

## ACB code programming description

The software used to carry out the programming is ARDUINO IDE. It is a free software that can be downloaded from the official Arduino webpage: <https://www.arduino.cc/en/software> (Illustration 1). Once on the webpage, click on the download link depending on the operating system that is being used for its installation. When the Arduino IDE is installed it can execute it, and a new Sketch will open (Illustration 2) in which we can start programming.

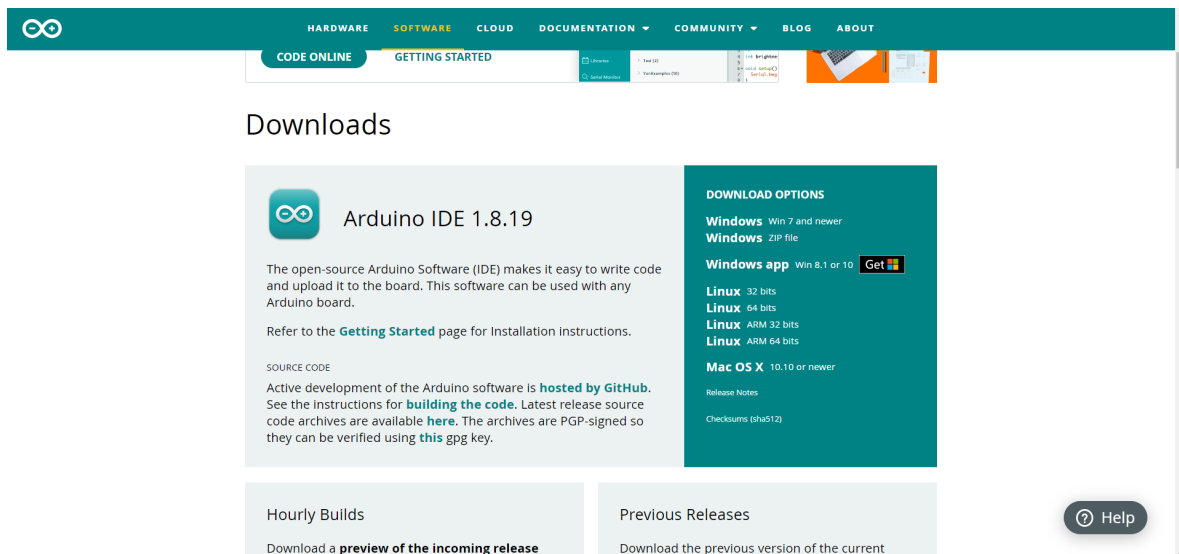


Ilustración 1



Ilustration 2

First, it is necessary to install libraries, it is done through the library manager of the Arduino IDE, clicking on Sketch\Include library\Manage Libraries (Illustration 3). A library search engine will be displayed on the screen (Illustration 4), write in the search engine “RTCLib” to search for the necessary library to use the RTC3231 and install (Illustration 5). Then, delete “RTCLib” from the browser and type “DFROBOTdfplayermini” to search for the necessary library to use the player and install it (Illustration 6).

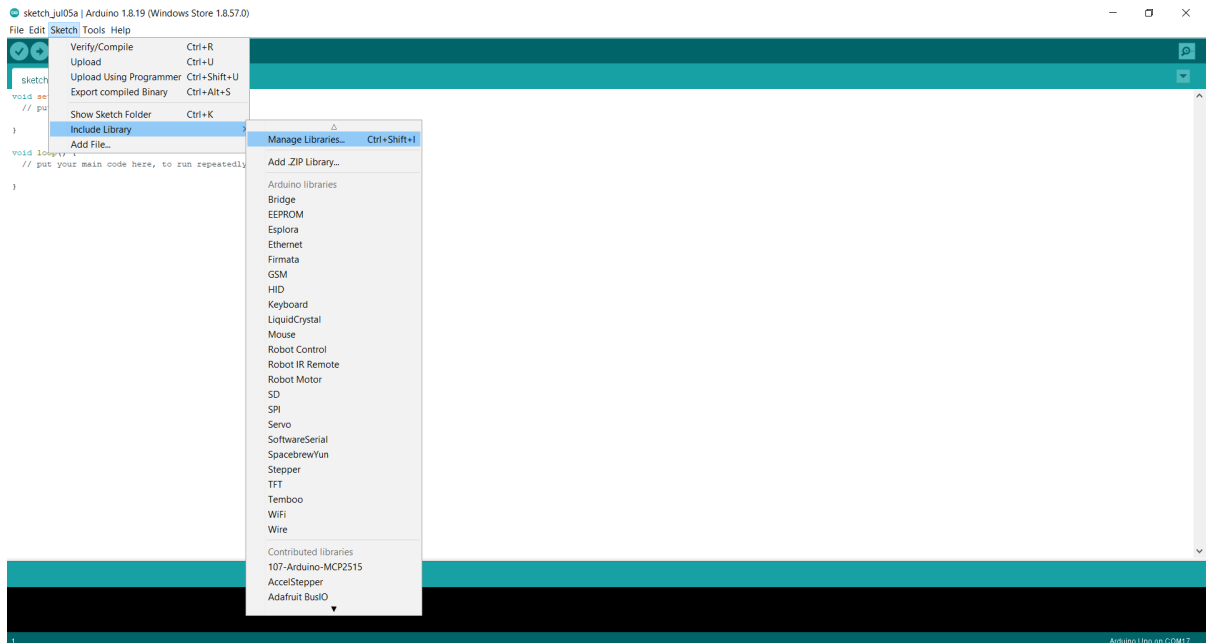


Ilustración 3

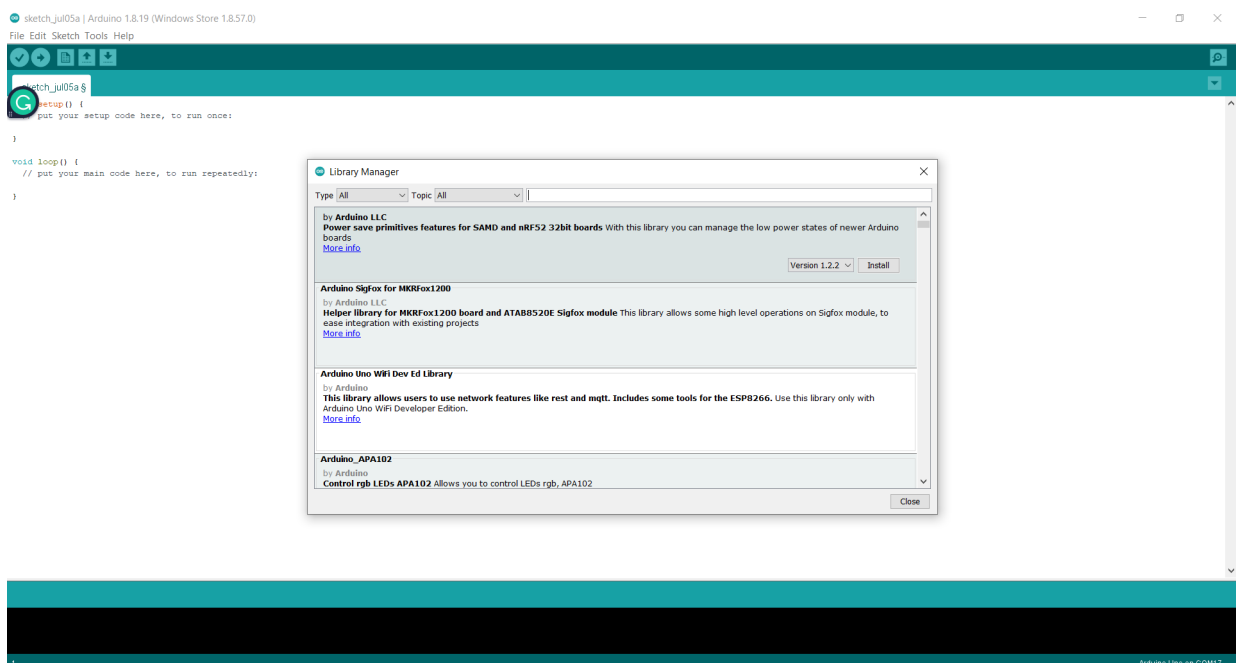


Ilustración 4

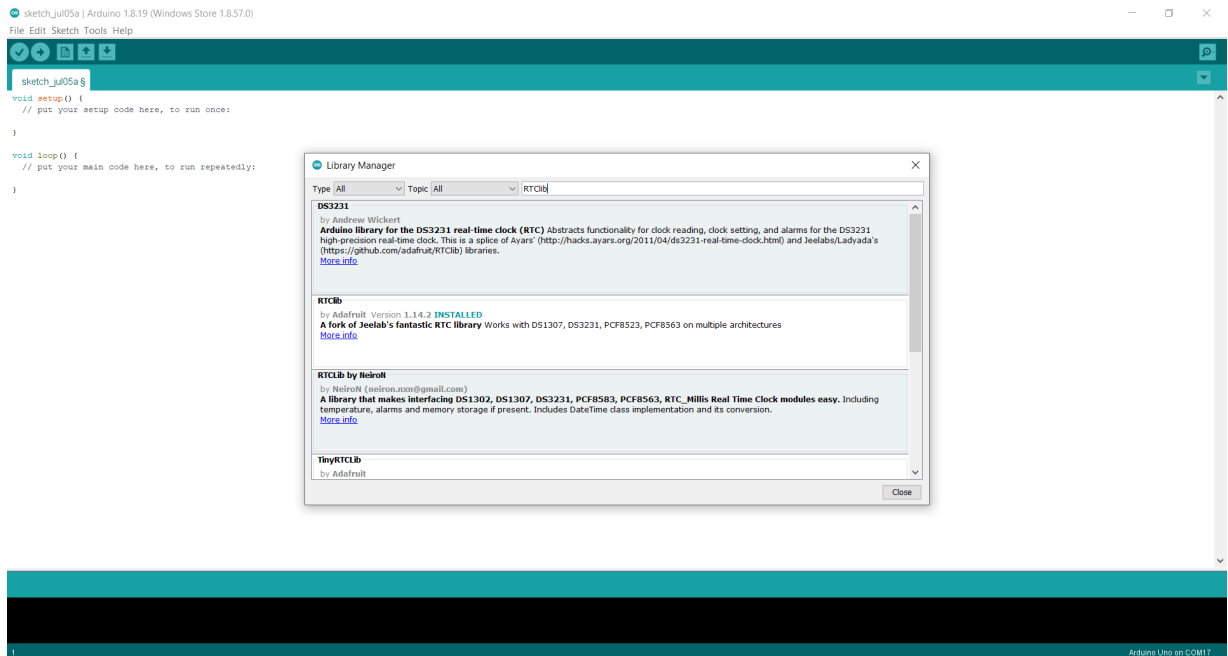


Ilustración 5

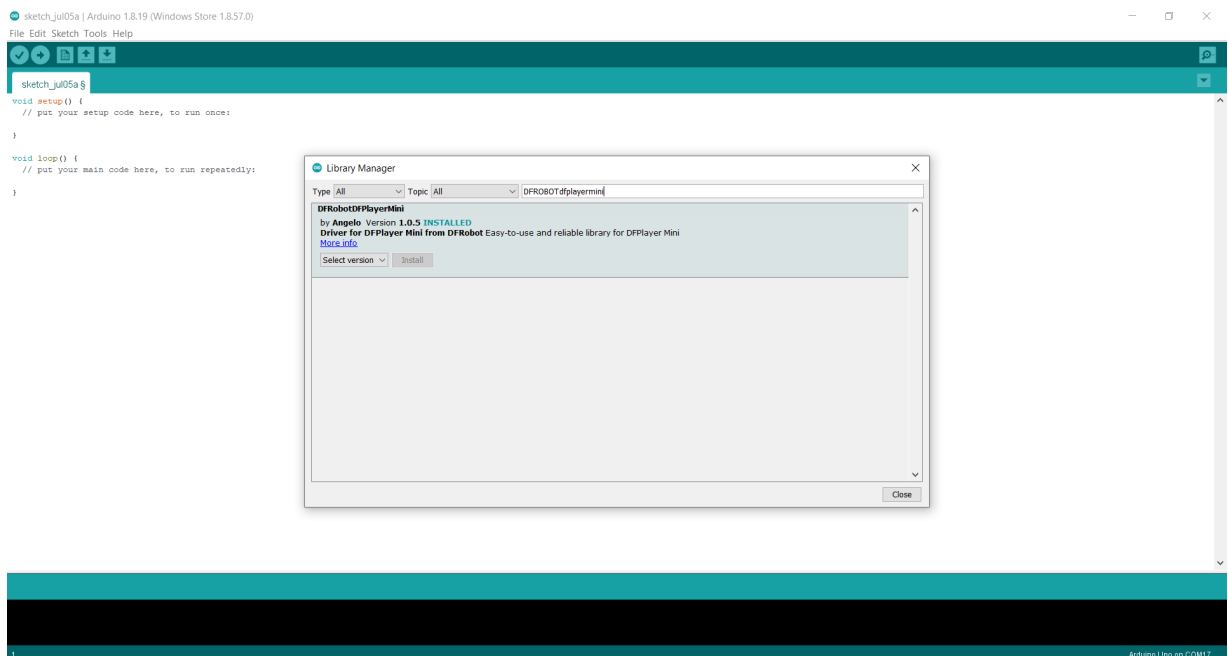


Ilustración 6

The first lines of code are the necessary libraries for the development of the project.

```
#include "SoftwareSerial.h"
```

```
#include "DFRobotDFPlayerMini.h"
```

```
#include "RTCLib.h"
```

Next, write the necessary lines of code for the declaration of the objects used in the code.

```
SoftwareSerial mySoftwareSerial(3, 2); // RX, TX
```

```
DFRobotDFPlayerMini myDFPlayer;
```

```
RTC_DS3231 rtc;
```

The following is the prototype of the necessary function to print details and to know some errors for which the MP3 would not work.

```
void printDetail(uint8_t type, int value);
```

Global variables that we are going to use in the project, the last variable for the use of millis, the playing variable is used to know if we are playing audio or not, and the variables m and h are used to save the current hour and minutes.

```
unsigned long last = 0;
```

```
int h, m;
```

```
boolean reproduciendo = false;
```

Inside the void setup, we have the necessary lines to initialize, the serial monitor, the RTC, and the dfplayermini. Information in grey or after // is a comment not a functional code.

```
mySoftwareSerial.begin(9600);
```

```
Serial.begin(9600);
```

```
if (!rtc.begin()) {
```

```
    Serial.println("Couldn't find RTC");
```

```
    Serial.flush();
```

```
    abort();
```

```
}
```

```
//rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));-----  
-----****
```

```
if (!myDFPlayer.begin(mySoftwareSerial)) { //Use softwareSerial to communicate with  
mp3.
```

```
    Serial.println(F("Unable to begin:"));
```

```
    Serial.println(F("1.Please recheck the connection!"));
```

```
    Serial.println(F("2.Please insert the SD card!"));
```

```
    while (true) {
```

```
        delay(0); // Code to compatible with ESP8266 watch dog.
```

```
    }
```

```
}
```

```
Serial.println(F("DFPlayer Mini online.));
```

```
myDFPlayer.volume(30); //Set volume value. From 0 to 30
```

```
//myDFPlayer.play(1); //Play the first mp3
```

In the void loop, we have the programming logic that will be executed cyclically. We will make use of the `millis()` function, this function keeps a count since the microcontroller is turned on in milliseconds. For example, if the microcontroller had been powered up for 5 seconds, the `millis()` function would return a value of 5000. In our particular case, we use the `millis()` function to execute the lines of code necessary to update the variables h and m every second and also compare and execute the execution of some audio if this condition is met.

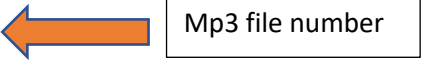
```
if (millis() - last >= 1000) {  
    last = millis();  
    DateTime now = rtc.now();  
    h = now.hour();  
    m = now.minute();  
    Serial.print(now.year(), DEC);  
    Serial.print('/');  
    Serial.print(now.month(), DEC);  
    Serial.print('/');  
    Serial.print(now.day(), DEC);  
    Serial.print("  ");  
    Serial.print(h);  
    Serial.print(':');  
    Serial.print(m);  
    Serial.print(':');  
    Serial.print(now.second(), DEC);  
    Serial.println();  
    //-----  
    if (h == 5 && m == 30 && reproduciendo == false) {  
        reproduciendo = true;  
        myDFPlayer.play(1);  
    }  
    if (h == 5 && m == 31 && reproduciendo == true) {  
        reproduciendo = false;  
    }  
}
```

The `if` will execute the lines of code within its braces only in the case that the variable `h` is equal to the current one and the variable `m` is equal to the current minutes and no audio is currently playing.

To program the time on RTC you need to use no line `(//)` to set the time from computer. The lines are then placed `(//)` to set up the code in Arduino and upload it (remember the ACB including: Arduino, RTC and DFplayer mini must be connected to the computer to be able to be programmed).

The sketch presented can be changed and adapted according to the research necessities. Times and number of playbacks must be in relation to the type of cue and target organism. Mp3 files should be named using numbers and not words. In the present sketch just one mp3 file was used, then we named as “1” and we call it in the code as that:

```
//-----  
if (h == 5 && m == 30 && reproduciendo == false) {  
  reproduciendo = true;  
  myDFPlayer.play(1);  
}  
if (h == 5 && m == 31 && reproduciendo == true) {  
  reproduciendo = false;  
}  
//-----
```



The code was made for using only time to program the system, but you can also add specific days using the following instructions in green within the original code:

```
if (millis() - last >= 1000) {  
  last = millis();  
  DateTime now = rtc.now();  
  h = now.hour();  
  m = now.minute();  
  día =now.day();  
  mes =now.month();  
  diaSemana= now.dayOfTheWeek();  
  Serial.print(now.year(), DEC);
```

```

Serial.print('/');
Serial.print(now.month(), DEC);
Serial.print('/');
Serial.print(now.day(), DEC);
Serial.print("  ");
Serial.print(h);
Serial.print(':');
Serial.print(m);
Serial.print(':');
Serial.print(now.second(), DEC);
Serial.println();

//-----
if (dia == 14 && mes == 1 && h == 5 && m == 30 && reproduciendo == false) {
    reproduciendo = true;
    myDFPlayer.play(1);
}
if (h == 5 && m == 31 && reproduciendo == true) {
    reproduciendo = false;
}

```

Using the following code you can use also days of a week:

0--> Sunday, 1--> Monday, 2-->Tuesday, 3-->Wednesday, 4--> Thursday, 5--> Friday, 6--> Saturday.

```

//-----
if (diaSemana== 3 && h== 5 && m==30 && reproduciendo == false) {
    reproduciendo = true;
    myDFPlayer.play(1);
}
if (diaSemana== 3 && h == 5 && m == 31 && reproduciendo == true) {
    reproduciendo = false;
}

```