

# Workshop 3: Program Flow

**Purpose:** Using the embedded controller to produce a programmed output (LED indication and sound) basic on the inputs.

**Objective :** This workshop is designed to familiarize the student with elements of the embedded microcontroller.

**Real-World Application:** Home security system.

**Requirements:**

1. Hardware:

- Picstic-3
- Three red LEDs, One green LED
- 8 ohm / .2 watt speaker

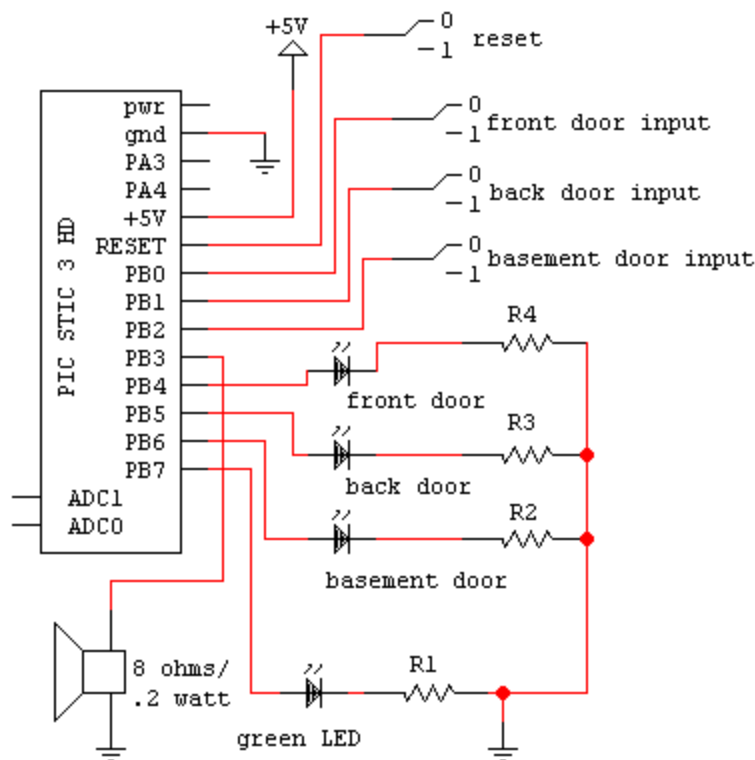
2. PBasic Software Commands

- Sound
- Toggle
- Pin(s)
- Dirs

**Procedure:**

Mount the LEDs vertically on the photoboard with the three red LEDs on top and the green on the bottom. Using a DIP switch as door inputs, program a security system that checks the back door, front door and basement doors locked. When the door is shut and locked the input to the Pic is a logic high and the corresponding red LED is lit, if the door is open then the input is a logic low and the corresponding red LED is off. When all the doors are shut the program flashes the red LEDs on and off for five seconds, leaving them off and lights the green LED. The system continuously monitors the door inputs, if any door is opened the green LED goes off, an audible alarm is sound and light the corresponding red LED is lit. Enable and disable the system using the Pic reset (via DIP switch).  
Hint: To prevent false alarms at start up the program should verify all doors are lock prior to going into the monitoring mode.

## Circuit Drawing for Workshop # 3



## PicBasic Code for Workshop #3

```

setup: low 4: low 5: low 6
start: b4 = pin0 'read pin 0 - check front door
b0 = b4& %00000001 'clear all bits except bit 1put into b0
if b0 = %00000001 then lock_1_is_set 'jump if front door is locked
goto start 'recheck front door
check2: b5 = pin1 'read pin 1 - check back door
b1 = b5&%00000001 'clear all bits except bit 1put into b1
if b1 = %00000001 then lock_2_is_set 'jump if back door is locked
goto check2 'recheck back door
check3: b6 = pin2 'read pin 2 - basement door
b2 = b6&%00000001 'clear all bits except bit 1put into b2
if b1 = %00000001 then lock_3_is_set 'jump if basement is locked
goto check3 'recheck basement door
lock_1_is_set:
high 4 'front door is locked light LED
goto check2 'now check back door
lock_2_is_set:
high 5 'back door is locked light LED
goto check3 'now check basement door
lock_3_is_set:
high 6 'basement door is locked light LED
alarm_is_set: 'flashes LEDs to signal that the doors
'are locked and the alarm is set
for b7=1 to 10
toggle 7 :toggle 6 : toggle 7: pause 500
next b7
low 4 'turn off back door LED
low 5 'turn off front door LED
low 6 'turn off basement door LED
high 7 'turn on system set LED
check_doors: 'final check before setting alarm
b9=pins
b8=b9&%00000111
if b8 = %00000111 then monitor_doors
goto setup
monitor_doors:
b9=pins 'read all pins into b9
b8=b9&%00000111 'clear all bits except first three (doors)
if b8 <>%00000111 then intruder_alarm
'if door(s) are unlocked sound alarm
goto monitor_doors
'recheck doors
intruder_alarm: 'lights the red LED for the door that has been
'opened, creates alarm tone on speaker and
'turns off green LED
low7: b1=0: b2=0
if b8=%00000110 then lit_front
if b8=%00000101 then lit_back
if b8=%00000011 then lit_basement
goto sounds
lit_front: low 4
goto sounds
lit_back: low 5
goto sounds
lit_basement: low 6
sounds: for b1=1 to 21 step 5
b2= b1 * 10
sound 3,(b1,20,b2,40)
pause 100
next b1
goto setup

```