

# EEE3096S - PRAC 1

BRMKEA001 and CLLSTE009

18 Aug 2021

---

## 1. What is I2C?

I<sup>2</sup>C is a communication protocol for integrated circuits and microcontrollers

## 2. What is an interrupt and why is it important in the world of Embedded Systems?

An interrupt takes preference over code that is being run in the main CPU thread. It is a way of calling for CPU time on a more urgent task. Usually I/O uses interrupts such as buttons.

## 3. Embedded Systems Good Practices

### a) Why do we use pull up and pull down resistors?

A pull up resistor is used to pull the microcontroller pin high when the button is not pressed and when the button is pressed it pulls the pin low. The pull-down resistor does the exact opposite. Choosing a pull up or pull down will determine what type of edge detection should be used. The resistor also limits the amount of current that is sunk or sourced by the microcontroller pin.

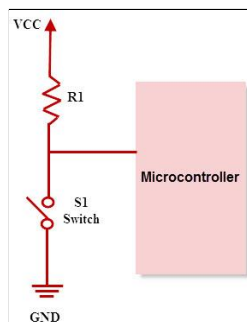


Figure 1: Pull-up resistor

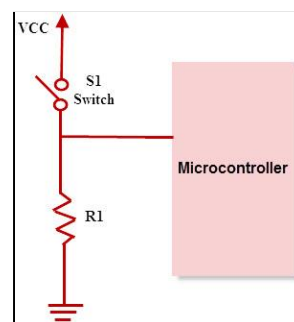


Figure 2: Pull-down resistor [1]

### b) Explain the difference between hardware debouncing and software debouncing, with an example of how you might implement each (i.e draw a circuit diagram and write a code snippet).

When a button is pressed, the mechanical switch can cause the analog electrical signal to fluctuate in the process of making contact with the circuit lines, resulting in the digital signal to trigger the off and on state multiple times before stabilizing.

Hardware debouncing uses physical measures to prevent the button from triggering multiple times i.e. inserting a capacitor in parallel with the button and the resistor to smooth the transition from high to low.

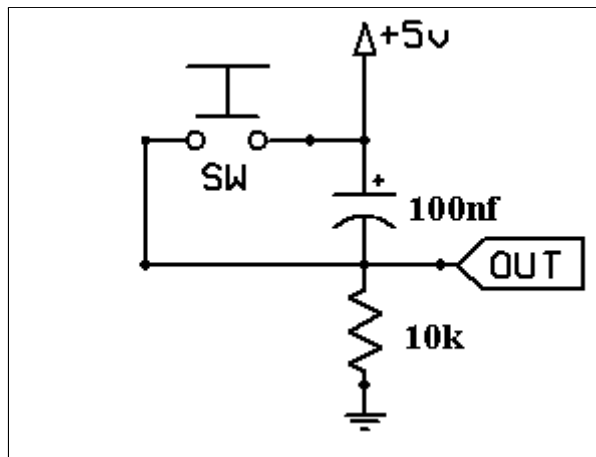


Figure 3: Hardware debouncing circuit [2]

Software debouncing makes use of code to prevent the button from triggering multiple times when being pressed. This can be in the form of checking if a specified duration of time has passed between interrupt calls before executing the function.

```
long lastInterruptTime = 0;
void handleInterrupt() {
    long interruptTime = millis();

    if (interruptTime - lastInterruptTime > 200) {
        // Execute code
    }
    lastInterruptTime = interruptTime;
}
```

**c) What is “polling” in this context, and why is it better to use an interrupt over polling?**

Polling refers to the process of repeatedly checking the status of an external device or input (i.e. a button) at regular intervals. An interrupt can be used to conditionally execute code without polling. Polling takes more CPU time, so using interrupts can save power in embedded systems.

#### 4. The circuit diagram of your implementation.

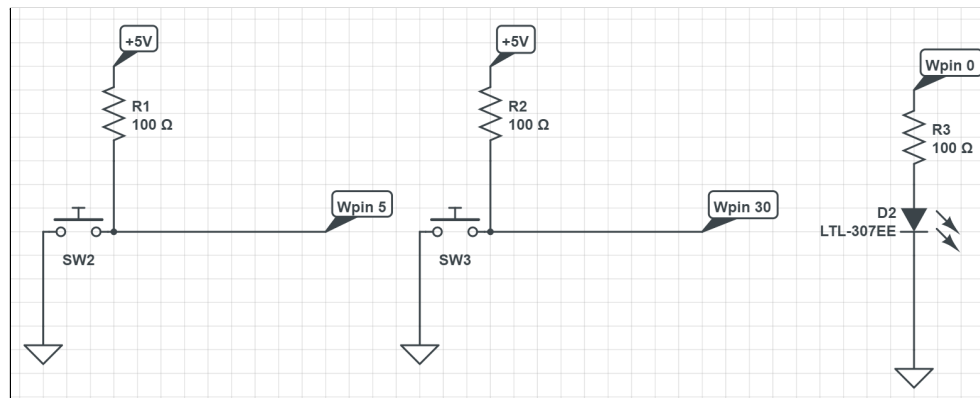


Figure 4: Circuit diagram

#### 5. The Core Functions of your code

Within the GPIOinit function the LED pin gets set to an output, and two interrupts with pointers to the hour inc and min inc function are set up.

There are 2 more functions that we wrote (Read time and write time). Read time is used to assign the global variables (hours, mins and secs) the hex compensated value that was read from the RTC. The write time function writes the dec compensated value of the hours, mins and secs back to the RTC.

The main loop toggles the LED on and off, with a 500ms delay between each state as well as the read time function being called.

Within the hour inc function the read time function is called, hours are then incremented by 1, then check for overflow and finally write time is called. The same happens in min Inc.

Finally within the Cleanup function, the LED and button pins are both made inputs and are set low.

#### 6. A Github link to your shared repository

<https://github.com/KealymB/EEE3096/tree/main/PRAC/P1>

## References

[1]"Working of Pull-up and Pull-down Resistors with Examples", *ElProCus - Electronic Projects for Engineering Students*. [Online]. Available: <https://www.elprocus.com/pull-up-and-pull-down-resistors-with-applications/>. [Accessed: 18-Aug- 2021]

[2]H. crosstalk and S. G, "Hardware buttons debouncing crosstalk", *Electrical Engineering Stack Exchange*, 2016. [Online]. Available: <https://electronics.stackexchange.com/q/232373/282847>. [Accessed: 18- Aug- 2021]