```
1
2  from machine import Pin, I2C
3
4  class Radio:
5      def __init__(self, freq, vol, mute):
6          self.Volume = 2
7          self.Frequency = 88
8          self.Mute = False
9
10         self.SetVolume(vol)
11         self.SetFrequency(freq)
12         self.SetMute(mute)
13
14         self.i2c_sda = Pin(26)
15         self.i2c_scl = Pin(27)
16         self.i2c_device = 1
17         self.i2c_device_address = 0x10
18         self.Settings = bytearray(8)
19         self.radio_i2c = I2C(self.i2c_device, scl=self.i2c_scl, sda=self.i2c_sda, freq=200000)
20         self.ProgramRadio()
21
22     def SetVolume(self, v):
23         try:
24             v = int(v)
25             if 0 <= v < 16:
26                 self.Volume = v
27                 return True
28         except:
29             pass
30         return False
31
32     def SetFrequency(self, f):
33         try:
34             f = float(f)
35             if 88.0 <= f <= 108.0:
36                 self.Frequency = f
37                 return True
38         except:
39             pass
40         return False
41
42     def SetMute(self, m):
43         try:
44             self.Mute = bool(int(m))
45             return True
46         except:
47             return False
48
49     def ComputeChannelSetting(self, f):
50         f = int(f * 10) - 870
51         return bytearray([(f >> 2) & 0xFF, ((f & 0x03) << 6) & 0xC0])
52
53     def UpdateSettings(self):
54         self.Settings[0] = 0x80 if self.Mute else 0xC0
55         self.Settings[1] = 0x09 | 0x04
56         self.Settings[2:4] = self.ComputeChannelSetting(self.Frequency)
57         self.Settings[3] |= 0x10
58         self.Settings[4] = 0x04
59         self.Settings[5] = 0x00
60         self.Settings[6] = 0x84
61         self.Settings[7] = 0x80 + self.Volume
62
63     def ProgramRadio(self):
64         self.UpdateSettings()
```

```
65          self.radio_i2c.writeto(self.i2c_device_address, self.Settings)
66
```