR0                         EQU   H'0001'
PCL                          EQU   H'0002'
STATUS                       EQU   H'0003'
FSR                          EQU   H'0004'
PORTA                        EQU   H'0005'
PORTB                        EQU   H'0006'
PORTC                        EQU   H'0007'
PORTD                        EQU   H'0008'
PORTE                        EQU   H'0009'
PCLATH                       EQU   H'000A'
INTCON                       EQU   H'000B'
PIR1                         EQU   H'000C'
PIR2                         EQU   H'000D'
TMR1                         EQU   H'000E'
TMR1L                        EQU   H'000E'
TMR1H                        EQU   H'000F'
T1CON                        EQU   H'0010'
TMR2                         EQU   H'0011'
T2CON                        EQU   H'0012'
SSPBUF                       EQU   H'0013'
SSPCON                       EQU   H'0014'
CCPR1                        EQU   H'0015'
CCPR1L                       EQU   H'0015'
CCPR1H                       EQU   H'0016'
CCP1CON                      EQU   H'0017'
RCSTA                        EQU   H'0018'
TXREG                        EQU   H'0019'
RCREG                        EQU   H'001A'
CCPR2                        EQU   H'001B'
CCPR2L                       EQU   H'001B'
CCPR2H                       EQU   H'001C'
```

```
;----- STATUS Bits --------------------------------------------------------
C                            EQU   H'0000'
DC                           EQU   H'0001'
Z                            EQU   H'0002'
NOT_PD                       EQU   H'0003'
NOT_TO                       EQU   H'0004'
IRP                          EQU   H'0007'

RP0                          EQU   H'0005'
RP1                          EQU   H'0006'

;----- PORTA Bits --------------------------------------------------------
RA0                          EQU   H'0000'
RA1                          EQU   H'0001'
RA2                          EQU   H'0002'
RA3                          EQU   H'0003'
RA4                          EQU   H'0004'
RA5                          EQU   H'0005'
RA6                          EQU   H'0006'
RA7                          EQU   H'0007'

;----- PORTB Bits --------------------------------------------------------
RB0                          EQU   H'0000'
RB1                          EQU   H'0001'
RB2                          EQU   H'0002'
RB3                          EQU   H'0003'
RB4                          EQU   H'0004'
RB5                          EQU   H'0005'
RB6                          EQU   H'0006'
RB7                          EQU   H'0007'

;----- PORTC Bits --------------------------------------------------------
RC0                          EQU   H'0000'
RC1                          EQU   H'0001'
RC2                          EQU   H'0002'
RC3                          EQU   H'0003'
RC4                          EQU   H'0004'
RC5                          EQU   H'0005'
RC6                          EQU   H'0006'
RC7                          EQU   H'0007'
```

```
;----- ADCON0 Bits --------------------------------------------------------
ADON                         EQU   H'0000'
CHS3                         EQU   H'0001'
GO_NOT_DONE                  EQU   H'0002'

GO                           EQU   H'0002'
CHS0                         EQU   H'0003'
CHS1                         EQU   H'0004'
CHS2                         EQU   H'0005'
ADCS0                        EQU   H'0006'
ADCS1                        EQU   H'0007'

NOT_DONE                     EQU   H'0002'

GO_DONE                      EQU   H'0002'

;----- ADCON1 Bits --------------------------------------------------------
ADCS2                        EQU   H'0006'
ADFM                         EQU   H'0007'

PCFG0                        EQU   H'0000'
PCFG1                        EQU   H'0001'
PCFG2                        EQU   H'0002'
PCFG3                        EQU   H'0003'
VCFG0                        EQU   H'0004'
VCFG1                        EQU   H'0005'
```

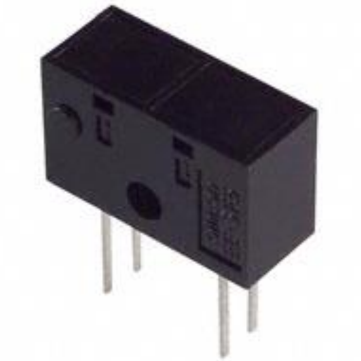
Octal Switch

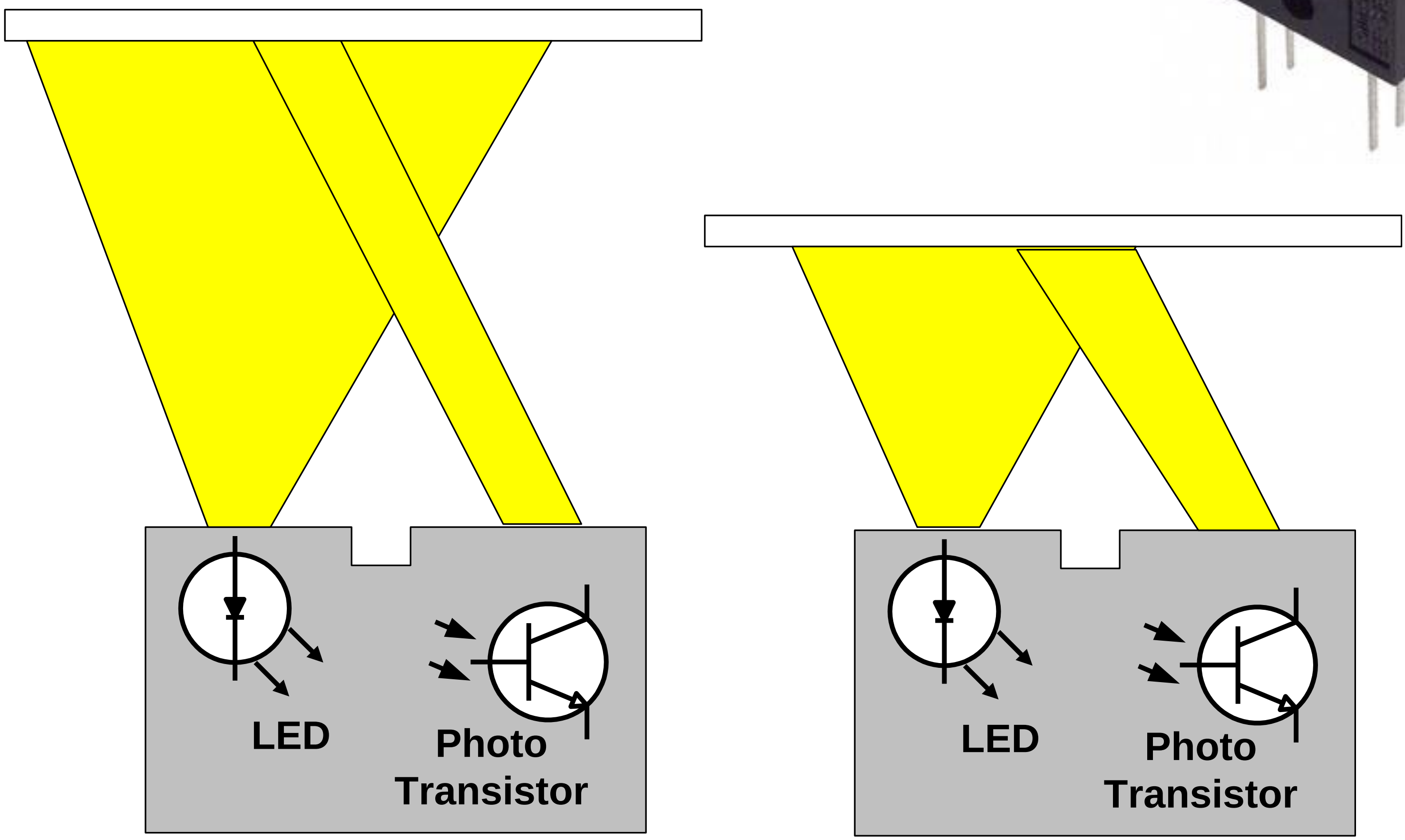


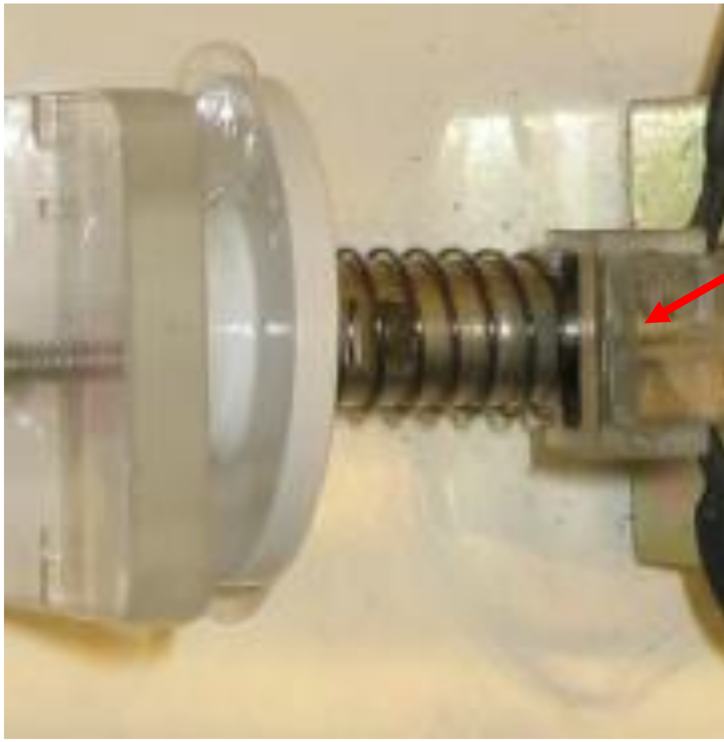
“ 4 “ switch (2^2)
“ 2 “ switch (2^1)
“ 1 “ switch (2^0)

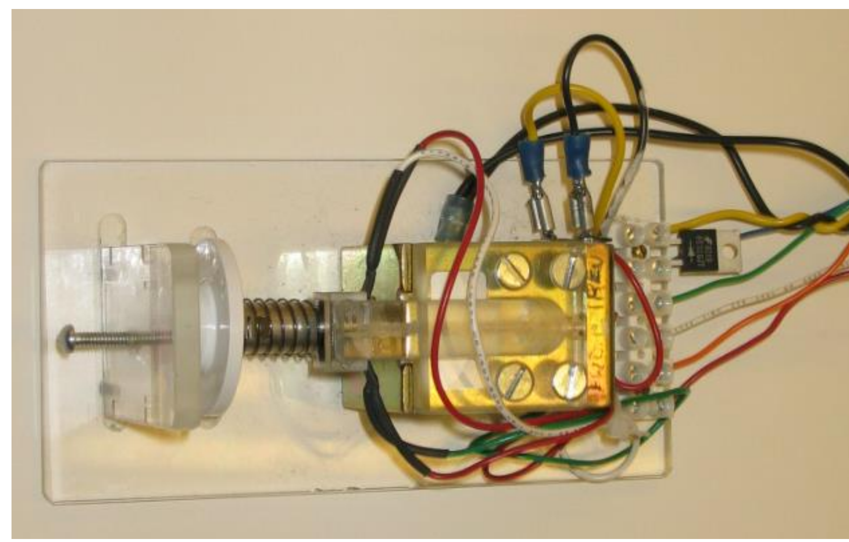
Used to select modes.

Optical Sensor

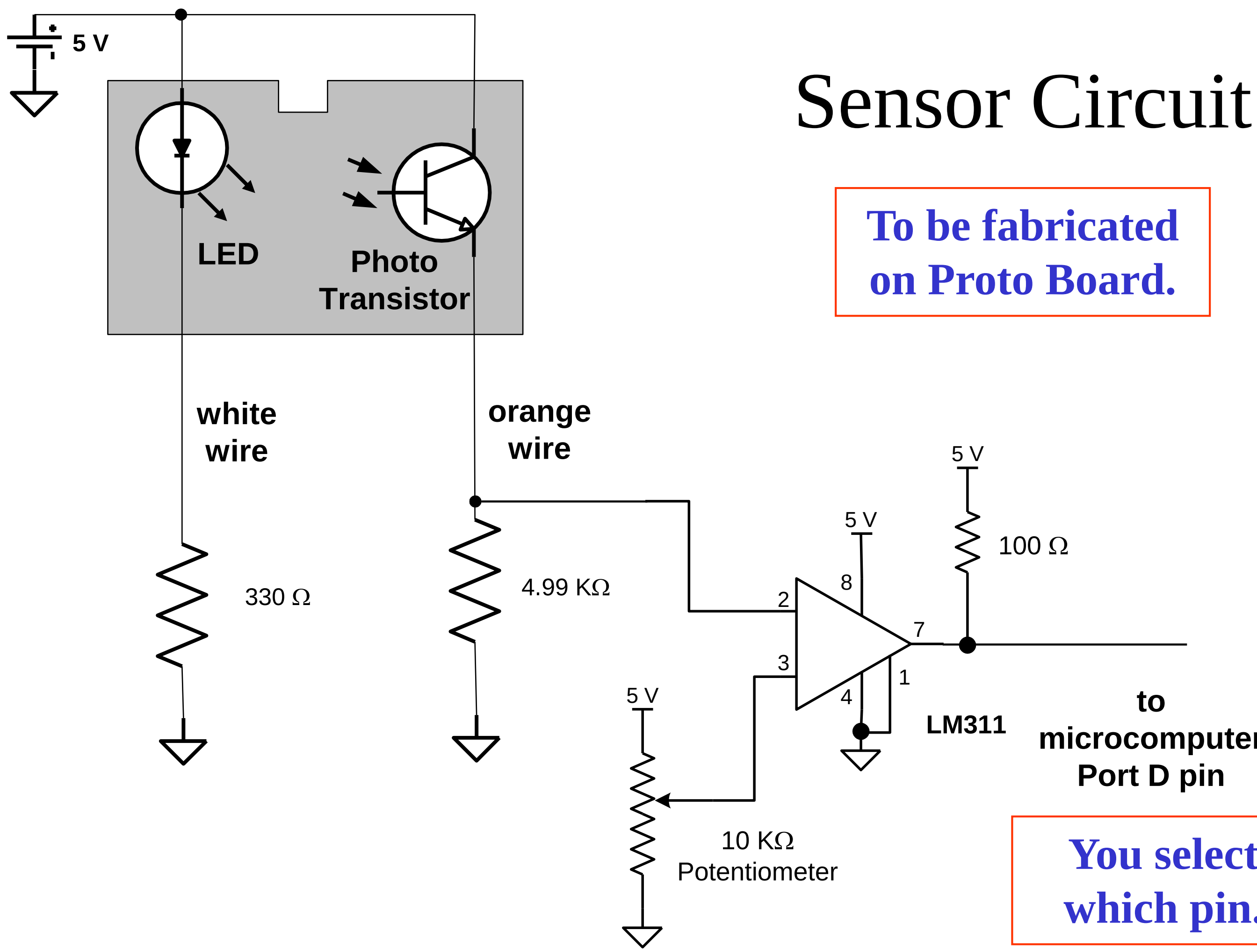





Optical Sensor



Sensor Circuit



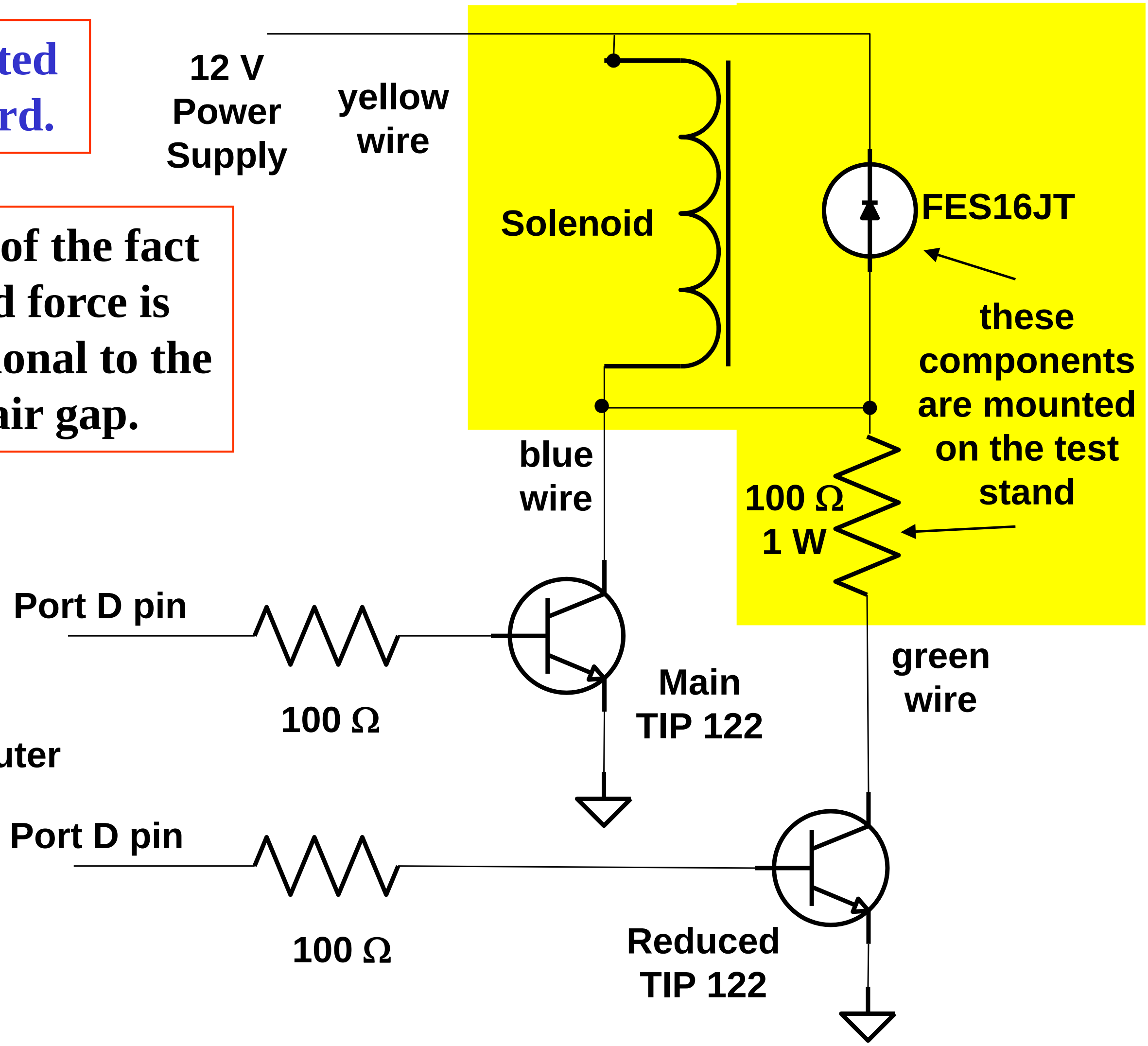
To be fabricated on Proto Board.

You select which pin.

Driver Circuit

To be fabricated on Proto Board.

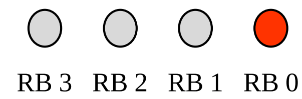
Takes advantage of the fact that the solenoid force is inversely proportional to the square of the air gap.



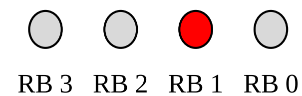
You select which pin.

Case Study Requirements

- Programming Different Modes of Operation
  - Mode 1 (indicator LEDs 0001)

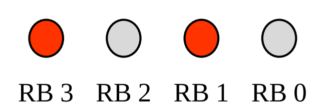


    - Press the red button, the solenoid engages.
    - Press the red button again, the solenoid disengages.
    - Repeats on and off with the red button.
    - Press the green button and a new mode is entered.
  - Mode 2 (indicator LEDs 0010)

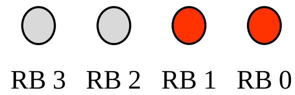


    - Read the value on the control pot
    - Press the red button, the solenoid engages for ¼ the value of the control pot in seconds.
    - Press the red button again before the timing finishes, the timing sequence restarts.
    - After finishing, press the red button again to repeat the process.
    - After finishing, press the green button to switch to a new mode.
    - If the reading of the A/D converter is 0, a fault is indicated.
      (indicator LEDs 1010)

Zero Fault



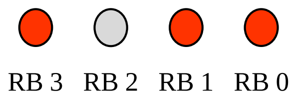
Case Study Requirements (cont.)



– Mode 3 (indicator LEDs 0011)

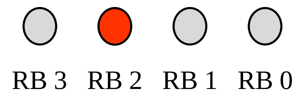
- Press the red button. This should activate the control. A Port D LED (not used for anything else should turn on (indicating that the control is active).
- Read the control pot.
- If the value on the pot read by the A/D converter is greater than 70h (about 4.4 on the potentiometer dial), the solenoid engages.
- If the value on the pot is below 70h, the solenoid should retract.
- Press the red button again. Control should become inactive. The LED indicating that control was active should turn off.
- The solenoid should extend.
- After finishing with control inactive, press the green button to switch to a new mode.
- If the reading of the A/D converter is 0, a fault is indicated. (indicator LEDs 1011)

Zero Fault

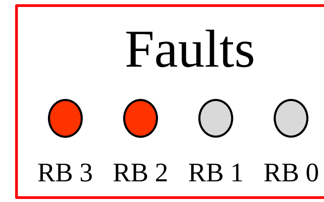


Case Study Requirements (cont.)

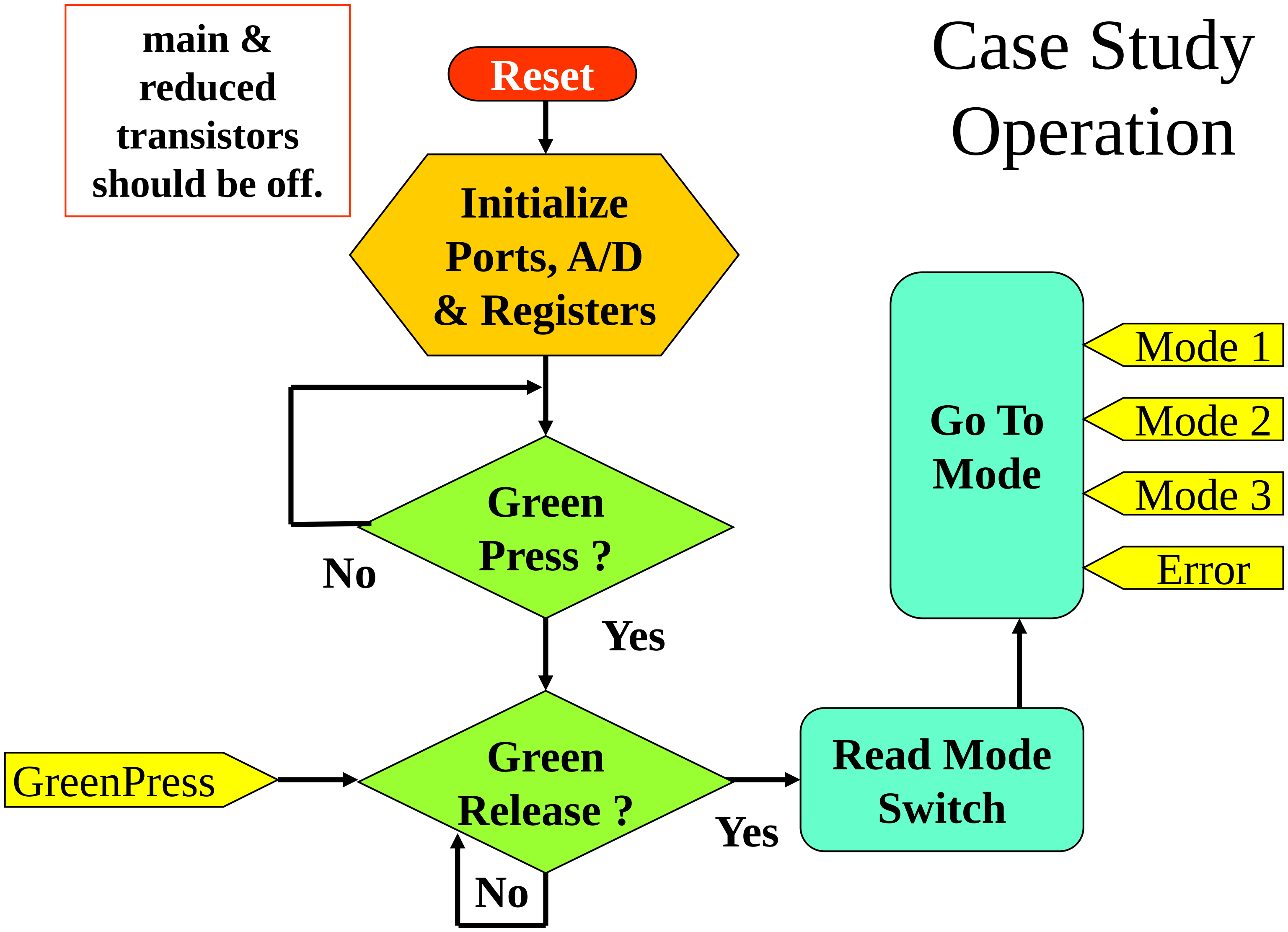
– Mode 4 (indicator LEDs 0100)



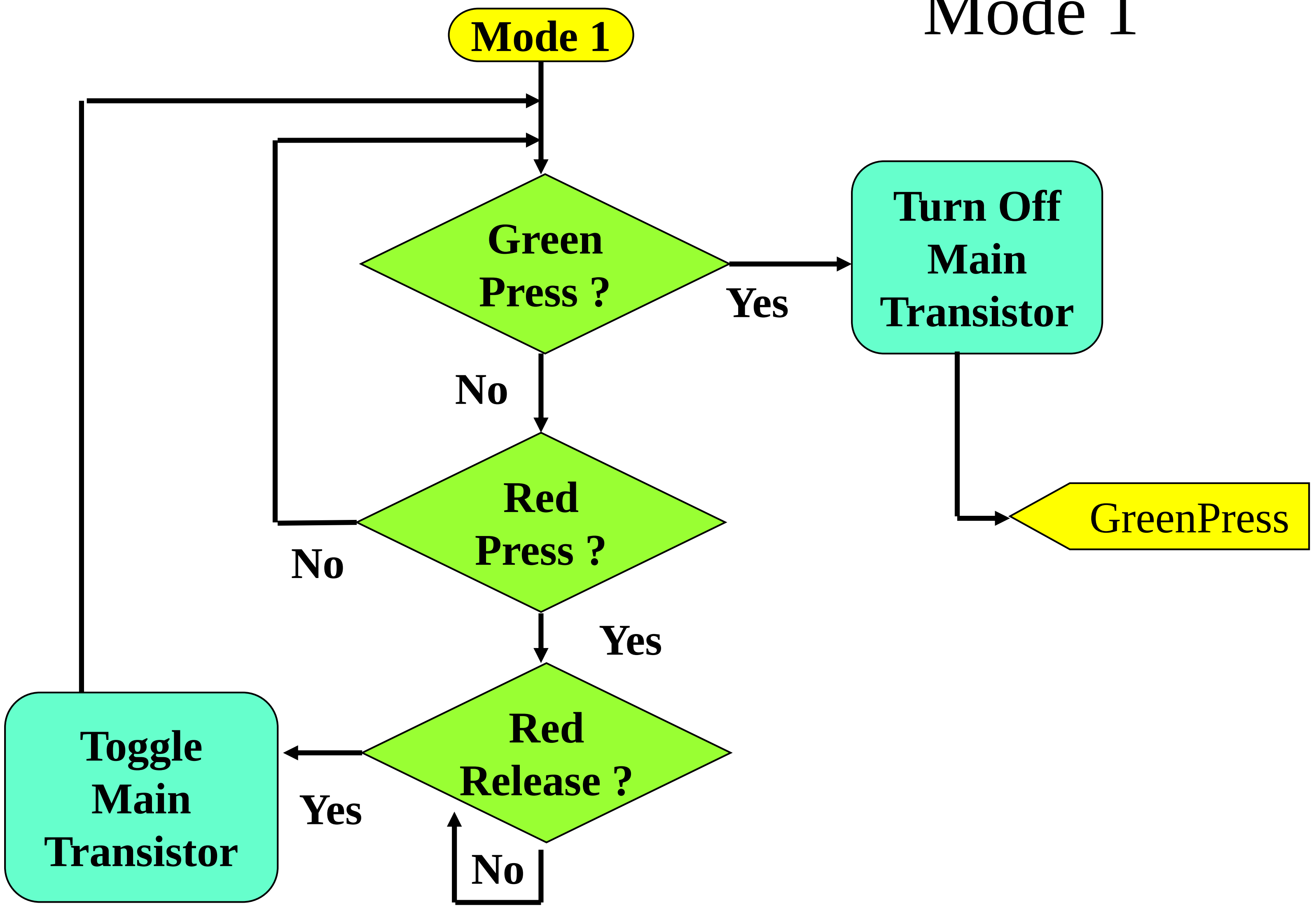
- Read the value on the control pot.
- Press the red button, the solenoid engages with the main transistor.
- As soon as the optical sensor indicates that the solenoid has retracted, turn on the reduced transistor and turn off the main transistor.
- The reduced transistor stays on for ¼ the value of the control pot in seconds.
- Pressing the red button again before the timing finishes DOES NOT restart the timing sequence.
- If the reading of the A/D converter is 0, a fault is indicated. (indicator LEDs 1100)
- If the optical sensor does not indicate that the solenoid has retracted in 10 seconds, turn off the main transistor and indicate a fault (indicator LEDs 1100)
- If the optical sensor indicates that the solenoid has disengaged when the reduced transistor in on, restart the whole sequence again (one time). If the optical sensor indicates that the solenoid has disengaged a second time when the reduced transistor in on, indicate a fault. (indicator LEDs 1100)
- If the solenoid is turned off and the optical sensor indicates that the solenoid is still retracted in 10 seconds, also indicate a fault. (indicator LEDs 1100)
- After finishing successfully, press the red button again to repeat the process.
- After finishing successfully, press the green button to switch to a new mode.
- If a fault, the microcomputer has to be reset with the black reset switch (green and red buttons are ignored).



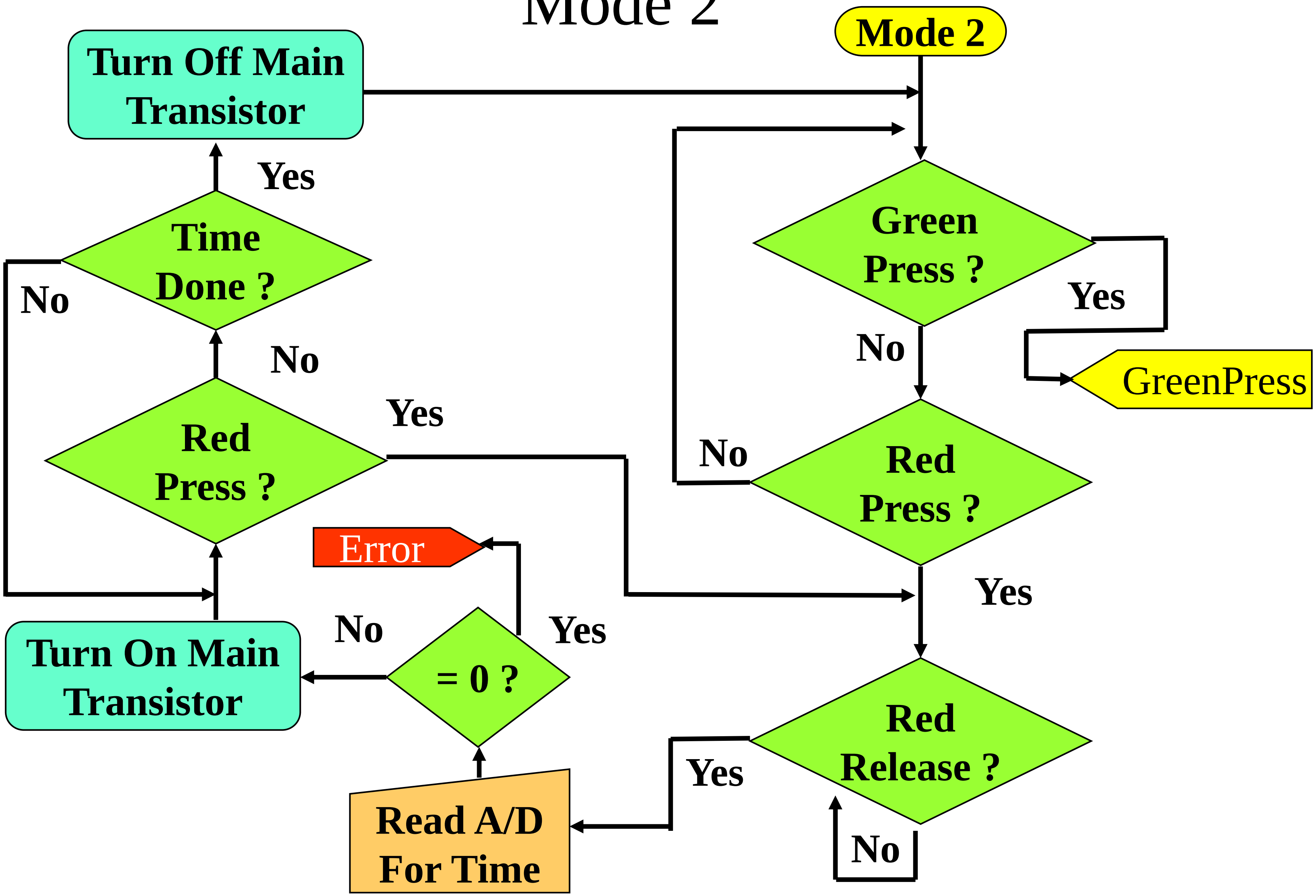
Case Study Operation



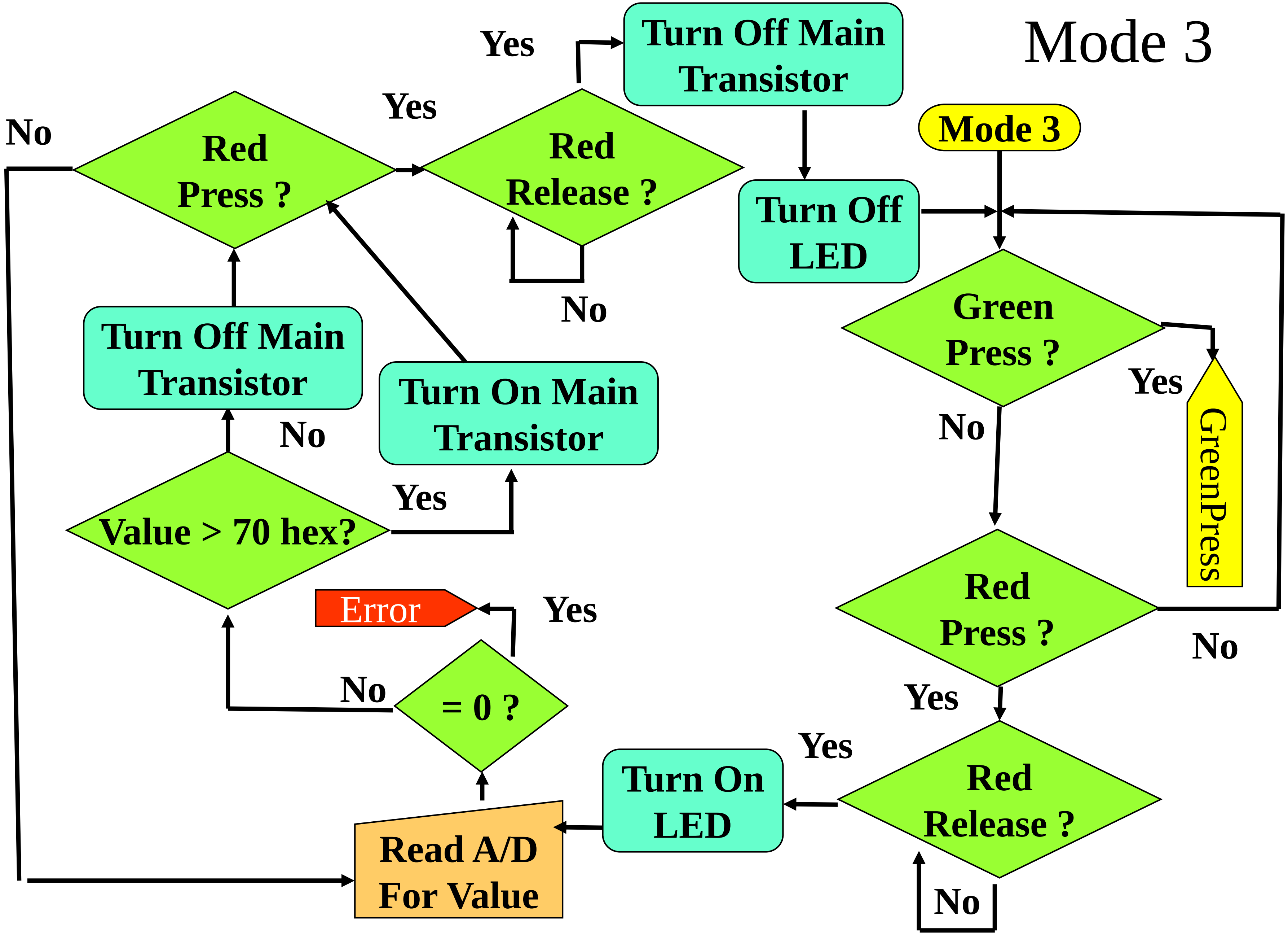
Mode 1


Mode 1
Green Press ?
Yes
Turn Off Main Transistor
No
GreenPress
Red Press ?
No
Yes
Red Release ?
Yes
Toggle Main Transistor
No

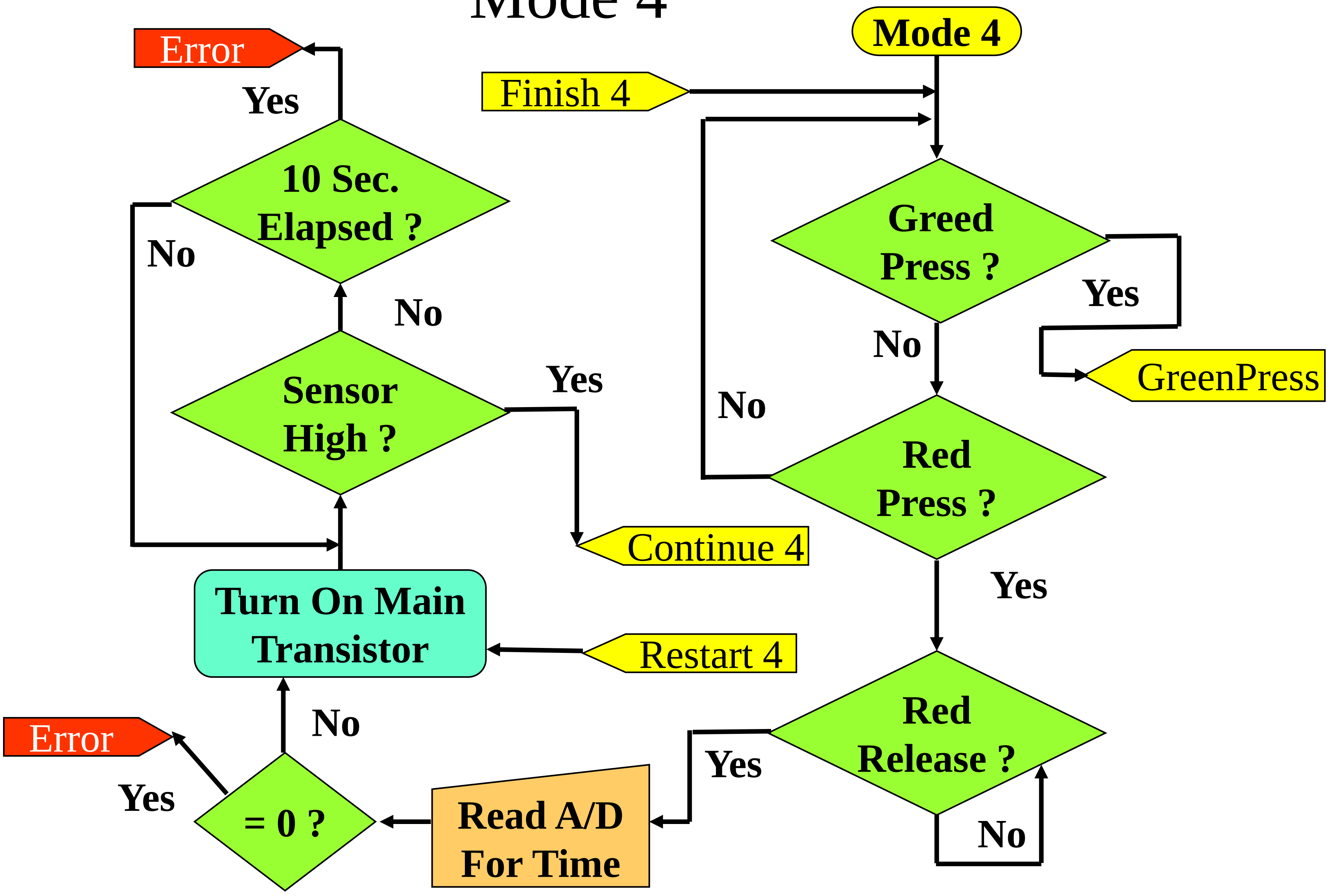
Mode 2


Mode 2
Turn Off Main Transistor
Green Press ?
Yes
GreenPress
No
Time Done ?
Yes
No
Red Press ?
No
Yes
Red Press ?
No
Yes
Error
= 0 ?
Yes
No
Turn On Main Transistor
Red Release ?
Yes
No
Read A/D For Time

Mode 3

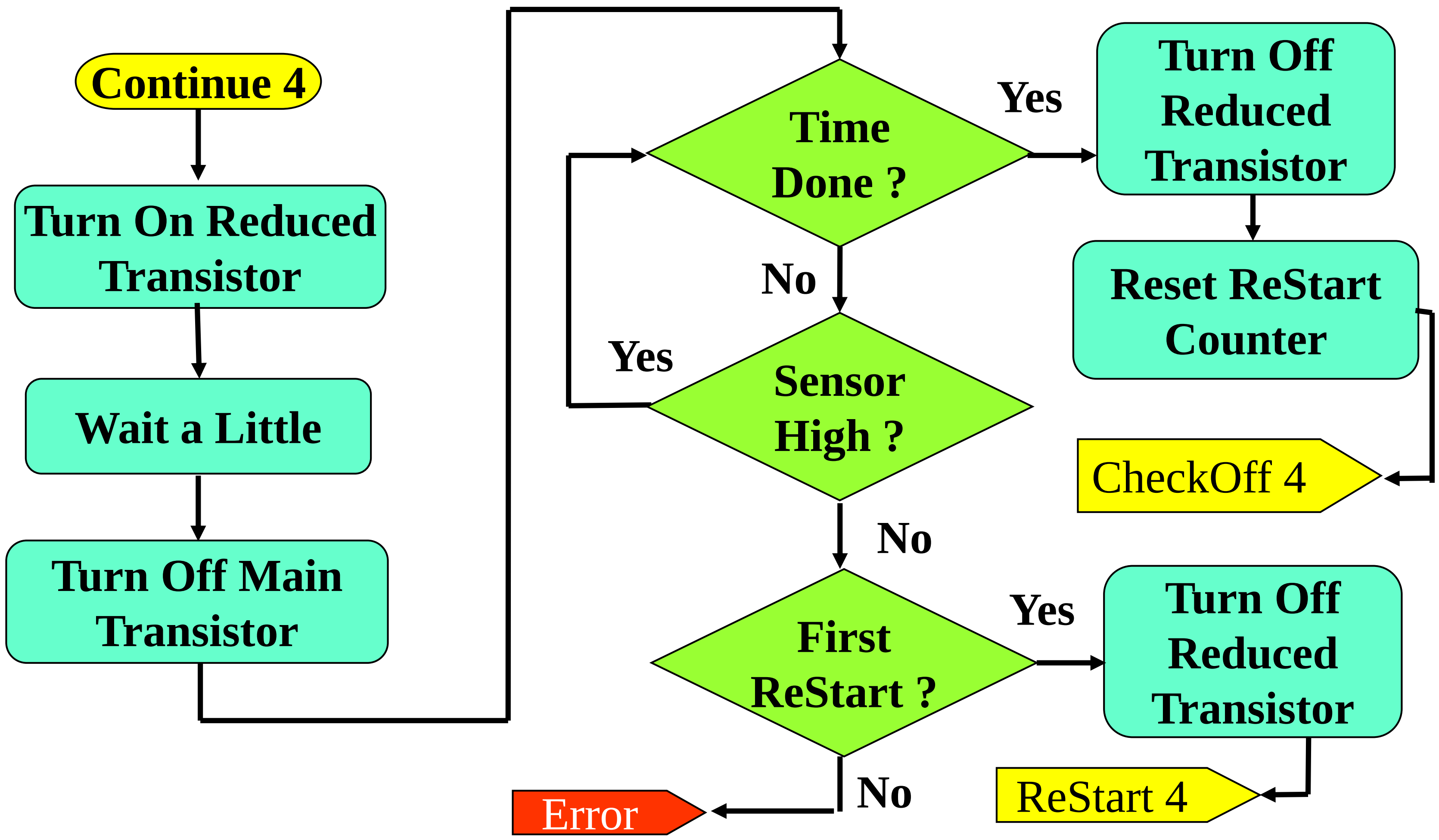

Mode 3
Red Press ?
Yes
No
Red Release ?
Yes
No
Turn Off Main Transistor
Turn Off LED
Green Press ?
Yes
GreenPress
No
Red Press ?
No
Yes
Red Release ?
No
Yes
Turn On LED
Read A/D For Value
= 0 ?
Yes
Error
No
Value > 70 hex?
Yes
No
Turn On Main Transistor
Turn Off Main Transistor

Mode 4

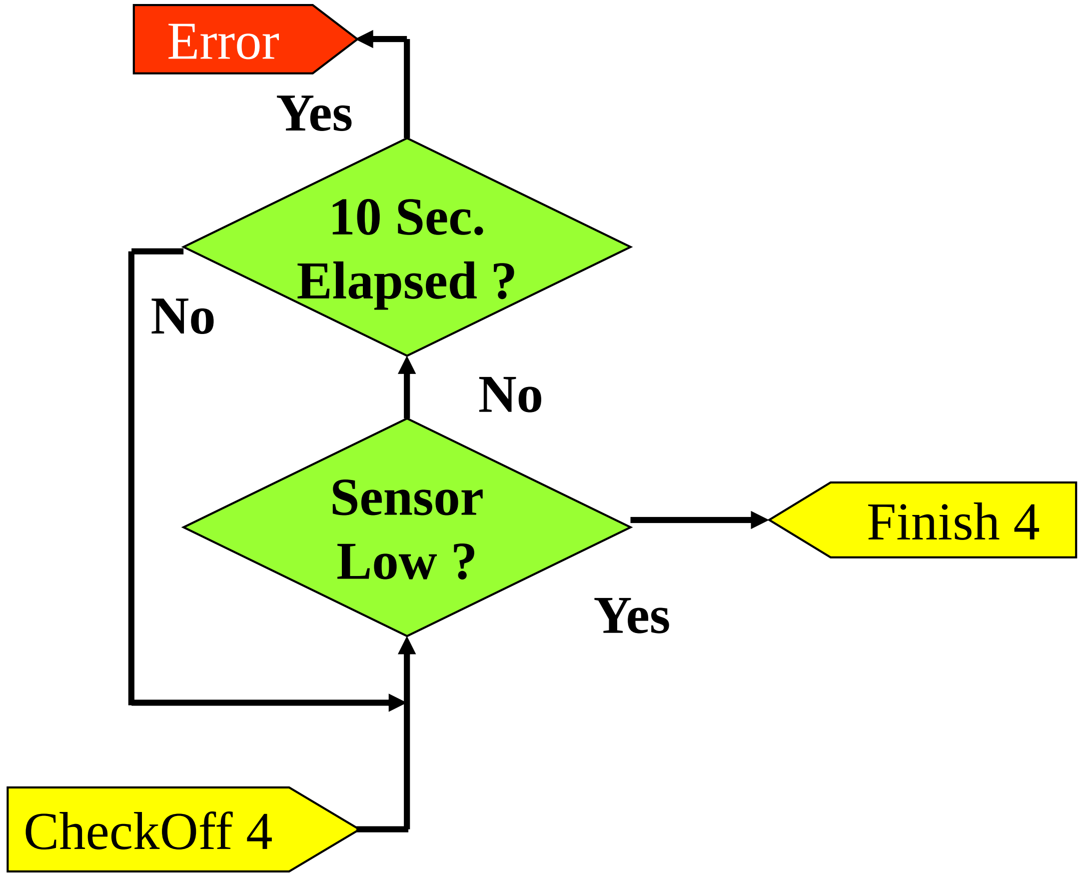


Mode 4
Finish 4
Greed Press ?
Yes
GreenPress
No
Red Press ?
No
Yes
Red Release ?
No
Yes
Read A/D For Time
= 0 ?
Yes
Error
No
Turn On Main Transistor
Restart 4
Sensor High ?
Yes
Continue 4
No
10 Sec. Elapsed ?
No
Yes
Error

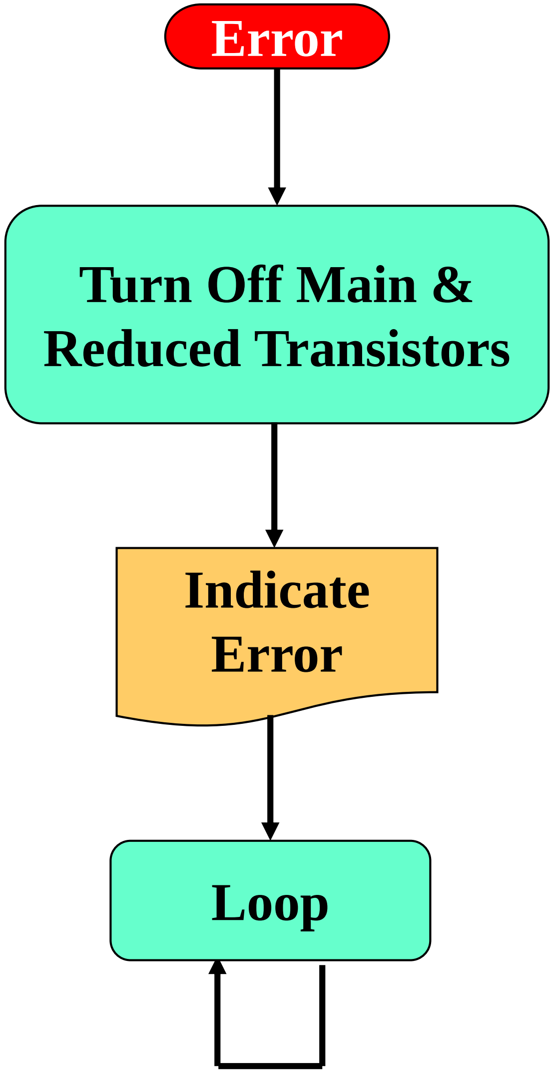
Mode 4 Continued



More Mode 4

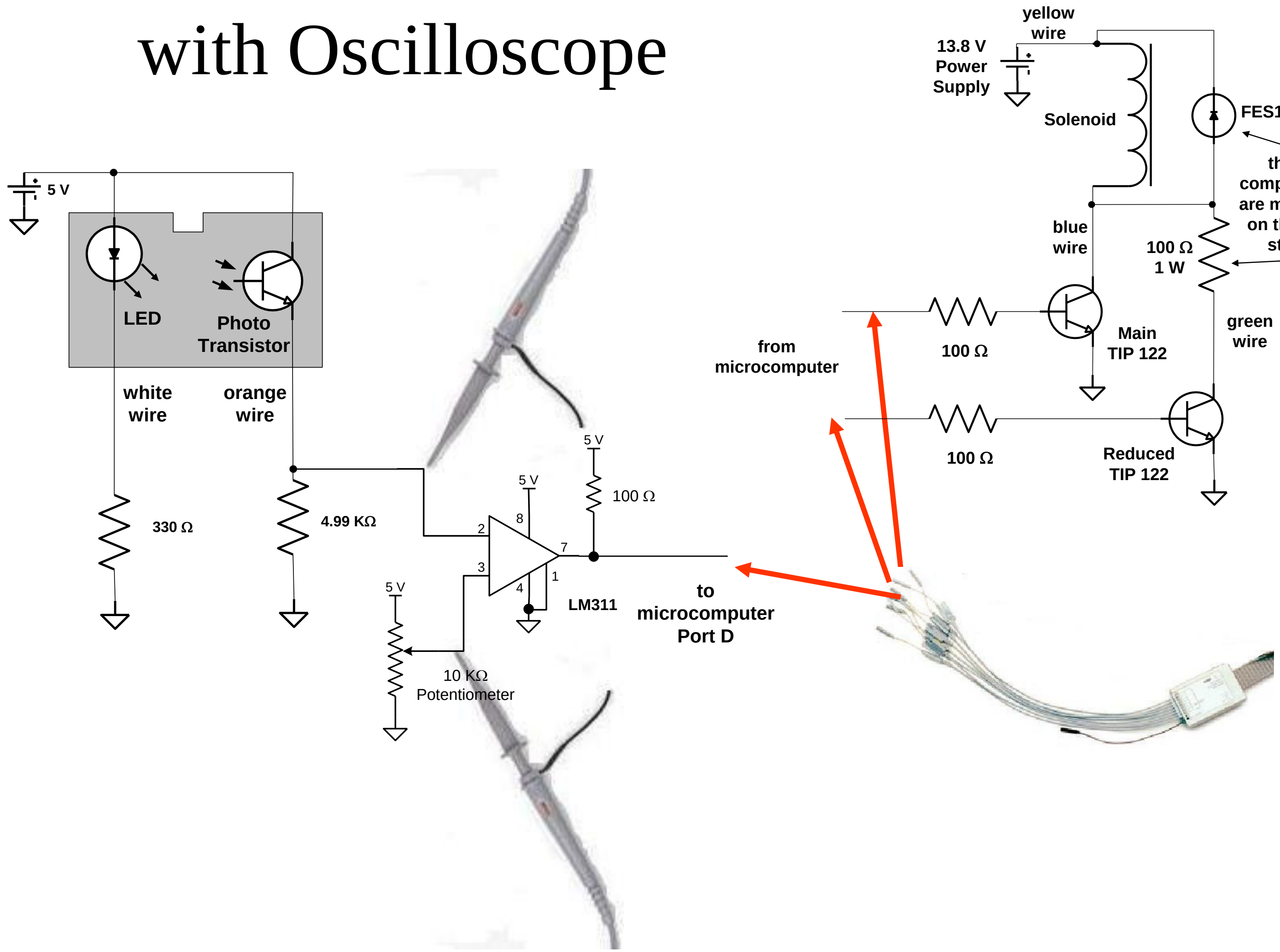

Error
Yes
10 Sec.
Elapsed ?
No
No
Sensor
Low ?
Finish 4
Yes
CheckOff 4

Error

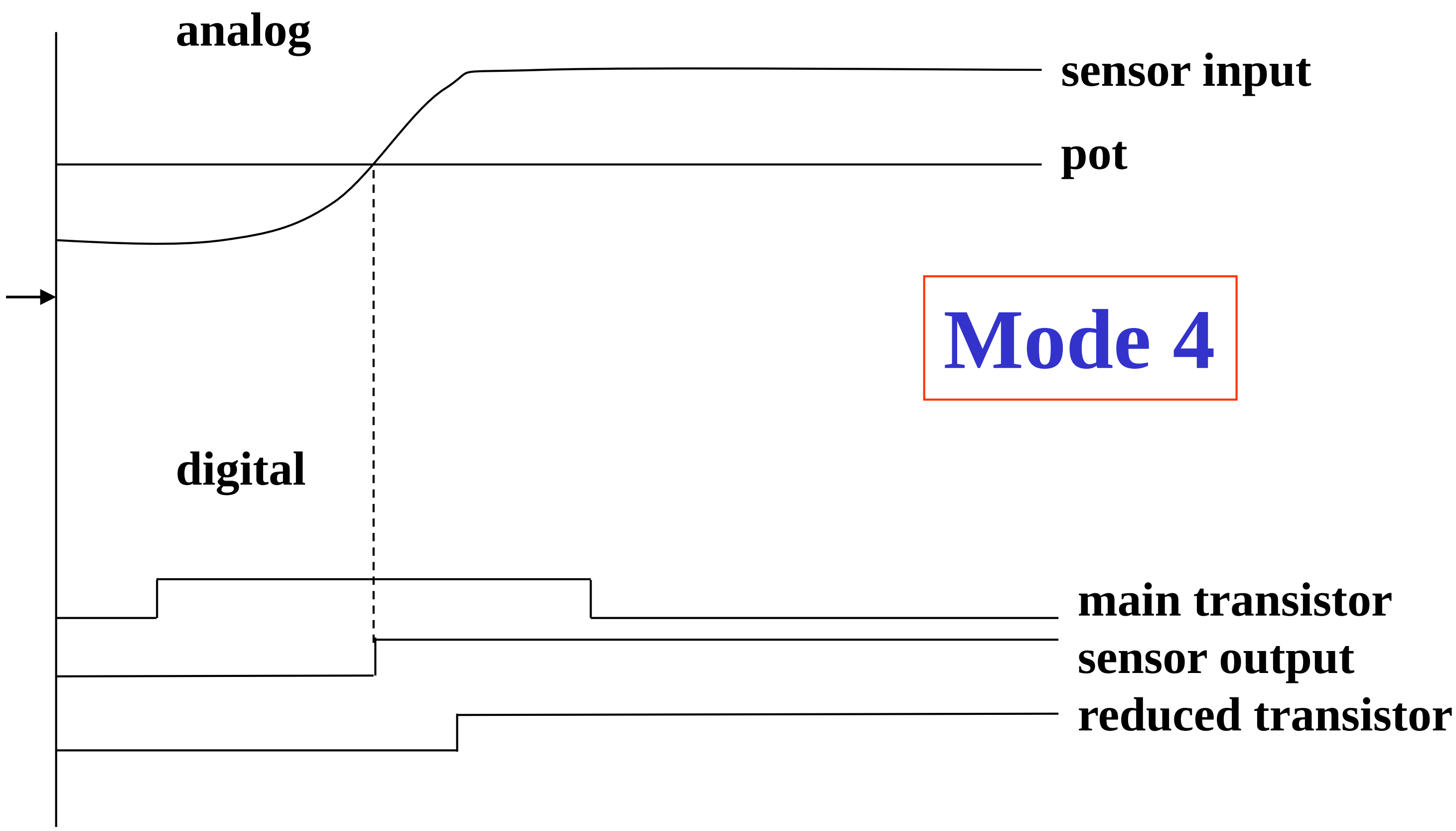


After a fault, the processor has to be reset (or powered off and on) to continue.

Measurement with Oscilloscope



Oscilloscope Traces



Grading

Grades for code case studies will be based on:

1	Successful execution of all required parts & bug free	(40 %)
2	Program Features	(50 %)
	Following the software standard	
	Liberal use of comments	
	Efficiency and creativity	
3	Answers to questions	(10 %)

Commonality

- There are many similarities between the modes.
- Subroutines that handle common elements should be written ONCE.
- For example: you do not want to program mode 2 and then start over again for modes 3 & 4
- Differences between the modes can be handled in the same routines.

Octal Switch

- Complement of value from octal switch as read on Port E is the mode in binary. This is the value that is output on the Port B LEDs in both normal and error conditions. True for all modes $\Rightarrow$ one routine
- For error condition, bit 3 of Port B is turned on.

```
Octal       EQU 30h

comf        PORTE, w
andlw       B'00000111
movwf       Octal
movwf       PORTB
```

For error:

```
bsf         PORTB,3
```

Mode

- There are only 8 modes 0 through 7
- Create a register to hold mode in 8 bits
- All bits are 0 except for bit that indicates mode

```
Mode    EQU    31h
```

For mode 2:

0	**0**	**0**	**0**	**0**	**1**	**0**	**0**
Mode 7	Mode 6	Mode 5	Mode 4	Mode 3	Mode 2	Mode 1	Mode 0

```
bsf     Mode,2

btfss   Mode,2
```

Common Timer

- Both mode 2 and mode 4 involve counting down the value from the potentiometer (A/D converter) in ¼ seconds
- In mode 2, you have to check for the red button and restart the timer
- In mode 4, you have to check the sensor and see if the solenoid failed
- Use bit test of register Mode

```
btfss    Mode,2

btfss    Mode,4
```

Common Errors

- Mode 2, Mode 3 and Mode 4 have an error if the value from the potentiometer (A/D converter) is 0
- One routine to check. Should be part of a single routine to read A/D converter.
- In mode 4, you have to check the sensor and see if the solenoid failed
- Use bit test of register Mode

```
btfss    Mode,2

btfss    Mode,4
```

Common Errors

- Mode 4 if you try to engage the solenoid and the sensor does not indicate that it engaged in 10 seconds, you declare Mode 4 to have an error. (looking for sensor hi)
- Similarly, if you try to disengage the solenoid and the sensor does not indicate that it disengaged in 10 seconds, you also declare Mode 4 to have an error. (looking for sensor lo)
- This should be one routine with 2 conditions – one when you engage the solenoid, one when you disengage.
- You can have a bit in the State register that you create

```
State       EQU   32h

bsf         State,0
btfss       State,0
```

Checking "*Greater Than*" or "*Less Than*"

- In Mode 3 you want to know if the value on the potentiometer is greater than or less than a number.
- **You do not care how much greater or how much less it is**

SUBWF Subtract W from f

Syntax:	[*label*] SUBWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) - (W) → destination
Status Affected:	C, DC, Z
Encoding:	00 | 0010 | dfff | ffff
Description:	Subtract (2's complement method) W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

```
Example 1:      SUBWF    REG1,1

  Case 1:       Before Instruction
                      REG1= 3
                      W   = 2
                      C   = x
                      Z   = x
                After Instruction
                      REG1= 1
                      W   = 2
                      C   = 1          ; result is positive
                      Z   = 0
  Case 2:       Before Instruction
                      REG1= 2
                      W   = 2
                      C   = x
                      Z   = x
                After Instruction
                      REG1= 0
                      W   = 2
                      C   = 1          ; result is zero
                      Z   = 1
  Case 3:       Before Instruction
                      REG1= 1
                      W   = 2
                      C   = x
                      Z   = x
                After Instruction
                      REG1= 0xFF
                      W   = 2
                      C   = 0          ; result is negative
                      Z   = 0
```

General Comments About on / off control

Many Mechatronic industrial applications involve on / off control. Why ?

- Low-Cost, Simple Actuators
  - Solenoids instead of linear actuators
  - Solenoid valves instead of proportional valves
  - AC (alternating current) motor contactors instead of variable frequency or vector drives
  - DC (direct current) motor contactors instead of amplifiers
- Low-Cost, Simple Sensors
  - Thermostatic switches instead of temperature sensors
  - Pressure switches instead of pressure transducers
  - Flow switches instead of flow transducers
  - Proximity switches instead of displacement sensors
- Low-Cost, Simple Control Components
  - Digital logic
  - Programmable Logic Controllers (PLC)

Some Mechatronic systems are inherently on / off

- HVAC (Heating Ventilating Air Conditioning) systems
- Pumps (sump pumps, bilge pumps)
- Cooling systems (radiator fan, solenoid valve)
- AC motor driven systems
- Power generating systems
- Hydraulic & pneumatic systems
- Manufacturing processes (stamping, folding, molding)

Very slow, very stable mechanical dynamics

On / Off Control

- What makes an on/off operation, on/off control?

a sensor

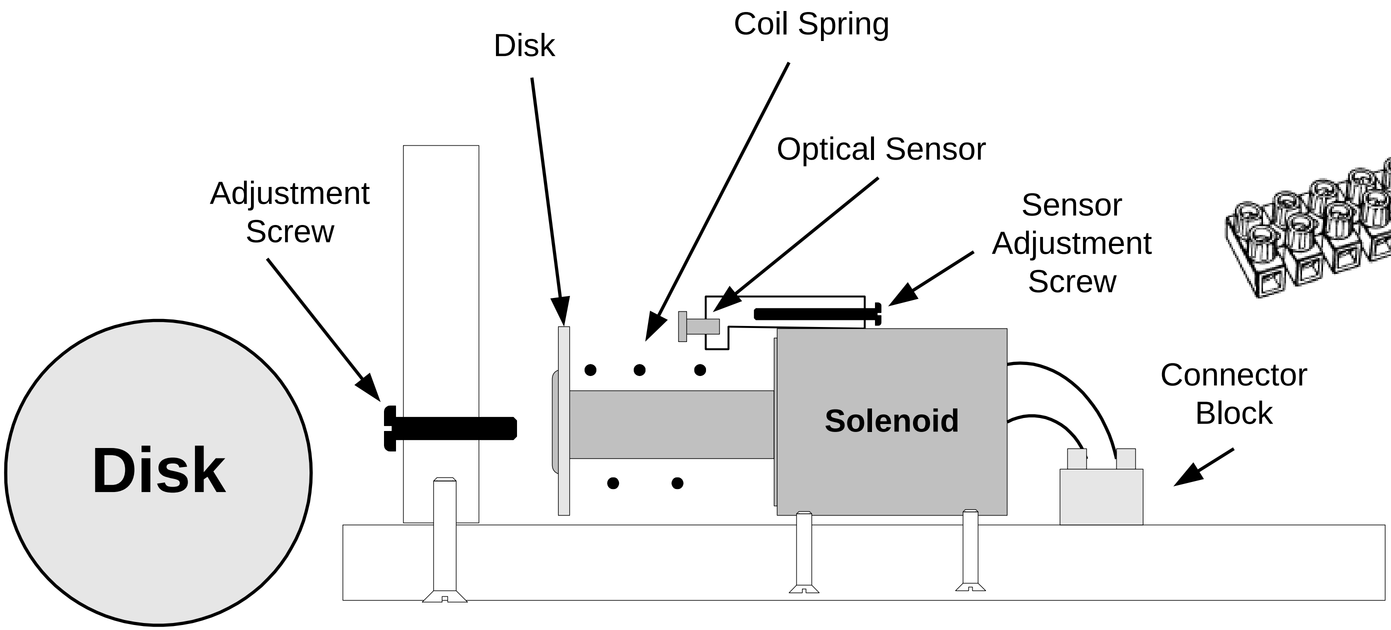
- When the actuator is turned on, the sensor determines if the process really went on.

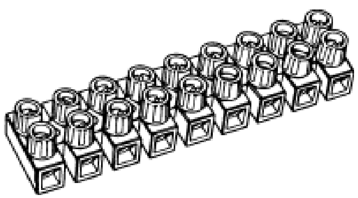
- When the actuator is turned off, the sensor determines if the process really went off.

The goal of this case study is to investigate aspects of using an embedded microcomputer for controlling on / off mechatronic systems.

- Control of a solenoid
  - First order mechanical dynamics
  - Slow process (in the microcomputer world)
  - Represents other on / off processes
- Optical displacement sensor (inexpensive sensing)
  - Used as a proximity switch
  - Feedback sensor
- Microcomputer functionality
  - Mode switching
  - User interfacing
  - Fault detection
  - Automatic recovery from faults
- Programming in Assembly
  - Microcomputer development system
  - Code simulation

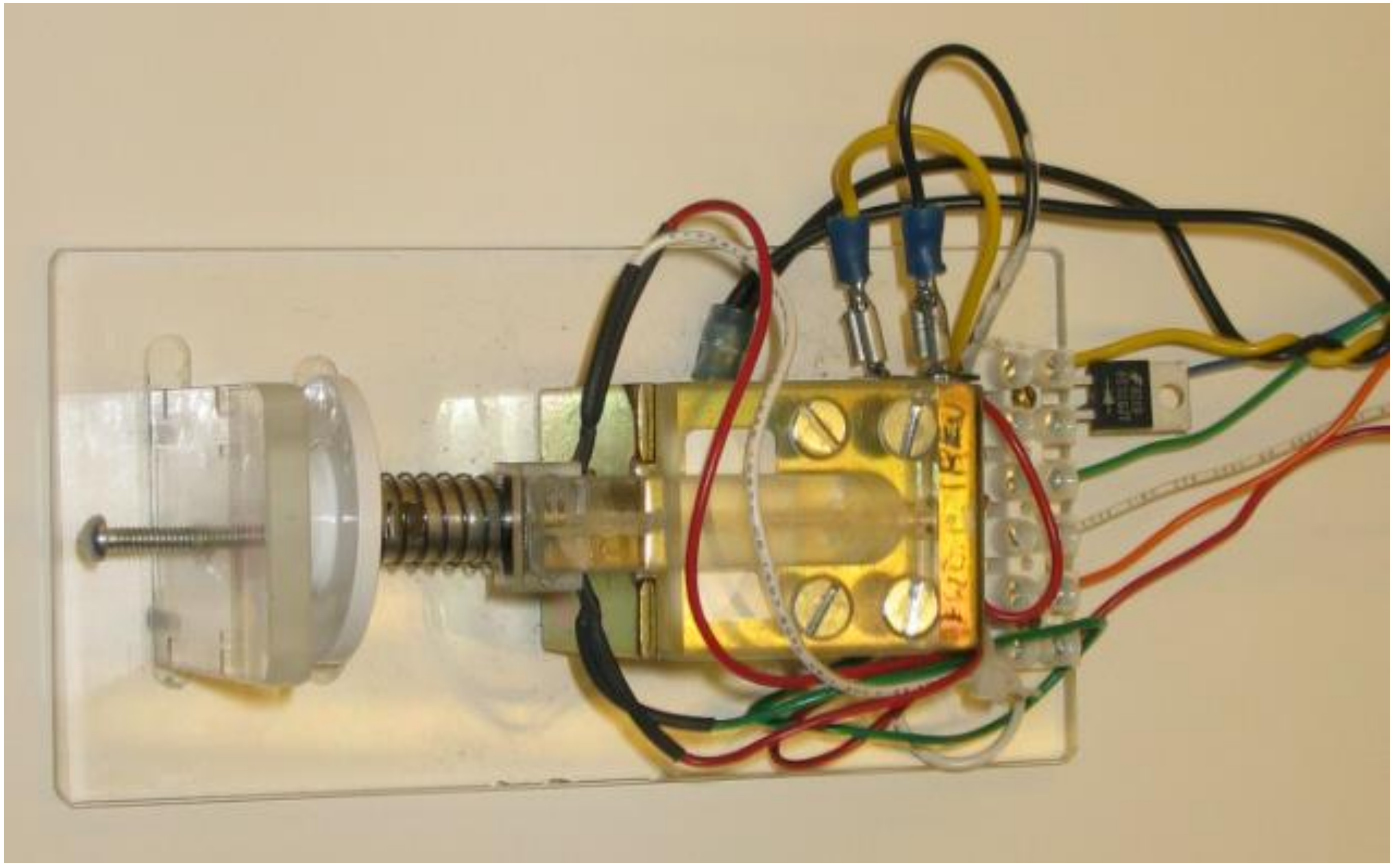
Test Fixture





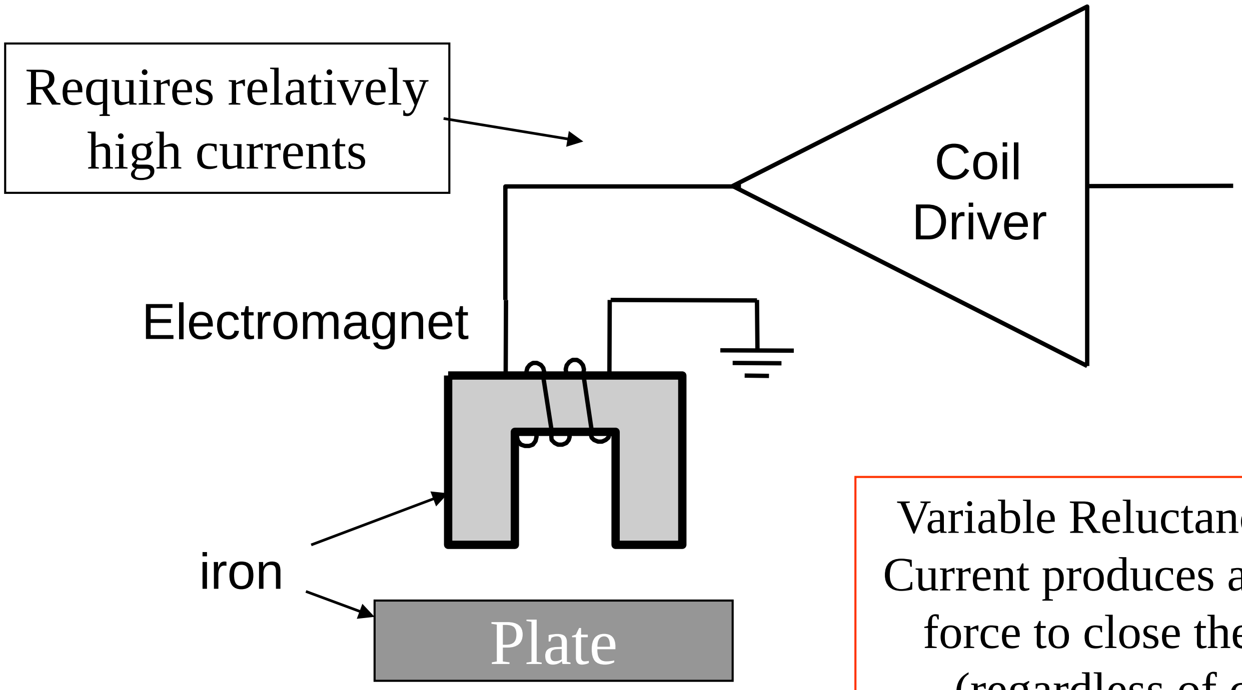
diode & resistor mounted on connector block

Test Fixture



Variable Reluctance Effect

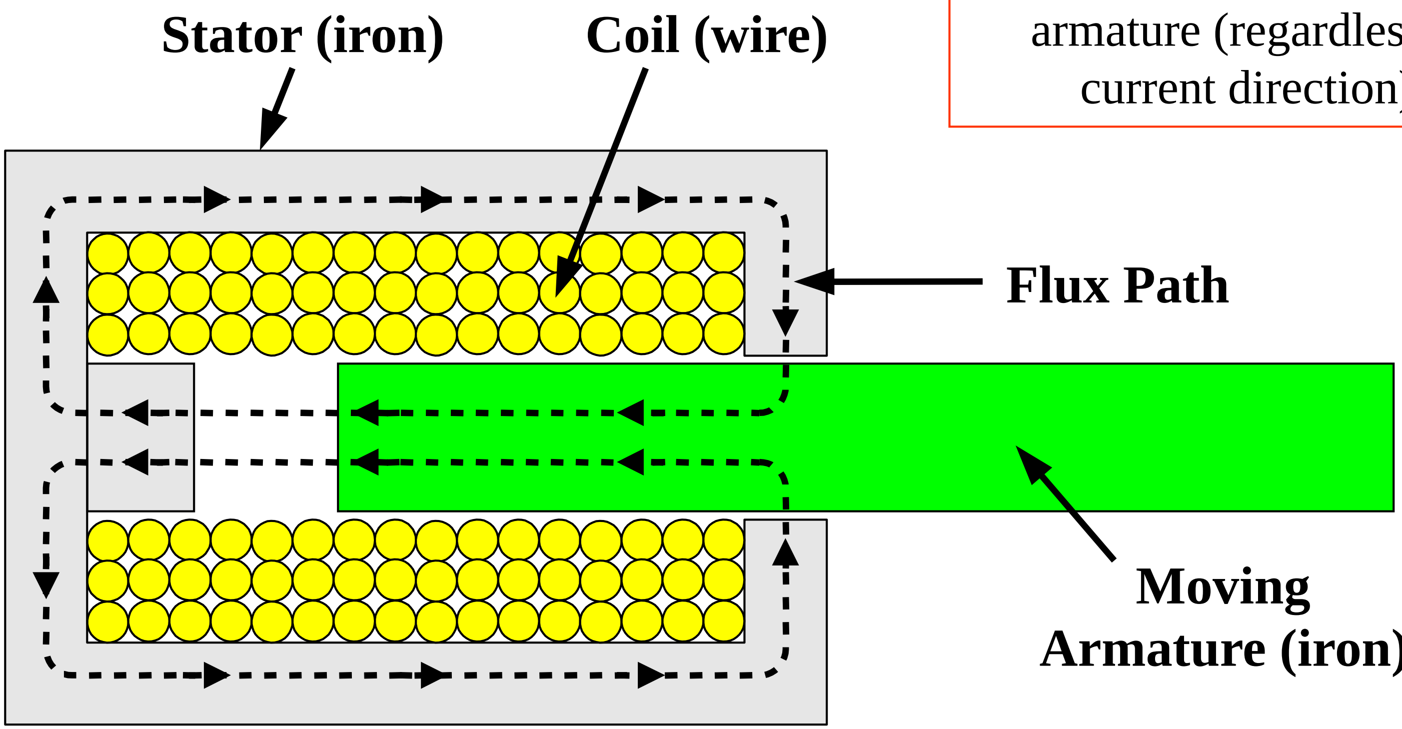
Used in Solenoids



Variable Reluctance Effect. Current produces a magnetic force to close the air gap (regardless of current direction).

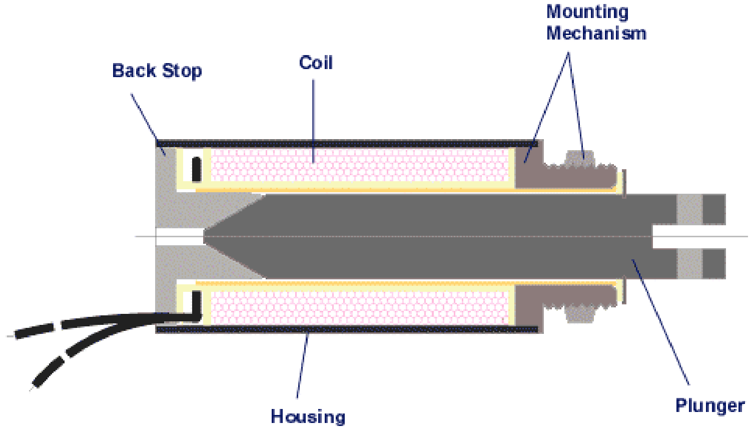
Solenoid Schematic

Variable Reluctance Effect. Force always retracts armature (regardless of current direction).

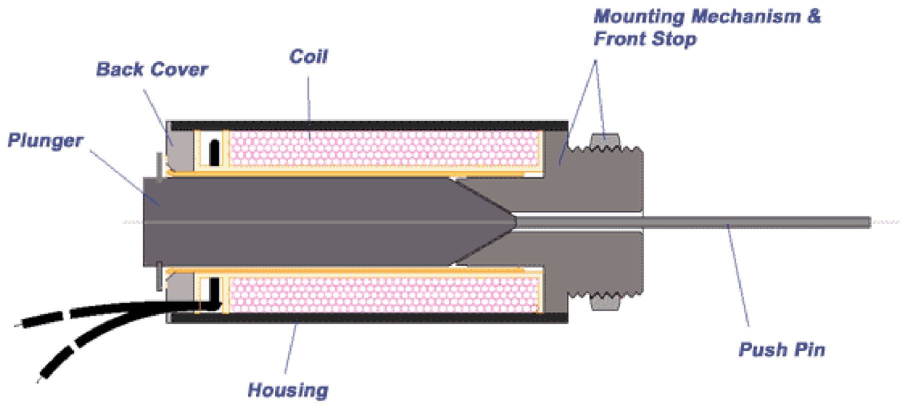


Current in the coil produces magnetic flux.
Armature moves to minimize the reluctance (minimize the air gap).

Push Solenoids



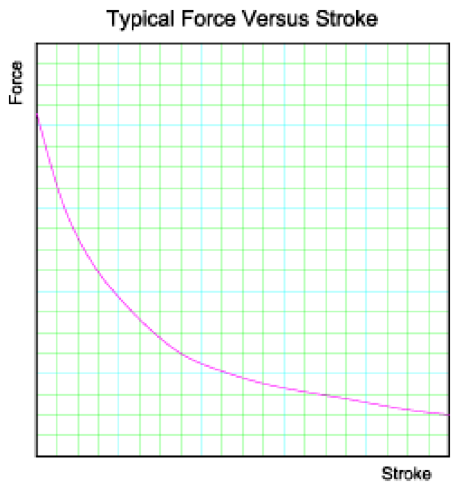
Pull Type Solenoid



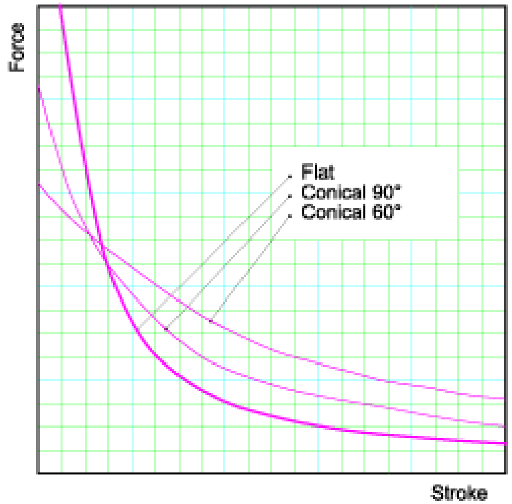
Push Type Solenoid

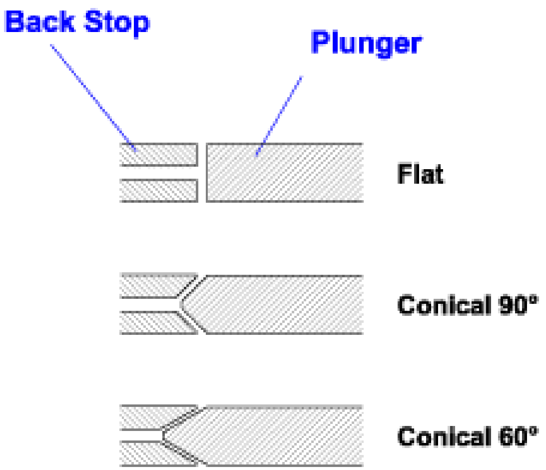
Push solenoids simply extend the armature through the stator.

Force vs. Stroke



For simple electromagnets, the force is inversely proportional to the square of the air gap.





Different shapes for solenoid armatures will result in different force vs. stroke curves. These are usually provided by the manufacturer.

Duty Cycle

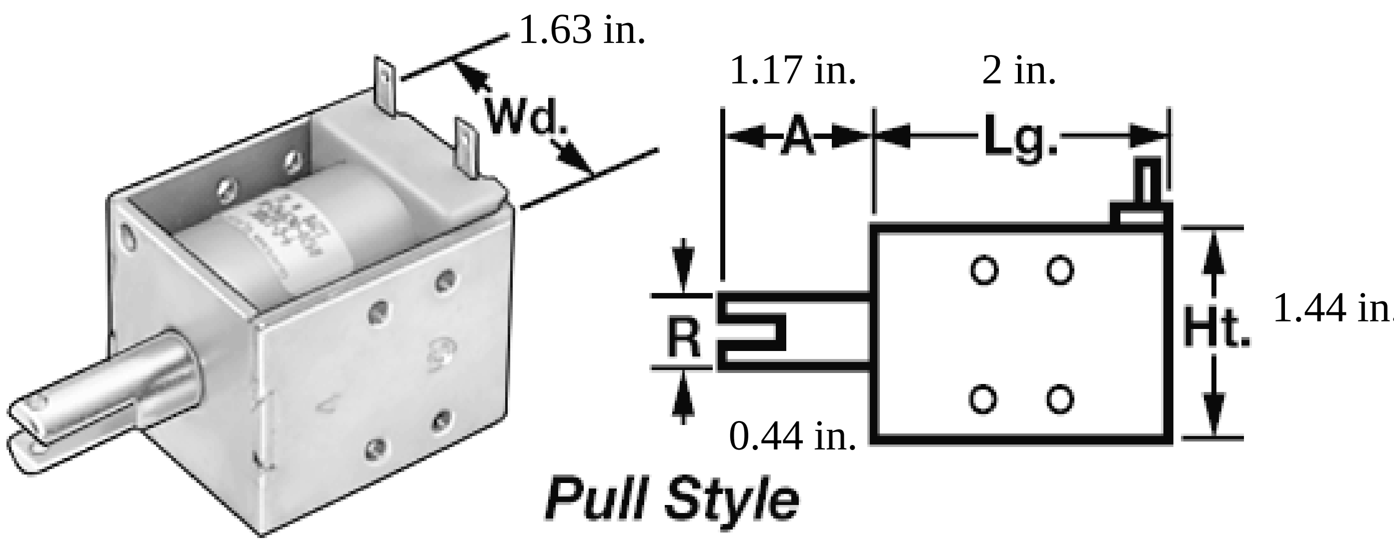
- Solenoids are also specified by their duty cycle where:

$$\text{Duty Cycle (\%)} = \frac{\text{"on" time}}{\text{"on" time} + \text{"off" time}}$$

- "Intermittent Duty" $\Rightarrow$ less than 100 % (can be very low)
- "Continuous Duty" $\Rightarrow$ 100 %

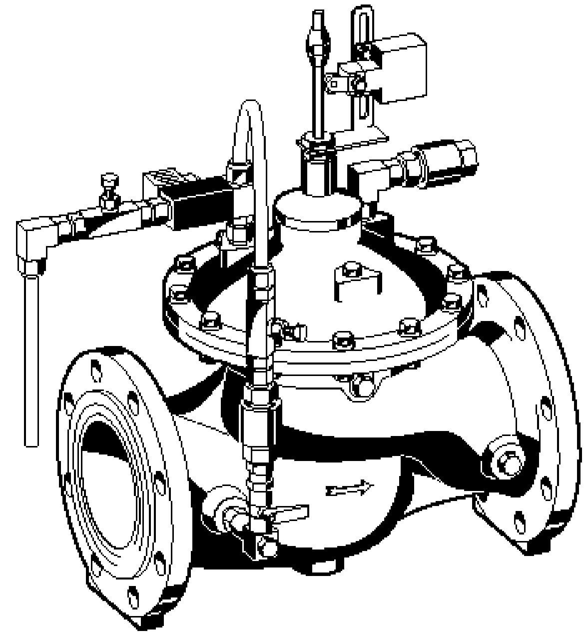
- Usually dictated by temperature rise in the coil from the power loss.
- Solenoid temperature is specified by "class"
  - Class A $\rightarrow$ 105^0 C max
  - Class $\rightarrow$ 130^0 C max
  - Class F $\rightarrow$ 155^0 C max
  - Class H $\rightarrow$ 180^0 C max

Case Study Solenoid



Continuous Duty	
Voltage:	12 V
Max. Stroke:	1 in.
Force at 1/8 in:	76 oz.
Power Rating:	11 W
Resistance	13.1 Ω

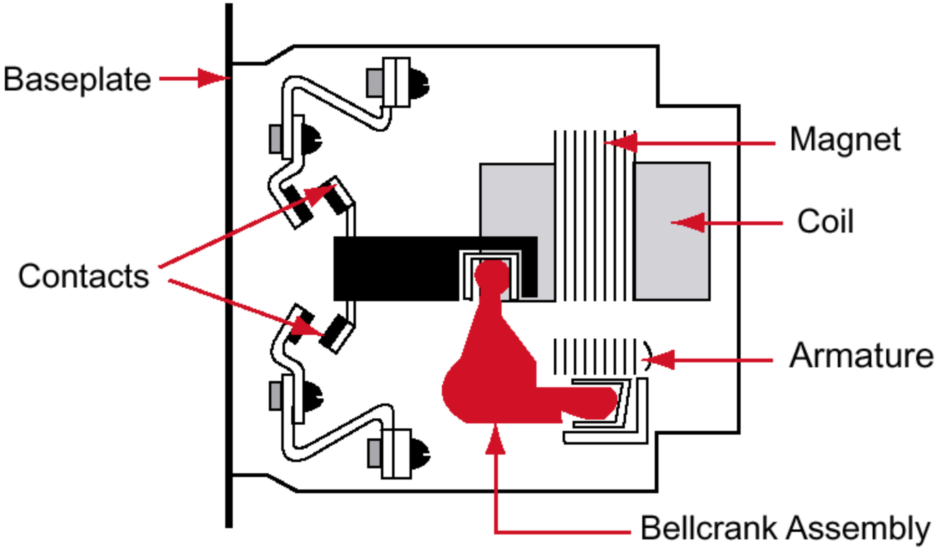
Solenoid Valves



Solenoid directly opens (or closes) the valve. Typical of valves for gas.

Solenoid shuttles fluid to open (or close) the valve. Typical of valves for liquids.

Motor Contactor



Capability Maturity Model
(Software Engineering Institute – Carnegie Mellon)

Five Levels:

1. **Initial** – ad hoc and chaotic. Few processes are defined. Success depends on individual rather than team effort.
2. **Repeatable** – intuitive. Basic process management processes are in place to track cost, schedule, functionality.
3. **Defined** – standard & consistent. Processes are documented. Standardized and integrated. All projects use an approved version of the organization's standard software process.
4. **Managed** – predictable. Detailed software process and product quality metrics establish a quantitative evaluation foundation. Variations in process performance can be distinguished from random noise.
5. **Optimizing** – characterized by continuous improvement. Organization has quantitative feedback systems in place to identify process weaknesses and strengthen them. Project teams analyze defects to determine causes. Software processes are evaluated.

Embedded Programming

- Software Improvement
  - Version Control
  - Software Standard
  - Breaking Up Code
- Code Inspections
- Track Errors
- Track Productivity

Software Improvement in This Course

- Version Control
  - Save a copy of code which works
  - Always copy code before making major changes
- Software Standard (provided)
  - All code written by a team should “look” the same
  - Consistent use of comments
  - Consistent formatting
  - Consistent names for subroutines
- Break up code into small parts
  - Collect common operations into subroutines
  - Delineate modes

Code Inspections

- Prior to Testing the Code
- After first clean compile

Team

- Moderator – leads the inspection process
- Reader – goes through the code paraphrasing the objective
- Recorder – takes notes on the errors (on a standard form)
- Author – contributes nothing except to answer questions and clarify unclear areas. Understands errors.