How to CODE Music for the Arduino IDE

# Know your notes

## Basic Note, first part

The 7 natural notes of the musical octave will be represented in code as follows:

CC, DD, EE, FF, GG, AA, BB

The complete 12 notes of the musical octave, including accidentals (sharps and flats) are represented as follows:

CC, **Cs or Db,** DD, **Ds or Eb**, EE, FF, **Fs or Gb,** GG, **Gs or Ab,** AA, **As or Bb,** BB

where 's' or 'b' represents sharp or flat, respectively. Each accidental can be represented as either a sharp or a flat, to make it easier to translate different key signatures.

## Add the Octave

The note value isn't complete without indicating the octave. In this code, we have octaves 0 through 8.

The octave number is added to the end of the note desired to represent the note completely:

CC4, DD4, EE4, FF4, GG4, AA4, BB4

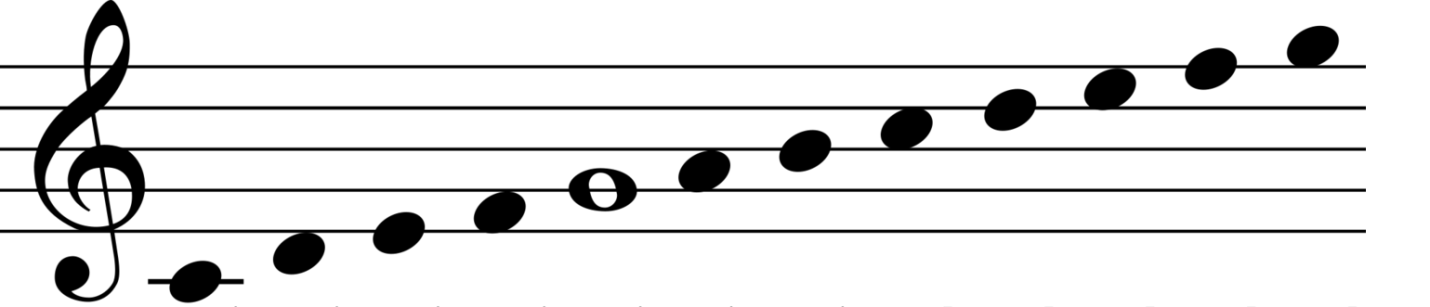
Where the numerical digit represents the octave number, 0 - 8.

In defining the scale here it is assumed:

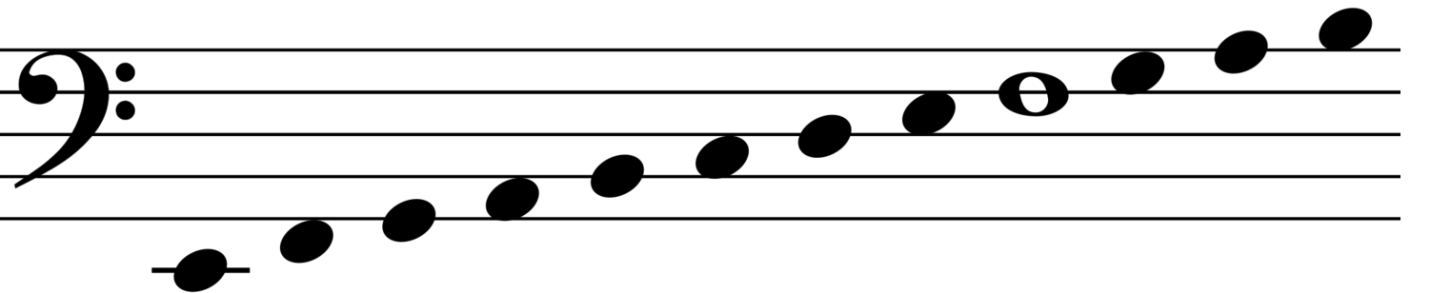
* Middle C here is CC4
* AA4 being 49th key of an ideal piano, having a frequency of 440 Hz

# Note Value

Given the above nomenclature, the value of each of the notes shown below can be also represented in code as:



CC4,DD4,EE4,FF4,GG4,AA4,BB4,CC5,DD5,EE5,FF5,GG5



EE2,FF2,GG2,AA2,BB2,CC3,DD3,EE3,FF3,GG3,AA3,BB3

# Note Length

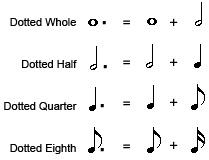
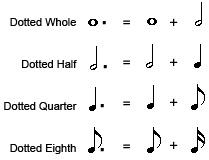
Here is how common note durations are represented in code:



W H Q E S\_S

(The software has 'S' and 'SS' often reserved, so to prevent problems we use 'S\_S' to represent a sixteenth note).

Dotted notes simply mean add half the note duration to that note. In code this is done simply with the '+' operator:



W+H

H+Q

Q+E

E+S\_S

# Making Notes into a Song

Finally, in the code, making a song involves putting all the above together into a form that the software can understand. Specifically, a two-dimensional array of integers.

// Oh Little Town of Bethlehem - HYMN 361 (LSB)

int song0[100][2] = { {AA4,Q},{AA4,Q},{AA4,Q},{Gs4,Q},{AA4,Q},{CC5,Q},{Bb4,Q},{DD4,Q},{GG4,Q},{FF4,Q},{EE4,E},{FF4,E},{GG4,Q},{CC4,Q},{AA4,H+Q},

{AA4,Q},{AA4,Q},{AA4,Q},{DD5,Q},{CC5,Q},{CC5,Q},{Bb4,Q},{DD4,Q},{GG4,Q},{FF4,Q},{EE4,E},{FF4,E},{AA4,Q},{GG4,Q},{FF4,H+Q},

{AA4,Q},{AA4,Q},{AA4,Q},{GG4,Q},{FF4,Q},{EE4,H},{EE4,Q},{EE4,Q},{DD4,Q},{EE4,Q},{FF4,Q},{GG4,Q},{AA4,H+Q},

{AA4,Q},{AA4,Q},{AA4,Q},{Gs4,Q},{AA4,Q},{CC5,Q},{Bb4,Q},{DD4,Q},{DD5,Q},{CC5,Q},{FF4,Q},{AA4,Q+E},{GG4,E},{FF4,H+Q}

};

Basically the above syntax can be used to code any basic song you like.

# Final touches to the Song

Finally, to get the song into the code...