**2190151 Computer Programming Lab**

Lab 4: Speaker, Digital clock and Stopwatch

Objectives:

1.Write a program to play a melody on M5Stack built-in Speaker.

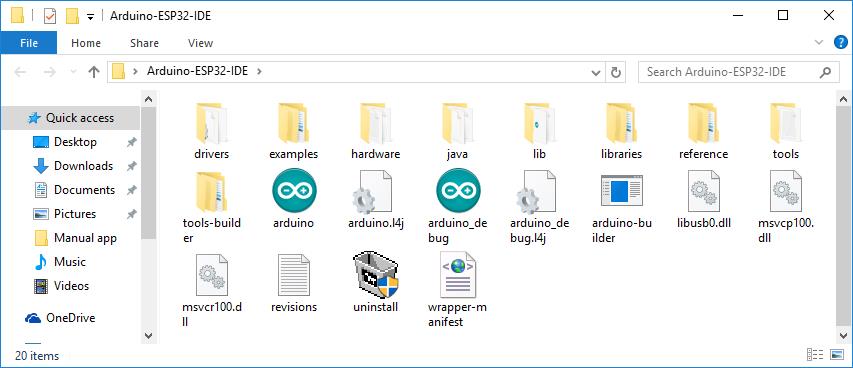
2.Write simple program to show the time on TFT LCD screen.

3.Design a program that use TFT and button function.

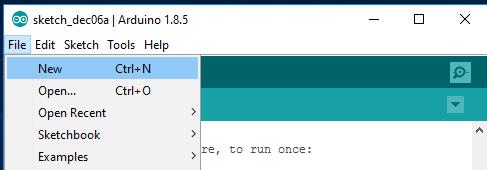
4. Create a simple watch function using selection and loop.

**Task 1: Speaker, play Jingle Bells**

**Step 1. Go to the folder Arduino-ESP32-IDE and open the app arduino.exe (Fig. 1);**

  
Figure 1. The archive with the Arduino IDE extracted to the folder

**Step 2. On the File menu, select New (Fig. 1.1);**

  
Figure 1.1. Create a new sketch

**Step 3. Select Include Library, M5Stack in the Sketch menu (Fig. 1.2);**

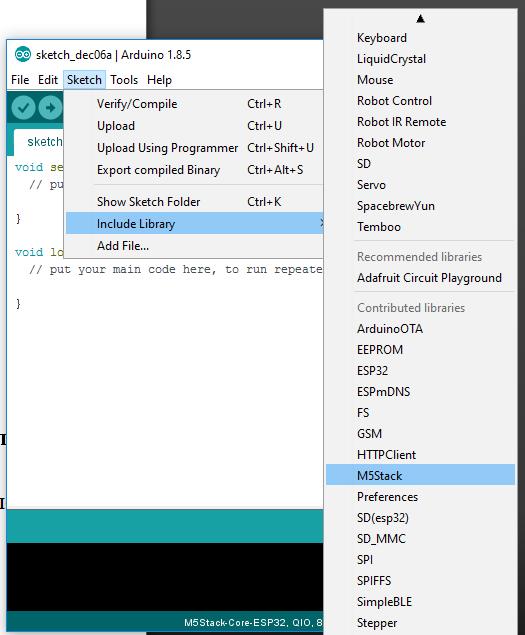
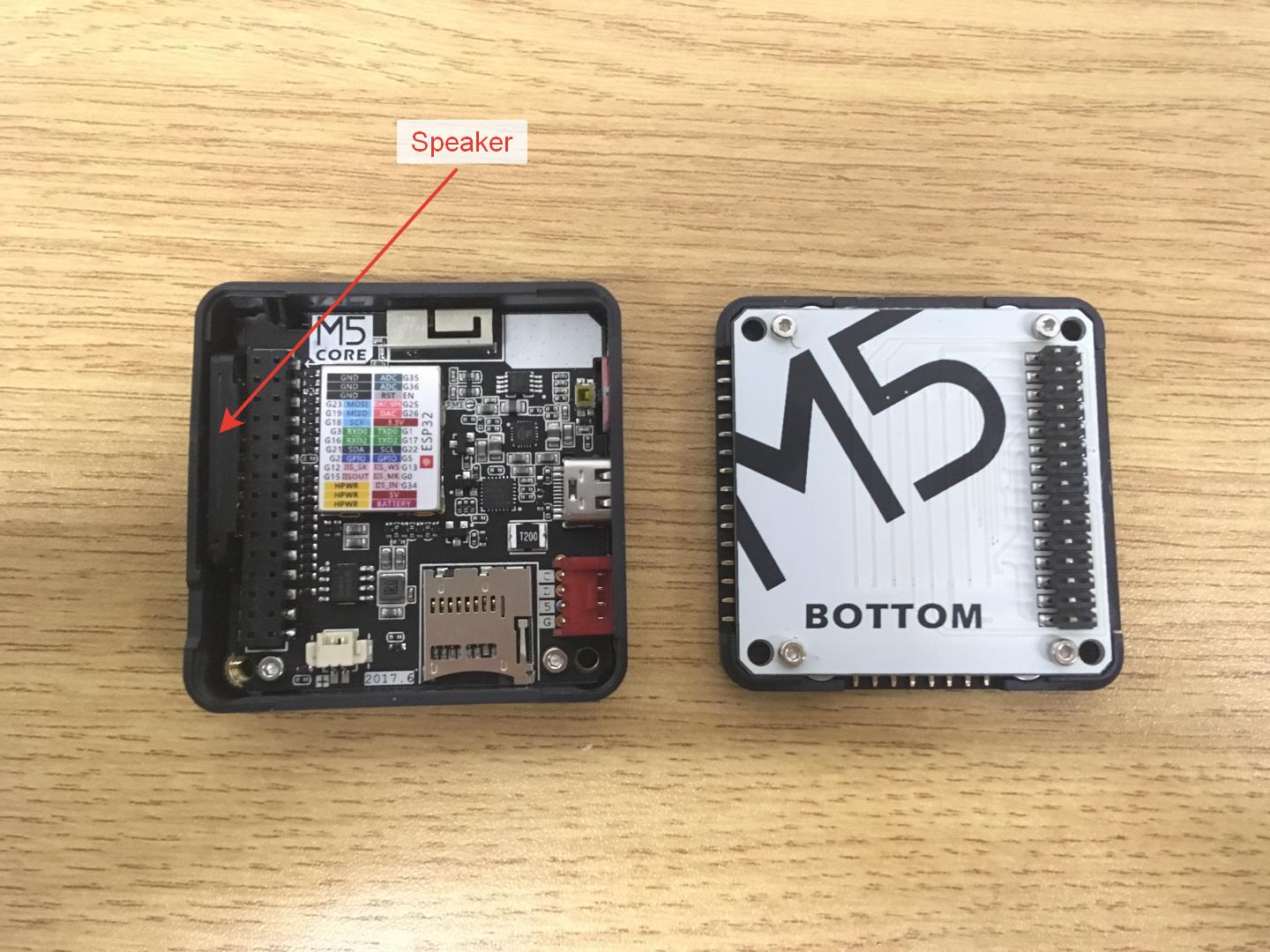
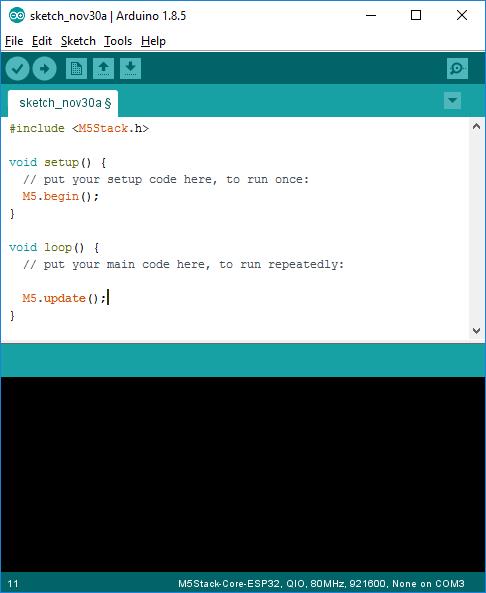
  
Figure 1.2. Connection library M5Stack

  
Figure 1.3. The holes for the sound from the built-in speaker are located on the right side of the device

  
Figure 1.4. Speaker is located on the left of the bus 2 to 15

**Step 4. Add to the beginning of system functions void setup() function call M5.begin (); because it is necessary to prepare the device to work. Add to the end system functions void loop() function call M5.update (); is required for correct operation of the device with built-in speaker. This step should be performed always when writing a new sketch (Fig. 2).**

  
Figure 2. An example of the correct preparation of the sketch for work with built-in speaker

**Step 5. To work with the speaker (Fig. 1.3, 1.4) use the following functions and constants notes in table 1 and 2 respectively:**

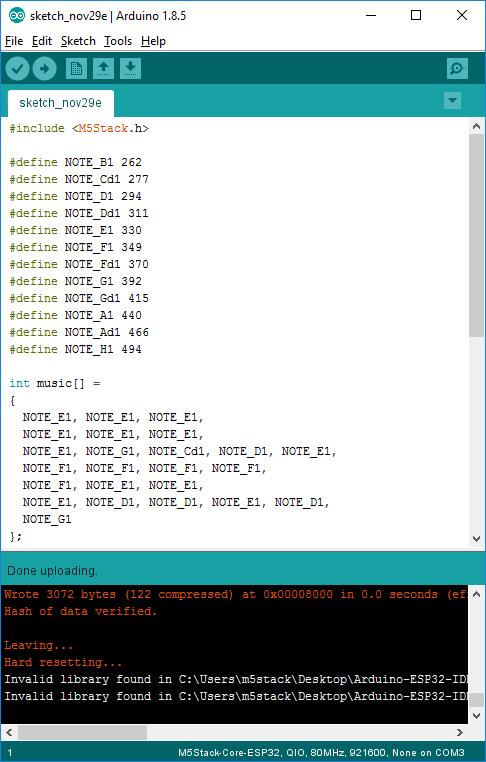
List 1. Functions for working with built-in speaker

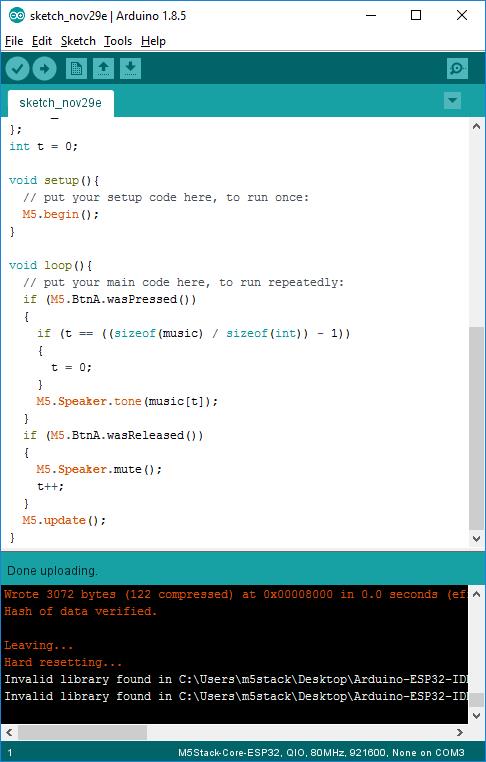
* **M5.Speaker.tone(uint32\_t freq);** to Generate a beep at a specified frequency (Hz)
* **M5.Speaker.mute();** Interrupt generation

List 2. Constant notes (frequencies rounded to whole numbers)

* **#define NOTE\_D0 -1 silence**  
  THE FIRST OCTAVE
* **#define NOTE\_B1 262 To (B)**
* **#define NOTE\_Cd1 277 C sharp (C#)**
* **#define NOTE\_D1 294 Re (D)**
* **#define NOTE\_Dd1 311 in D sharp (D#)**
* **#define NOTE\_E1 330 Mi (E)**
* **#define NOTE\_F1 349 FA (F)**
* **#define NOTE\_Fd1 370 f-sharp (F#)**
* **#define NOTE\_G1 392 Sol (G)**
* **#define NOTE\_Gd1 415 g sharp (G#)**
* **#define NOTE\_A1 440 La (A)**
* **#define NOTE\_Ad1 466 La-a sharp (A#)**
* **#define NOTE\_H1 494 si (H)**

**Step 6. Write a music box (Fig. 2.1, 2.2);**

  
Figure 2.1. Code music box. Page first

  
Figure 2.2. Code music box. Page two

**Step 7. Click the Upload button (Fig. 3) in order to flash the device;**

  
Figure 3. Download the firmware to the device

**Step 8. When the device firmware is completed, press the A button and enjoy the melody :)**

**Task 2 Create a program that allow M5 stack to be a digital clock using the following code**

****

|  |
| --- |
|  |
| /\* |
|  | An example digital clock using a TFT LCD screen to show the time. |
|  | Demonstrates use of the font printing routines. (Time updates but date does not.) |
|  |  |
|  | It uses the time of compile/upload to set the time |
|  | For a more accurate clock, it would be better to use the RTClib library. |
|  | But this is just a demo... |
|  |  |
|  | Make sure all the display driver and pin connections are correct by |
|  | editing the User\_Setup.h file in the TFT\_eSPI library folder. |
|  |  |
|  | ######################################################################### |
|  | ###### DON'T FORGET TO UPDATE THE User\_Setup.h FILE IN THE LIBRARY ###### |
|  | ######################################################################### |
|  |  |
|  | Based on clock sketch by Gilchrist 6/2/2014 1.0 |
|  |  |
|  | A few colour codes: |
|  |  |
|  | code color |
|  | 0x0000 Black |
|  | 0xFFFF White |
|  | 0xBDF7 Light Gray |
|  | 0x7BEF Dark Gray |
|  | 0xF800 Red |
|  | 0xFFE0 Yellow |
|  | 0xFBE0 Orange |
|  | 0x79E0 Brown |
|  | 0x7E0 Green |
|  | 0x7FF Cyan |
|  | 0x1F Blue |
|  | 0xF81F Pink |
|  |  |
|  | \*/ |
|  |  |
|  |  |
|  |  |
|  | M5.Lcd.fillScreen(TFT\_BLACK); |
|  |  |
|  | M5.Lcd.setTextSize(1); |
|  | M5.Lcd.setTextColor(TFT\_YELLOW, TFT\_BLACK); |
|  |  |
|  | targetTime = millis() + 1000; |
|  | } |
|  |  |
|  | void loop() { |
|  | if (targetTime < millis()) { |
|  | // Set next update for 1 second later |
|  | targetTime = millis() + 1000; |
|  |  |
|  | // Adjust the time values by adding 1 second |
|  | ss++; // Advance second |
|  | if (ss == 60) { // Check for roll-over |
|  | ss = 0; // Reset seconds to zero |
|  | omm = mm; // Save last minute time for display update |
|  | mm++; // Advance minute |
|  | if (mm > 59) { // Check for roll-over |
|  | mm = 0; |
|  | hh++; // Advance hour |
|  | if (hh > 23) { // Check for 24hr roll-over (could roll-over on 13) |
|  | hh = 0; // 0 for 24 hour clock, set to 1 for 12 hour clock |
|  | } |
|  | } |
|  | } |
|  |  |
|  |  |
|  | // Update digital time |
|  | int xpos = 0; |
|  | int ypos = 85; // Top left corner ot clock text, about half way down |
|  | int ysecs = ypos + 24; |
|  |  |
|  | if (omm != mm) { // Redraw hours and minutes time every minute |
|  | omm = mm; |
|  | // Draw hours and minutes |
|  | if (hh < 10) xpos += M5.Lcd.drawChar('0', xpos, ypos, 8); // Add hours leading zero for 24 hr clock |
|  | xpos += M5.Lcd.drawNumber(hh, xpos, ypos, 8); // Draw hours |
|  | xcolon = xpos; // Save colon coord for later to flash on/off later |
|  | xpos += M5.Lcd.drawChar(':', xpos, ypos - 8, 8); |
|  | if (mm < 10) xpos += M5.Lcd.drawChar('0', xpos, ypos, 8); // Add minutes leading zero |
|  | xpos += M5.Lcd.drawNumber(mm, xpos, ypos, 8); // Draw minutes |
|  | xsecs = xpos; // Sae seconds 'x' position for later display updates |
|  | } |
|  | if (oss != ss) { // Redraw seconds time every second |
|  | oss = ss; |
|  | xpos = xsecs; |
|  |  |
|  | if (ss % 2) { // Flash the colons on/off |
|  | M5.Lcd.setTextColor(0x39C4, TFT\_BLACK); // Set colour to grey to dim colon |
|  | M5.Lcd.drawChar(':', xcolon, ypos - 8, 8); // Hour:minute colon |
|  | xpos += M5.Lcd.drawChar(':', xsecs, ysecs, 6); // Seconds colon |
|  | M5.Lcd.setTextColor(TFT\_YELLOW, TFT\_BLACK); // Set colour back to yellow |
|  | } |
|  | else { |
|  | M5.Lcd.drawChar(':', xcolon, ypos - 8, 8); // Hour:minute colon |
|  | xpos += M5.Lcd.drawChar(':', xsecs, ysecs, 6); // Seconds colon |
|  | } |
|  |  |
|  | //Draw seconds |
|  | if (ss < 10) xpos += M5.Lcd.drawChar('0', xpos, ysecs, 6); // Add leading zero |
|  | M5.Lcd.drawNumber(ss, xpos, ysecs, 6); // Draw seconds |
|  | } |
|  | } |
|  | } |
|  |  |
|  |  |
|  | // Function to extract numbers from compile time string |
|  | static uint8\_t conv2d(const char\* p) { |
|  | uint8\_t v = 0; |
|  | if ('0' <= \*p && \*p <= '9') |
|  | v = \*p - '0'; |
|  | return 10 \* v + \*++p - '0'; |
|  | } |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
|  |
|  | |
| #include <M5Stack.h> | |
|  | |
| #define TFT\_GREY 0x5AEB | |
|  | |
| uint32\_t targetTime = 0; // for next 1 second timeout | |
|  | |
| static uint8\_t conv2d(const char\* p); // Forward declaration needed for IDE 1.6.x | |
|  | |
| uint8\_t hh = conv2d(\_\_TIME\_\_), mm = conv2d(\_\_TIME\_\_ + 3);  uint8\_t ss = conv2d(\_\_TIME\_\_ + 6); // Get H, M, S from compile time | |
|  | |
| byte omm = 99, oss = 99; | |
| byte xcolon = 0, xsecs = 0; | |
| unsigned int colour = 0; | |
|  | |
| void setup() { | |
| //Serial.begin(115200); | |
| M5.begin(); | |
| // M5.Lcd.setRotation(1); | |
| M5.Lcd.fillScreen(TFT\_BLACK); | |
|  | |
| M5.Lcd.setTextSize(1); | |
| M5.Lcd.setTextColor(TFT\_YELLOW, TFT\_BLACK); | |
|  | |
| targetTime = millis() + 1000; | |
| } | |
|  | |
| void loop() { | |
| if (targetTime < millis()) { | |
| // Set next update for 1 second later | |
| targetTime = millis() + 1000; | |
|  | |
| // Adjust the time values by adding 1 second | |
| ss++; // Advance second | |
| if (ss == 60) { // Check for roll-over | |
| ss = 0; // Reset seconds to zero | |
| omm = mm; // Save last minute time for display update | |
| mm++; // Advance minute | |
| if (mm > 59) { // Check for roll-over | |
| mm = 0; | |
| hh++; // Advance hour | |
| if (hh > 23) { // Check for 24hr roll-over (could roll-over on13) | |
| hh = 0; // 0 for 24 hour clock, set to 1 for 12 hour clock | |
| } | |
| } | |
| } | |
|  | |
|  |  | |
|  | // Update digital time | |
|  | int xpos = 0; | |
|  | int ypos = 85; // Top left corner ot clock text, about half way down | |
|  | int ysecs = ypos + 24; | |
|  |  | |
|  | if (omm != mm) { // Redraw hours and minutes time every minute | |
|  | omm = mm; | |
|  | // Draw hours and minutes | |
|  | if (hh < 10) xpos += M5.Lcd.drawChar('0', xpos, ypos, 8); // Add hours leading zero for 24 hr clock | |
|  | xpos += M5.Lcd.drawNumber(hh, xpos, ypos, 8); // Draw hours | |
|  | xcolon = xpos; // Save colon coord for later to flash on/off later | |
|  | xpos += M5.Lcd.drawChar(':', xpos, ypos - 8, 8); | |
|  | if (mm < 10) xpos += M5.Lcd.drawChar('0', xpos, ypos, 8); // Add minutes leading zero | |
|  | xpos += M5.Lcd.drawNumber(mm, xpos, ypos, 8); // Draw minutes | |
|  | xsecs = xpos; // Sae seconds 'x' position for later display updates | |
|  | } | |
|  | if (oss != ss) { // Redraw seconds time every second | |
|  | oss = ss; | |
|  | xpos = xsecs; | |
|  |  | |
|  | if (ss % 2) { // Flash the colons on/off | |
|  | M5.Lcd.setTextColor(0x39C4, TFT\_BLACK); // Set colour to grey to dim colon | |
|  | M5.Lcd.drawChar(':', xcolon, ypos - 8, 8); // Hour:minute colon | |
|  | xpos += M5.Lcd.drawChar(':', xsecs, ysecs, 6); // Seconds colon | |
|  | M5.Lcd.setTextColor(TFT\_YELLOW, TFT\_BLACK); // Set colour back to yellow | |
|  | } | |
|  | else { | |
|  | M5.Lcd.drawChar(':', xcolon, ypos - 8, 8); // Hour:minute colon | |
|  | xpos += M5.Lcd.drawChar(':', xsecs, ysecs, 6); // Seconds colon | |
|  | } | |
|  |  | |
|  | //Draw seconds | |
|  | if (ss < 10) xpos += M5.Lcd.drawChar('0', xpos, ysecs, 6); // Add leading zero | |
|  | M5.Lcd.drawNumber(ss, xpos, ysecs, 6); // Draw seconds | |
|  | } | |
|  | } | |
|  | } | |
|  |  | |
|  |  | |
|  | // Function to extract numbers from compile time string | |
|  | static uint8\_t conv2d(const char\* p) { | |
|  | uint8\_t v = 0; | |
|  | if ('0' <= \*p && \*p <= '9') | |
|  | v = \*p - '0'; | |
|  | return 10 \* v + \*++p - '0'; | |
|  | } | |
|  |  | |

|  |
| --- |
|  |
|  |
| //Update digital time  int xpos = 0; |
| int ypos = 85; // Top left corner to clock text, about half way down |
| int ysecs = ypos + 24; |
|  |
| if (omm != mm) { // Redraw hours and minutes time every minute |
| omm = mm; |
| // Draw hours and minutes  // Add hours leading zero for 24 hr clock |
| if (hh < 10) xpos += M5.Lcd.drawChar('0', xpos, ypos, 8); |
| xpos += M5.Lcd.drawNumber(hh, xpos, ypos, 8); // Draw hours |
| xcolon = xpos; // Save colon coord for later to flash on/off later |
| xpos += M5.Lcd.drawChar(':', xpos, ypos - 8, 8);  // Add minutes leading zero |
| if (mm < 10) xpos += M5.Lcd.drawChar('0', xpos, ypos, 8); |
| xpos += M5.Lcd.drawNumber(mm, xpos, ypos, 8); // Draw minutes |
| xsecs = xpos; // Save seconds 'x' position for later display updates |
| } |
| if (oss != ss) { // Redraw seconds time every second |
| oss = ss; |
| xpos = xsecs; |
|  |
| if (ss % 2) { // Flash the colons on/off  // Set colour to grey to dim colon |
| M5.Lcd.setTextColor(0x39C4, TFT\_BLACK); |
| M5.Lcd.drawChar(':', xcolon, ypos - 8, 8); // Hour:minute colon |
| xpos += M5.Lcd.drawChar(':', xsecs, ysecs, 6); // Seconds colon  // Set colour back to yellow |
| M5.Lcd.setTextColor(TFT\_YELLOW, TFT\_BLACK); |
| } |
| else { |
| M5.Lcd.drawChar(':', xcolon, ypos - 8, 8); // Hour:minute colon |
| xpos += M5.Lcd.drawChar(':', xsecs, ysecs, 6); // Seconds colon |
| } |
| //Draw seconds  // Add leading zero |
| if (ss < 10) xpos += M5.Lcd.drawChar('0', xpos, ysecs, 6); |
| M5.Lcd.drawNumber(ss, xpos, ysecs, 6); // Draw seconds |
| } |
| } |
| } |
|  |

// function to extract number from compile string (don’t need to understand this for now)

|  |  |
| --- | --- |
| static uint8\_t conv2d(const char\* p) { | |
| uint8\_t v = 0; | |
| if ('0' <= \*p && \*p <= '9') | |
| v = \*p - '0'; | |
| return 10 \* v + \*++p - '0'; | |
| } | |
|  |  | |

**Task 3: Create stop watch:**

The stop watch should the show hour, minute, second and 1/100th second, where button A is start, button B is stop and button C is reset.

0:01:23

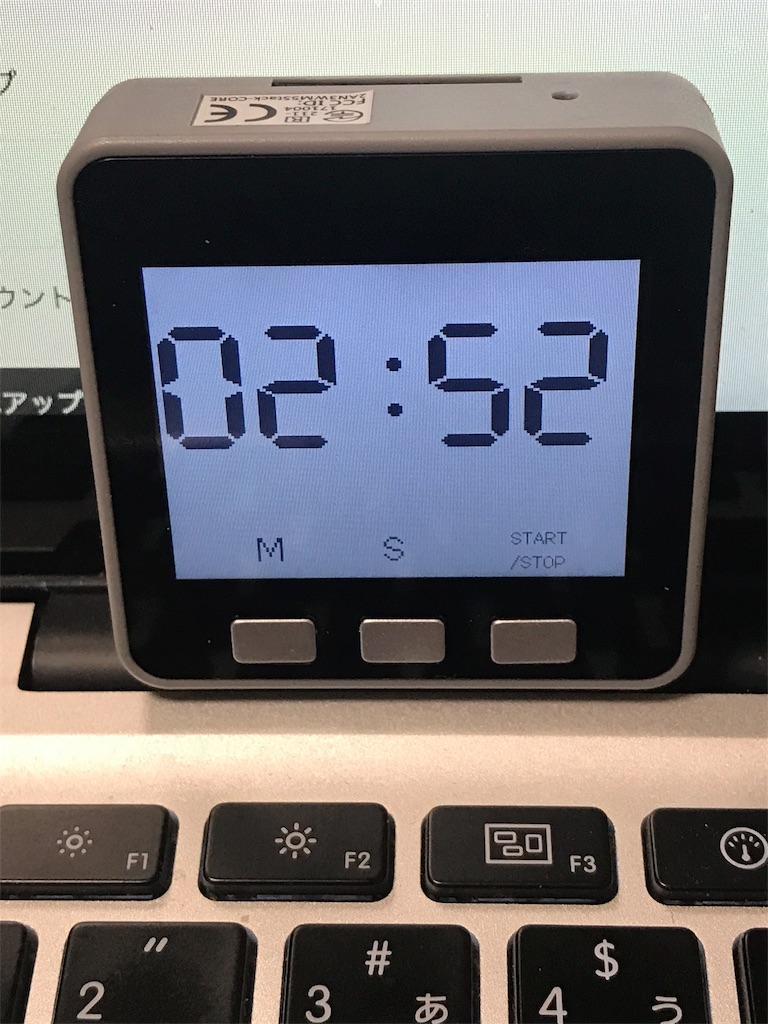
Start

Reset

Stop

**Task 4: Create Countdown timer:**

Button A for adjusting minute, Button B for adjusting second, Button C is to start or stop timer. Button A and B is working as following: each press will increase the value of minute or second by one, if the button A is pressed longer than 5 seconds the value of minute become zero. If the button B is pressed longer than 5 seconds the value of second become zero. When the counter is running, pressing button A or B has no effect, you must press stop button first. Each press on Button C will toggle the status of the timer (start/stop). If you press Button C long enough (more than 5 seconds) the value of counter become 0:00. Once the time reach 0.00, it should give a beep or some musical tone until any button is pressed.



**Section\_\_\_\_\_\_\_\_\_\_\_\_\_date\_\_\_\_\_\_\_\_\_\_ Group No.\_\_\_\_\_\_\_\_\_**

**Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Student ID\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Student ID\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Student ID\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Laboratory 4:Speaker, Digital alarm clock**

**Task1: Speaker, Jingle Bell**

Graded by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Task2: Basic digital clock**

Graded by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Task 3: Stop watch**

Graded by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Task 4: Count down timer**

Graded by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_