

```
' *****
' * HelloWorld.bas *
' * Version 1.0 *
' * PIC18F452 - 40 MHz *
' *****

Device = 18F452

OPTIMISER_LEVEL = 9

XTAL = 40

PORTA = %00000000
PORTB = %00000000
PORTC = %00000000
PORTD = %00000000
PORTE = %00000000

TRISA = %00000000
TRISB = %00000000
TRISC = %00000000
TRISD = %00000000
TRISE = %00000000

' *****

'ADPCM compression decoder var's '16 Bit Max Min

Dim dPRED_SAMPLE_DE As DWord '33270 -34572
Dim dINDEX_DE As DWord '80 -1
Dim dSTATE_PREV_SAMPLE_DE As DWord '32767 -32767

Dim ySTATE_PREV_INDEX_DE As Byte '80 0
Dim yCODE_DE As Byte
Dim wDIFFQ_DE As Word '13372 0
Dim wSTEP_DE As Word '15289 0
Dim wPRED_SAMPLE_DE_TEMP As Word '34572 0
Dim iHIGHBYTE_LOWBYTE_DE As Bit
Dim yBYTE_2_4BITS_DE As Byte
Dim iINTERRUPT_SET As Bit
Dim dTIMERPRELOAD As DWord
Dim wTIMER1 As TMR1L.Word 'Access 16 bits of TMR1
Dim ySAMPLE_RATE As Byte
Dim wSOUNDADDRESS As Word

' *****

RCON.7 = 1 '0 Disables priority levels on
interrupts (16F compatability)

T1CON = %10000000 'Bits XX OF %10XX0001 = 00 = 1:1
Prescale 'T1CON.0 = 0 '0 = Timer 1 off

PIE1 = %00000001 'Enable TMR1 overflow interrupt
INTCON = %11000000 'Enable global and peripheral
interrupts

On_Hardware_Interrupt GoTo HARDWARE_INTERRUPT_ROUTINE '425 OK '4520 Not OK

GoTo start 'Jump past subroutines

' *****

HARDWARE_INTERRUPT_ROUTINE: 'Assembly hardware TMR 1 interrupt
```

```

iINTERRUPT_SET = 1

bcf PIR1,0                               'Clear interrupt flag and exit...
wTIMER1 = dTIMERPRELOAD
retfie fast                               'End assembly hardware TMR 1 interrupt

'*****

Include "sound.bas"

start:

ySAMPLE_RATE = 8
dTIMERPRELOAD = 100000/(ySAMPLE_RATE)    '10000 * 100
dTIMERPRELOAD = dTIMERPRELOAD + 55        'Round up
dTIMERPRELOAD = dTIMERPRELOAD / 100       '10000 / 100
dTIMERPRELOAD = 65536 - dTIMERPRELOAD + 11

T1CON.0 = 0                              '0 = Timer 1 off, should be off at
stop play and file config
wTIMER1 = dTIMERPRELOAD
T1CON.0 = 1                              '1 = Timer 1 on
iINTERRUPT_SET = 0

dSTATE_PREV_SAMPLE_DE = 0
ySTATE_PREV_INDEX_DE = 0

'*****

GoSub Read_Init

Main:

    GoSub Read_HelloWorld
    DelayMS 2000

GoTo Main

'*****

Read_Init:

    For wSOUNDADDRESS = 1 To Init_sample
        yBYTE_2_4BITS_DE = LRead8 Init[wSOUNDADDRESS]
        yCODE_DE = yBYTE_2_4BITS_DE >> 4
        GoSub subDECODE_ADPCM
        yCODE_DE = yBYTE_2_4BITS_DE << 4        '2nD: Decode lower 4 bits to byte
        yCODE_DE = yCODE_DE >> 4
        GoSub subDECODE_ADPCM
    Next wSOUNDADDRESS

Return

'*****

Read_HelloWorld:

    For wSOUNDADDRESS = 1 To HelloWorld_sample
        yBYTE_2_4BITS_DE = LRead8 HelloWorld[wSOUNDADDRESS]
        yCODE_DE = yBYTE_2_4BITS_DE >> 4
        GoSub subDECODE_ADPCM
        yCODE_DE = yBYTE_2_4BITS_DE << 4        '2nD: Decode lower 4 bits to byte
        yCODE_DE = yCODE_DE >> 4
        GoSub subDECODE_ADPCM
    Next wSOUNDADDRESS

```

Return

```
'*****
```

```
INDEX_TABLE:                                     'For subdDECODE_ADPCM
```

```
  LData As DWord -1, -1, -1, -1, 2, 4, 6, 8, _  
                  -1, -1, -1, -1, 2, 4, 6, 8
```

```
'*****
```

```
STEPSIZE_TABLE:
```

```
  LData As Word 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, _  
                19, 21, 23, 25, 28, 31, 34, 37, 41, 45, _  
                50, 55, 60, 66, 73, 80, 88, 97, 107, 118, _  
                130, 143, 157, 173, 190, 209, 230, 253, 279, 307, _  
                337, 371, 408, 449, 494, 544, 598, 658, 724, 796, _  
                876, 963, 1060, 1166, 1282, 1411, 1552, 1707, 1878, 2066, _  
                2272, 2499, 2749, 3024, 3327, 3660, 4026, 4428, 4871, 5358, _  
                5894, 6484, 7132, 7845, 8630, 9493, 10442, 11487, 12635, 13899, _  
                15289, 16818, 18500, 20350, 22385, 24623, 27086, 29794, 32767
```

```
'*****
```

```
subDECODE_ADPCM:
```

```
  dPRED_SAMPLE_DE = dSTATE_PREV_SAMPLE_DE  
  dINDEX_DE = ySTATE_PREV_INDEX_DE  
  wSTEP_DE = LRead16 STEPSIZE_TABLE[dINDEX_DE]  
  wDIFFQ_DE = wSTEP_DE >> 3
```

```
  If (yCODE_DE & 4) = 4 Then  
    wDIFFQ_DE = wDIFFQ_DE + wSTEP_DE  
  EndIf
```

```
  If (yCODE_DE & 2) = 2 Then  
    wDIFFQ_DE = wDIFFQ_DE + (wSTEP_DE >> 1)  
  EndIf
```

```
  If (yCODE_DE & 1) = 1 Then  
    wDIFFQ_DE = wDIFFQ_DE + (wSTEP_DE >> 2)  
  EndIf
```

```
  If (yCODE_DE & 8) = 8 Then  
    dPRED_SAMPLE_DE = dPRED_SAMPLE_DE - wDIFFQ_DE  
  Else  
    dPRED_SAMPLE_DE = dPRED_SAMPLE_DE + wDIFFQ_DE  
  EndIf
```

```
  If dPRED_SAMPLE_DE > 65535 Then  
    dPRED_SAMPLE_DE = 65535  
  ElseIf dPRED_SAMPLE_DE < 0 Then  
    dPRED_SAMPLE_DE = 0  
  EndIf
```

```
  dINDEX_DE = dINDEX_DE + LRead32 INDEX_TABLE[yCODE_DE]
```

```
  If dINDEX_DE < 0 Then  
    dINDEX_DE = 0  
  EndIf
```

```
  If dINDEX_DE > 88 Then  
    dINDEX_DE = 88  
  EndIf
```

```
dSTATE_PREV_SAMPLE_DE = dPRED_SAMPLE_DE  
ySTATE_PREV_INDEX_DE = dINDEX_DE
```

```
While iINTERRUPT_SET == 0  
Wend
```

```
iINTERRUPT_SET = 0
```

```
PORTB = dPRED_SAMPLE_DE>>8
```

```
Return
```

```
' *****
```