

```
DEFINE OSC 8
OSCCON.4 = 1
OSCCON.5 = 1
OSCCON.6 = 1
```

```
ANSEL = 0
ANSEL.0 = 1
```

```
led      var PORTB.0           'LED for checking PIC status
buzzer   var PORTB.4           'Buzzer for audio
voltage  var PORTA.0           'Read voltage line from resistor array on battery
serial_in var PORTA.1           'Recieve line for serial communication with Arduino
serial_out var PORTA.2         'Send line for serial communication with Arduino
voltage_val var byte           'Variable for storing voltage value
baud_rate con 2                 'Communication rate with Arduino set to 9600
serial_val var byte            'Variable for storing value sent serially from Arduino
```

```
'When the PIC first starts up...
```

```
pause 1000           'Pause for 1 sec
Gosub Blink           'Blink the LED three times
gosub Blink
gosub Blink
pause 1000           'Pause for 1 sec
```

```
'Continuously loop
```

```
start:
```

```
    adcin voltage, voltage_val           'Check voltage of battery
    if (voltage_val < %11011100) then    'If battery voltage is less than 11V (corresponding to a value
of 220 from A/D converter)...
        freqout buzzer, 2000, 20000     'Sound the buzzer for 2 sec at 20000 Hz
        pause 1000                       'Pause for 1 sec
    endif
```

```
    serin serial_in, baud_rate, serial_val           'Check serial line from Arduino
    if (serial_val > %00100000) && (serial_val < %10000000) then 'If the PIC recieves a value between
64 and 256 from the Arduino (expected value of 128)
```

```
        adcin voltage, voltage_val           'Check the voltage of the battery
        serout serial_out, baud_rate, [voltage_val] 'Send the value of the voltage to the
Arduino
```

```
        gosub Blink                           'Blink the LED to confirm
that the PIC sent the value
```

```
        elseif (serial_val > %00001000) && (serial_val < %00100000) then 'If the PIC recieves a value between
16 and 64 from the Arduino (expected value of 32)
```

```
            freqout buzzer, 2000, 15000     'Sound the buzzer for 2
sec at 15000 Hz
```

```
        endif
        goto start
    end
```

```
'Blink function
```

```
Blink:  
  High led  
  pause 250  
  low led  
  pause 250  
Return
```