HANDOUT

Project X Session 5 - Programming the Mood Card

Oli Sharratt, Junzhe Chen, and Yuxuan Han

Introduction
In the fifth (and final) Project X session, you will be introduced how to program the *ESP32-C3 SuperMini* on your Mood Credit Card to display a representation of your current heart rate.

Contents

1. Programming

1.1 Setting up the File

To begin with, we need to setup the file for all components to operate as expected. The MAX3010x library need to be installed to make everything up and running. To do this, go to `Tools/Manage Libraries` and search for `MAX3010x` to install the correct library. You should obtain something shown in Fig. 1

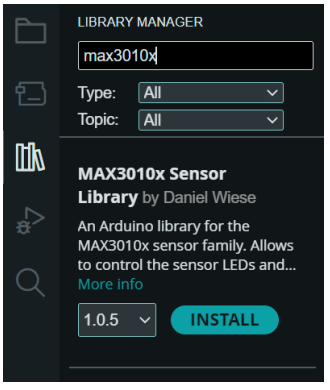


Figure 1. Install the library for the sensor

Each LED is connected to a different pin on the microcontroller, which have different names that can be used to address them - a digital pin can be referred to by its number (e.g. `13` for digital pin 13), and an analogue pin can be referred to by `A` followed by its number (e.g. `A2` for analogue pin A2). Looking at the PCB schematic, we are using pins `A0` to `A4`, as shown in Fig. 2.

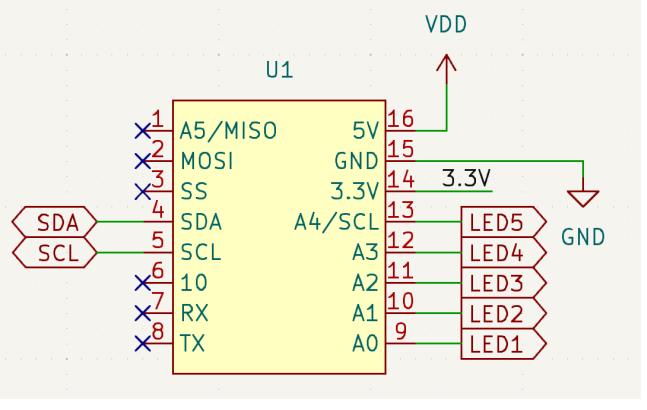


Figure 2. PCB Schematic of Generic Board Showing Microcontroller Connections to LEDs.

Task: Define the pins of each LED using the *#define* command to variable names (e.g. LED1, LED2, etc.), then try to light them up.

Depending on how you choose to address the LEDs, the `pinMode` command may be required or not. See section 1.3 for more information.

1.2 Getting a value of BPM

To receive an integer value of the *beats per minute (BPM)*, call the function `readBPM()`. When this is assigned to a variable (in the scope of `main()`), this value will update every time the loop runs. Then you can use it for LEDs representation in whatever way you choose.

Task: Declare a temporary variable to store the integer value of BPM above the *setup()* section. In *loop()*, set the output of *readBPM()* to this variable.

```
============ After Setup End =============
The BPM reading is: -1
The BPM reading is: -1
The BPM reading is: -1
The BPM reading is: -1
The BPM reading is: -1
The BPM reading is: 0
The BPM reading is: 0
The BPM reading is: 0
The BPM reading is: 23
The BPM reading is: 60
The BPM reading is: 60
The BPM reading is: 64
The BPM reading is: 66
The BPM reading is: 65
The BPM reading is: 65
The BPM reading is: 66
The BPM reading is: 66
```

Figure 3. Screenshot of Serial Monitor after obtaining the sensor reading

1.3 Addressing LEDs

Outputting to an LED can be done in a few different ways and produce different results. We have two commands we can use: *digitalWrite()* or *analogWrite()*. A digital pin will use a digital signal (logic 0 (LOW) or logic 1 (HIGH)), and will be a fixed brightness. Using the `digitalWrite()` command will take two values; the LED pin being addressed (via its corresponding variable name), and the logic level desired - e.g. `digitalWrite(LED1, HIGH)`. However, as stated earlier, an additional command is required *pinMode()* needs to be defined for each pin. This command belongs in the `setup()` function and takes two arguments - the LED being addressed, and the mode it should operate in (INPUT, INPUT_PULLUP, or OUTPUT). In this case, LEDS are output devices, so all LED pins should be set to OUTPUT, e.g. `pinMode(LED1, OUTPUT)`.

An analogue pin will output a Pulse Width Modulated (PWM) signal which is a square wave of varying on/off time and can provide a range of brightness values according to the ratio of time on to time off. Once again, this will take two values: the LED pin being addressed and the output level. The output level, is a value between 0-255, simulating 0V to 3.3V. This should correlate with the brightness of the LED, and `analogWrite(LED1, 128)` should be approximately half brightness.

2. Troubleshooting

2.1 Boot Mode & Reset Button

If you get an error of '`Exit status 2`' and the code does not upload, unplug the board, then plug it back in whilst holding the `BOOT` button on the ESP32-C3. See Fig. 4a for more details. This should fix the uploading issue, but the board still may not be working correctly - try pressing the `RST` (or Reset) button, to see if that helps.

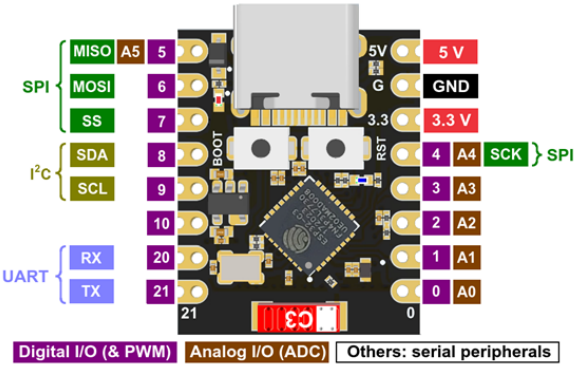
2.2 I'm not getting a reading from the sensor!

This is a common issue and can be related to a number of things. First, check that the PCB is powered, the microcontroller has LEDs on, and the heart rate sensor is glowing red. The second issue may be related to the positioning of the finger on the sensor. When using it, hold still, and apply consistent pressure. However, don't press too hard, as no reading will be given, but also don't press to lightly, as this will give an erroneous reading.

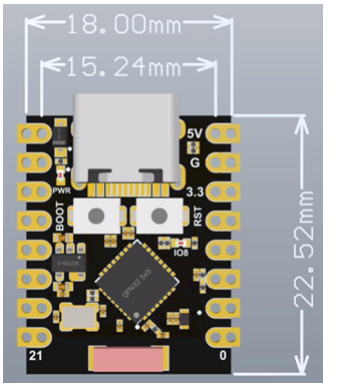
Congratulations

Congratulations for making it to the end! You should now have a fully functional heart-rate sensing device.

3. Appendix I: ESP32C3 Super Mini



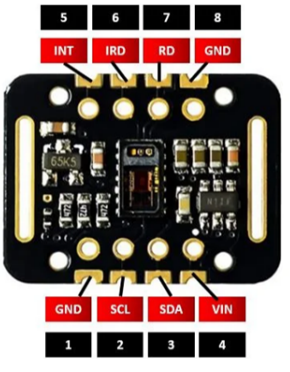
(a) Pinout of the ESP32C3 Super Mini Board



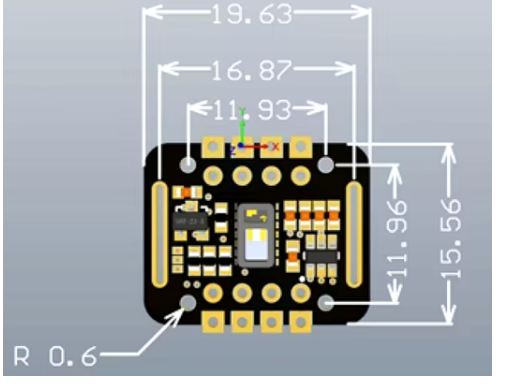
(b) Dimensions of the ESP32C3 Super Mini Board

Figure 4. ESP32C3 Super Mini Board pinout and dimensions.

4. Appendix II: MAX30102



(a) Pinout of MAX30102



(b) Dimensions of MAX30102

Figure 5. MAX30102 pinout and dimensions.