

Workshop 2: Digital Input and Output

Purpose: Using the embedded controller to produce a programmed output (LED indication).

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

Real-World Application: Industrial equipment control system.

Requirements:

1. Hardware:

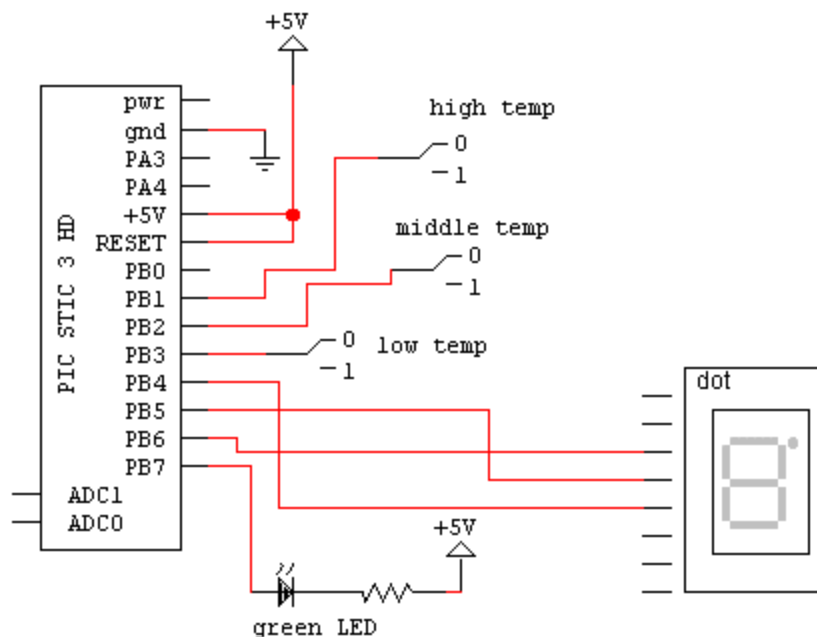
- Picstic-3
- Seven segment LED
- Green LED
- DIP switch

2. PBasic Software Commands:

- Branch
- If / Then
- For / Next
- Toggle

Procedure: The temperature control is vital to the outcome of the product. Design a circuit that monitors the temperature level of the product during manufacture. As the temperature increases to the three predetermined levels the program will light the horizontal bars on the seven LED and set off a visual alarm. Use a DIP switch in place of thermistor to indicate that a temperature is reached. Operate the switch to place a logic high on the input pin (PB0-7). When a temperature level is reached (switch is operated) the program should light the correct horizontal bar on the seven segment LED, then flash the green LED on and off for 100 times for .5 seconds - leaving the green LED off at the end of the sequence. The seven segment horizontal bar should light from the bottom up as the temperatures are achieved.

Circuit Drawing for Workshop #2



PicBasic Code for Workshop #2

setup:

low 7: low 6: low 5

start: b5 = 0 'clear byte 5

b5 = pin3 'sample 1st temperature input at

'pin 0 put in b5

if b5 = 1 then set0 'test fluid temperature 1

```

low 6: low 5: low 4 'turn off all LEDs if fluid
'temperature 1 has not been reached
goto start 'resample temperature 1
next1: b3 = 0 'clear byte 3
b3 = pin2 'sample 2nd temperature input at
'pin 1 put in b3
if b3 = 1 then set1 'test fluid temperature 2
low 5: low 4 'turn off remaining LEDs if fluid
'temperature 2 has not been reached
goto next1 'resample temperature 2
next2: b1 = 0 'clear byte 1
b1 = pin1 'sample 3rd temperature input at
'pin 2 put in b1
if b1 = 1 then set2 'test fluid temperature 3
low 4 'turn off last LEDs if fluid
'temperature 3 has not been reached
goto next2 'resample level 3
set0: b4 = 0 'set b4 to 0 for branching
goto test 'start branching
set1: b4 = 1 'set b4 to 1 for branching
goto test 'start branching
set2: b4 = 2 'set b4 to 2 for branching
goto test 'start branching
test: branch b4, (fluid_lvl_1,fluid_lvl_2,fluid_lvl_3)
'depending on the value of b4 ... branch
'to the correct temperature alarm
fluid_lvl_1: high 6 'turn on temperature 1 alarm LED
gosub alm 'visual alarm for temperature 1
goto next1 'start sampling the next
'temperature
fluid_lvl_2: high 5 'turn on temperature 2 alarm LED
gosub alm 'sound alarm for temperature 2
goto next2 'start sampling the next
'temperature
fluid_lvl_3: high 4 'turn on temperature 3 alarm LED
gosub alm 'sound alarm for temperature 3
goto end 'process complete
alm: 'sound alarm if correct fluid temperature
'has been reached
for b2 = 1 to 100
high 7: pause 500: low 7
next b2
return
end

```