```
// Arduino Buildup
// Arduino Commander

// Board Description
// 4,5,6 tied to gnd through 470 ohm resistor
// 7,8,9 input switches, N/O, through resistor to grnd
// 12 piezo speaker through resistor to grnd
// 2 input switch, N/O, through resistor to grnd

// Defines
#define      MAXCMD      50// Max Command Length
#define     MAXTIK     10000// Note Duration
#define       PTONE      12// Port for Piezo

// Includes
#include <string.h>
#include <avr/sleep.h>

// Globals
char                     b;// Input buffer
char cmd[MAXCMD+1] = {""};  // Command buffer

// Input buttons
int         btn[3]={1,1,1};// Button Status Register

// Notes
char    *note     =     0;// Pointer to next note
long    ntick      =      0;// How long has not played
unsigned char notes[] = {
'B','0',' ',31,0,
'C','1',' ',33,0,
'C','S','1',35,0,
'D','1',' ',37,0,
'D','S','1',39,0,
'E','1',' ',41,0,
'F','1',' ',44,0,
'F','S','1',46,0,
'G','1',' ',49,0,
'G','S','1',52,0,
'A','1',' ',55,0,
'A','S','1',58,0,
'B','1',' ',62,0,
'C','2',' ',65,0,
'C','S','2',69,0,
'D','2',' ',73,0,
'D','S','2',78,0,
'E','2',' ',82,0,
'F','2',' ',87,0,
'F','S','2',93,0,
'G','2',' ',98,0,
'G','S','2',104,0,
'A','2',' ',110,0,
'A','S','2',117,0,
'B','2',' ',123,0,
'C','3',' ',131,0,
'C','S','3',139,0,
'D','3',' ',147,0,
'D','S','3',156,0,
'E','3',' ',165,0,
'F','3',' ',175,0,
'F','S','3',185,0,
'G','3',' ',196,0,
'G','S','3',208,0,
'A','3',' ',220,0,
'A','S','3',233,0,
'B','3',' ',247,0,
'C','4',' ',6,1,   //262
'C','S','4',15,1,  //277
'D','4',' ',38,1,  //294
'D','S','4',55,1,  //311
'E','4',' ',74,1,  //330
'F','4',' ',93,1,  //349
'F','S','4',114,1, //370
'G','4',' ',136,1, //392
'G','S','4',159,1, //415
'A','4',' ',184,1, //440
```

```
'A','S','4',210,1, //466
'B','4',' ',238,1, //494
'C','5',' ',11,2,  //523
'C','S','5',42,2,  //554
'D','5',' ',75,2, //587
'D','S','5',110,2, //622
'E','5',' ',147,2, //659
'F','5',' ',186,2, //698
'F','S','5',228,2, //740
'G','5',' ',16,3,  //784
'G','S','5',63,3,  //831
'A','5',' ',112,3, //880
'A','S','5',164,3, //932
'B','5',' ',220,3, //988
'C','6',' ',23,4,  //1047
'C','S','6',85,4,  //1109
'D','6',' ',151,4, //1175
'D','S','6',221,4, //1245
'E','6',' ',39,5,  //1319
'F','6',' ',117,5, //1397
'F','S','6',200,5, //1480
'G','6',' ',32,6,  //1568
'G','S','6',125,6, //1661
'A','6',' ',224,6, //1760
'A','S','6',73,7,  //1865
'B','6',' ',184,7, //1976
'C','7',' ',45,8,  //2093
'C','S','7',169,8, //2217
'D','7',' ',45,9,  //2349
'D','S','7',185,9, //2489
'E','7',' ',77,10, //2637
'F','7',' ',234,10, //2794
'F','S','7',144,11, //2960
'G','7',' ',64,12,  //3136
'G','S','7',250,12, //3322
'A','7',' ',192,13, //3520
'A','S','7',145,14, //3729
'B','7',' ',111,15, //3951
```

```
'C','8',' ',90,16,  //4186
'C','S','8',83,17,  //4435
'D','8',' ',91,18,  //4699
'D','S','8',114,19, //4978
0,0,0,0,0 };

// Interrupt Flag
volatile int iflag = 0;

void setup() {
  // Initialize Output Pins
  for (int i=4;i<=6;i++) {
    pinMode(i,OUTPUT);
    digitalWrite(i,LOW);
    pinMode(i+3,INPUT_PULLUP);
  }
  //:Attach Interrupt
  pinMode(2,INPUT_PULLUP);
  attachInterrupt(0,  Interrupt,FALLING);

  //: Start up Serial
  Serial.begin(9600);
  Serial.println("Arduino Commander v1.2");
  Ready();
}

void loop() {

  // Interrupt
  if (iflag) {
    iflag = 0;
    Serial.println("");
    Serial.println("Caught Interrupt");
  }

  // Play next note?
  if  (note)
    if (++ntick > MAXTIK)
```

```
      PlayNote();

  // Read Serial Port, build command
  if (Serial.available()) {
     b =Serial.read();
    //: If /, then process command
    if (b=='/') {
      ProcessCommand();
      return;
    }
    int i = strlen(cmd);
    if (i < MAXCMD) {
      cmd[i] = b;
      cmd[i+1]= 0;
      return;
    }
  }

  // Input Polling
  int dirty = 0;
  for (int i=7;i<=9;i++) {
    b=digitalRead(i);
    if (b!=btn[i-7]) {
      btn[i-7]=b;
      dirty = 1;
    }
  }
  if (dirty) {
    Serial.println("");
    Serial.print("Input Value = ");
    Serial.println(btn[0] + btn[1]*2 + btn[2]*4);
  }


}

// Command Processor
void ProcessCommand(){
```

```
Serial.println();
Serial.println(cmd);

// HELP
if (strcasestr(cmd,"HELP")) {
  Syntax();
  return;
}

// PLAY
if (strcasestr(cmd,"PLAY")) {
  note = cmd+2;
  return;
}

// SLEEP
if (strcasestr(cmd,"SLEEP")) {
   set_sleep_mode(SLEEP_MODE_PWR_DOWN);
 sleep_mode();  o
   sleep_disable();
 delay(1000);
 Serial.flush();
 Serial.println("Ok! Ok! I'm Awake! I'm Awake!");
 Ready();
 return;
}

// Command
b = cmd[0] -'0';
if (b>=4 && b<=6) {
  // valid Port Number
  Serial.println("");
  Serial.print("Port ");
  Serial.print(b,DEC);
  Serial.print(" set ");
  if (strcasestr(cmd,"ON")) {
    digitalWrite(b,HIGH);
```

```
      Serial.println("ON");
    }
    else {
      digitalWrite(b,LOW);
      Serial.println("OFF");
    }
    Ready();
    return;
  }

  // Syntax Error
  Serial.println("Syntax Error!");
  Syntax();
}

void Syntax() {
  Serial.println("");
  Serial.println("SYNTAX: COMMAND/");
  Serial.println("PORT(4-6) ACTION(ON/OFF) ie: 4 ON/");
  Serial.println("PLAY NOTE [NOTE [...]] ie: PLAY C1 C2 /");
  Serial.println("SLEEP/");
  Ready();
}

// Ready Prompt
void Ready() {
  Serial.println("");
  Serial.print("Ready >");
  ClearCMD();
}

void ClearCMD(){
 for (int i=0;i<MAXCMD;i++) cmd[i]=0;
}

void PlayNote() {
 char   n[4];// holds next note text
 n[3]=0;
 char c[4];
 c[3]=0;
 unsigned char     *p;// pointer into note array
 int       t;// tone;
 note+=3;
 if (note-cmd >= strlen(cmd)) {
   note = 0;
   noTone(PTONE);
   Ready();
   return;
 }
 // note to play
 strncpy(n,note,3);
 //Serial.print("Searching for: ");
 //Serial.println(n);
 // itterate through notes
 for (p = notes; *p; p+=5) {
   strncpy(c,(char *)p,3);
   //Serial.print("Saw: ");
   //Serial.println(c);
   if (strcasestr(n,c)) {
     //Serial.println(*(p+3),DEC);
     //Serial.println(*(p+4),DEC);
     noTone(PTONE);
     delay(100);
     unsigned int itone = *(p+3) + (*(p+4) * 256);
     //Serial.println(itone,DEC);
     tone(PTONE,itone);
        ntick  =  0;
     return;
   }
 }
}

void Interrupt() {
  iflag = 1;
}
```