**Sierra College CIE-01**

**Jim Weir 530.272.2203**

[**jweir43@gmail.com**](mailto:jweir43@gmail.com)

**www.rstengineering.com/sierra**

**Theory 11**

**Intermediate BASIC Programming for a PIC16F688**

1. **The IF-THEN-ELSE Command**
2. Sometimes we want the program to do a particular job if another part of the program is at a certain condition and do something else if that program is NOT at that condition. For example, if variable BIRTHDAY is 16 we want LAMP16 to light and if it is not to light the LAMP0. We would write the program:

IF birthday = 16 THEN LAMP16 = high

ELSE LAMP0 = high

1. The PBP Reference Manual (C:/Program Files/PBP3\_1/PBP3/Documentation/PBP\_Reference\_Manual.pdf on page 164 has a fairly good explanation of this command. Please find the manual and look at this page.

.

1. We are going to use this command to initiate the BLINK program we wrote last week. This will require us to manufacture a pushbutton and rewrite the BLINK program with something I call the PUSHBUTN program It looks something like this:

' Name : PUSHBUTN jw C2.bas

' Compiler : PICBASIC PRO Compiler 3.1

' Assembler : MPLAB X IPE v5.15

' Target PIC : 16F688

' Hardware : Lab Protoboard

' Oscillator : Internal 4 MHz.

' Keywords : LED

' Description : PICBASIC PRO program to light an LED connected

' through a resistor to PORTC.4

' if a pushbutton is pressed on PORT C.2.

' PORTC.4 is pin 6 of the 16F688

' **PORTC.2 is pin 8 of the 16F688**

#CONFIG

\_\_config \_INTRC\_OSC\_NOCLKOUT & \_WDT\_ON & \_MCLRE\_ON & \_CP\_OFF

#ENDCONFIG

**ANSEL = %00000000**

TRISA = %00001000

**TRISC = %00000100**

L1 var PORTC.4 ' Alias PORTC.4 to LED

**pushbutn:**

**IF PORTC.2 = 0 THEN ;Button is pushed,**

**PAUSE 50 ;debounce sensitivity**

**IF PORTC.2 = 0 THEN ;Button is still pushed, debounced**

**GOTO mainloop ;If the button is still pushed, go to mainloop**

**ENDIF**

**ENDIF**

**PAUSE 100 ;Wait 100 milliseconds for things to settle down**

**GOTO pushbutn**

mainloop:

High L1 ' Turn port C.4 to 5 volts to light a LED connected

' through a resistor to C.4

Pause 500 ' stop executing for half a second

low L1 ' Turn the led off

pause 500 ' stop executing for half a second

Goto mainloop ' Go back to mainloop and blink LED forever

End

1. We should note a few things:
   1. We can use any port on the A register or the C register that we wish. I arbitrarily said that I wanted to use PORTC.2
   2. Since the ‘688 is both an analog and a digital device, depending on how we want to use it, I had to disable the analog ports in order to get an input for the switch. I did this by using the ANSEL command which set all of the ports to digital.
   3. I also set PORTC.2 to an input with a 1 in the C.2 register: **TRISC = %00000100**
   4. As you will see in “Applications” I forced C.2 high with a 1 kilohm resistor.
   5. Then I started the **pushbutn:** label.
   6. A switch is going to be connected between C.2 (pin 8) to ground.
   7. IF that switch is pushed, the next line is executed, which is wait 50 milliseconds
   8. The look to see if C.2 is still at ground. This is called “debouncing” the switch
   9. IF the switch is still at ground THEN I go to mainloop to start the light flashing
   10. IF the switch is not at ground, then I end BOTH of the IF commands, wait a hundred milliseconds for things to settle down, and then go back to pushbutn to wait for the switch closure..

**GO TO APPLICATION 12 TO PUT THIS NEW PROGRAM INTO YOUR 688 MICROCONTROLLER.**

1. **THE FOR-NEXT Loop Command**
2. If we want the program to loop around a particular part of the program for a specific number of times, we use the FOR-NEXT command.
3. The basic form of the command is FOR variable = number1 TO number2
4. Number2 has to be greater than number1.
5. At some point in the program where you want to loop through the program again, there has to be a NEXT command that goes back to the FOR command, adds one to the variable, and then goes through the program again. And again, until the variable equals number2, and the next command after the NEXT command is executed.
6. There is a variation of this command that uses the STEP command in the FOR line to increment OR decrement the variable by some number other than one. For example, FOR variable = 10 to 1 STEP -1 would start the loop off at 10 and the next loop would be 9, then 8 and so forth until it equals the variable to zero, which then lets the program go past the NEXT command.
7. Example:

ANSEL = %00000000

TRISA = %00001000

TRISC = %00000100

L1 var PORTC.4 ' Alias PORTC.4 to LED

**F1 var BYTE ' F1 is the FOR-NEXT Variable**

pushbutn:

IF PORTC.2 = 0 THEN ;Button is pushed,

PAUSE 50 ;debounce sensitivity

IF PORTC.2 = 0 THEN ;Button is still pushed, debounced

GOTO mainloop ;If the button is still pushed, go to mainloop

ENDIF

ENDIF

PAUSE 100 ;Wait 100 milliseconds for things to settle down

GOTO pushbutn

mainloop:

**FOR F1 = 1 to 10 ' Loop to go through mainloop 10 times**

High L1 ' Turn port C.4 to 5 volts tolight a LED connected

' through a resistor to C.4

**Pause (500 - F1\*45) ' stop executing for a while**

low L1 ' Turn the led off

**pause (500 - F1\*45) ' stop executing for a while**

**Next F1 ' add 1 to F1 and go around again**

**Goto pushbutn ' Go back to pushbutn and wait**

**GO TO APPLICATION 12 TO PUT THIS NEW PROGRAM INTO YOUR 688 MICROCONTROLLER.**

1. **The GOSUB RETURN command**
2. This is the easiest one of all. If I have a routine that I’m going to use in more than one place, or more than once in any location I’ll write it as a GOSUB.
3. The GOSUB command simply says, “Go to a label, execute that label, and at the end of it, RETURN right where you left off.
4. In this program I’m going to have a second LED that simply blinks when nothing else is going on. In particular, I’m going to blink an LED if the FOR-NEXT loop isn’t being run, which means that the pushbutton has not been pushed.
5. In programming slang this is called a NANNY circuit. It bugs you to do something to turn off the annoying blinking light.
6. Program follows:

‘ PORTC.4 is pin 6 of the 16F688

' PORTC.2 is pin 8 of the 16F688

**' PORTC.3 is pin 7 of the 16F688**

#CONFIG

\_\_config \_INTRC\_OSC\_NOCLKOUT & \_WDT\_ON & \_MCLRE\_ON & \_CP\_OFF

#ENDCONFIG

ANSEL = %00000000

TRISA = %00001000

TRISC = %00000100

L1 var PORTC.4 'Alias PORTC.4 to L1

**L2 var PORTC.3 'Alias PORTC.3 to L2**

F1 var BYTE 'F1 is the FOR-NEXT Variable

pushbutn:

IF PORTC.2 = 0 THEN ;Button is pushed,

PAUSE 50 ;debounce sensitivity

IF PORTC.2 = 0 THEN ;Button is still pushed, debounced

GOTO mainloop ;If the button is still pushed, go to mainloop

ENDIF

ENDIF

**GOSUB LED2**

GOTO pushbutn

mainloop:

FOR F1 = 1 to 10 ' Loop to go through mainloop 10 times

High L1 ' Turn port C.4 to 5 volts tolight a LED connected

' through a resistor to C.4

Pause (500 - F1\*45) ' stop executing for a while

low L1 ' Turn the led off

pause (500 - F1\*45) ' stop executing for a while

Next F1 ' add 1 to F1 and go around again

Goto pushbutn ' Go back to pushbutn and wait

**LED2:**

**Pause 100 'wait a bit**

**High L2 'Turn LED #2 on**

**Pause 100 'Wait a bit**

**LOW L2 'Turn LED #2 off**

**Return 'Go back to the program**

End

1. **I want you to experiment to the end of this session how you can make the LEDs do what you want. You have all the tools you need. I want to see some creative genius and some experiments that just didn’t work the way you expected them to.**

**I’m happy to program your problem children if you will bring me your hex files.**