Introduction

In this project, an electronic system based on the Tiva™ TM4C123GH6PM microcontroller has been implemented. The task consisted of programming said microcontroller so it would be able to detect objects within an area of action, measuring light level and communicate the measurements.

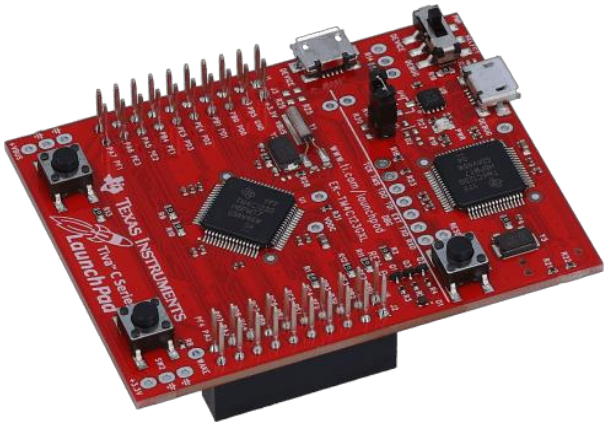


Figure 1: Tiva™ TM4C123GH6PM microcontroller

The programming will be used in CCS (code composer studio), a powerful tool with integrated libraries for this kind of microcontroller.

Tasks

During the different Lab sessions, the system has been implemented following an incremental way, specifically following the following steps:

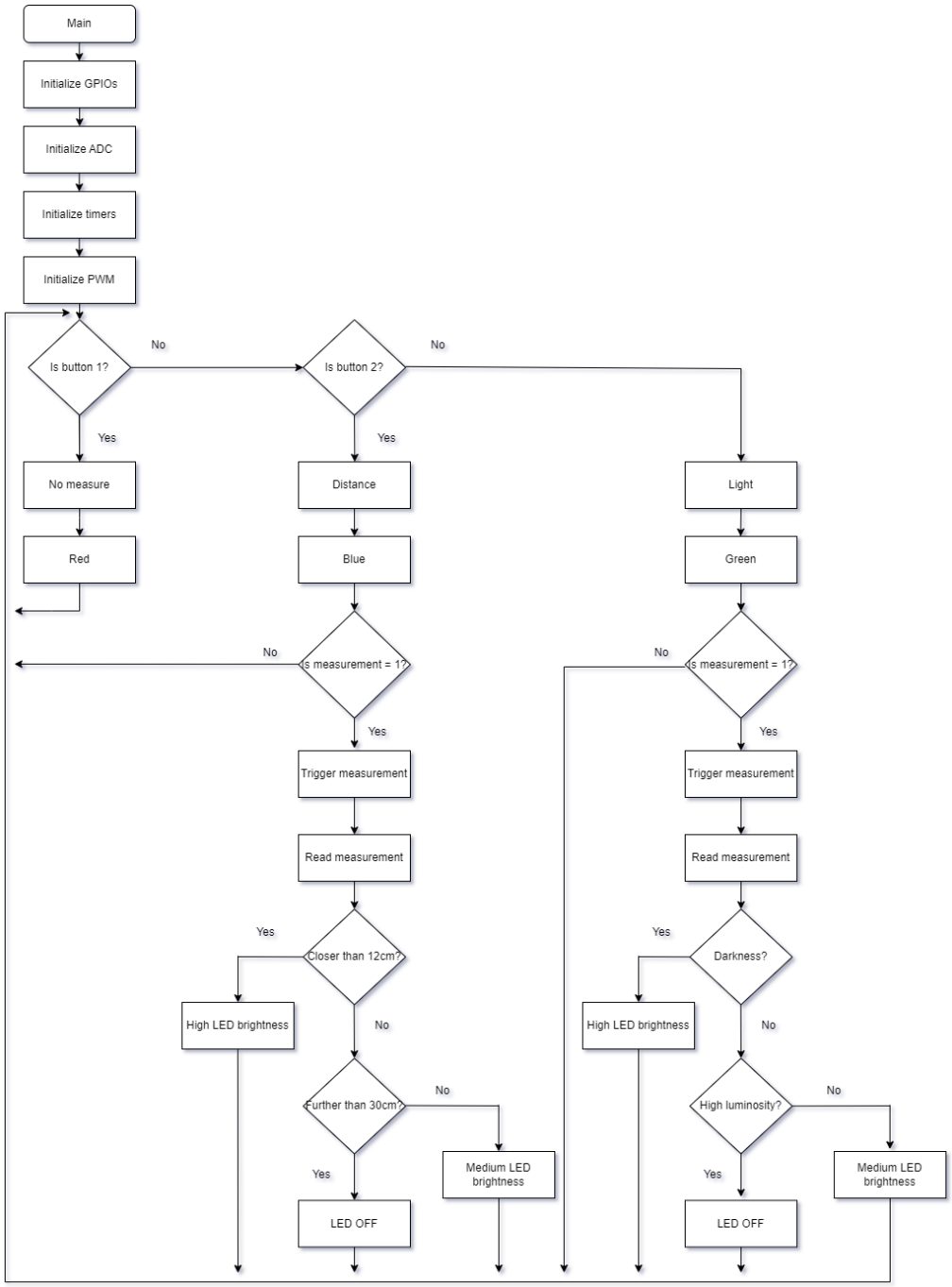
- Task 1: Selecting the mode
  - In this task, preliminary steps of the system were achieved. The mode change when pushing the button was implemented, as well as a different LED color associated to each mode; *No Measure*, *Distance* and *Light* will have a *Red*, *Blue* and *Green* LED color, respectively.
- Task 2: Scheduling the measurements
  - To continue, the measurements were implemented. They would be performed periodically when the mode is *Distance* and *Light*, and these said modes would have associated a blinking period of 1s, just like the measurements.
- Task 3: Distance mode
  - A white LED is connected to the microcontroller. This LED will vary depending of the measurements the system performs. The Ultrasonic ranging module HC - SR04 is now also connected to the microcontroller. The LED intensity will vary as follows:
    - High white LED brightness: Objects closer than 12 cm.
    - Low white LED brightness: Objects in the range 12-30 cm.
    - Off: Objects further than 30 cm.

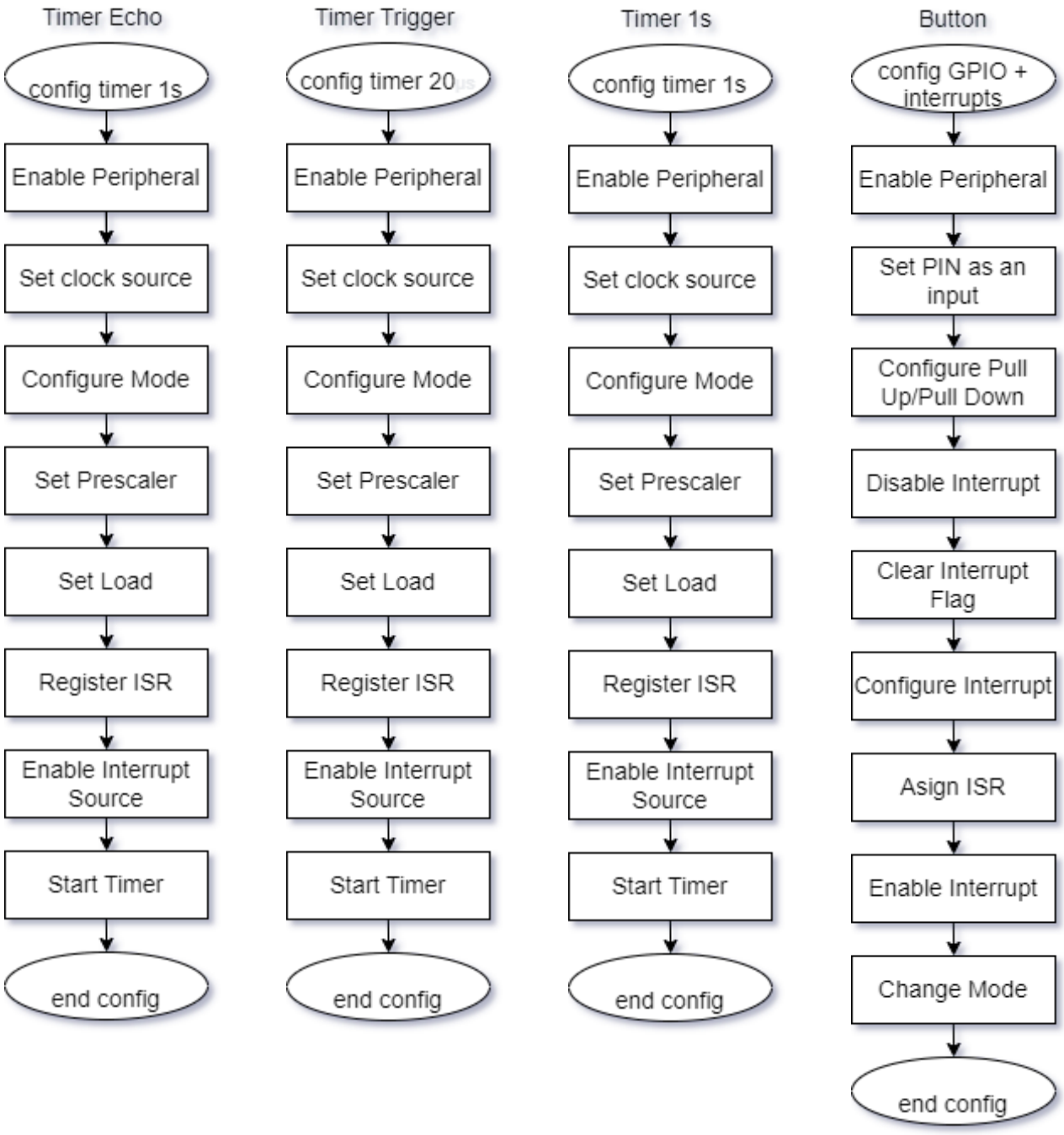
- Task 4: Light mode
    - Making use of the integrated light sensor of the microcontroller, the white LED will perform as follows:
        - High white LED brightness: Darkness condition.
        - Low white LED brightness: Room light level.
        - Off: High luminosity.
- Task 5: Final behavior
    - The system must be implemented in a way that handles the interchanging of modes and the performance of the overall system.

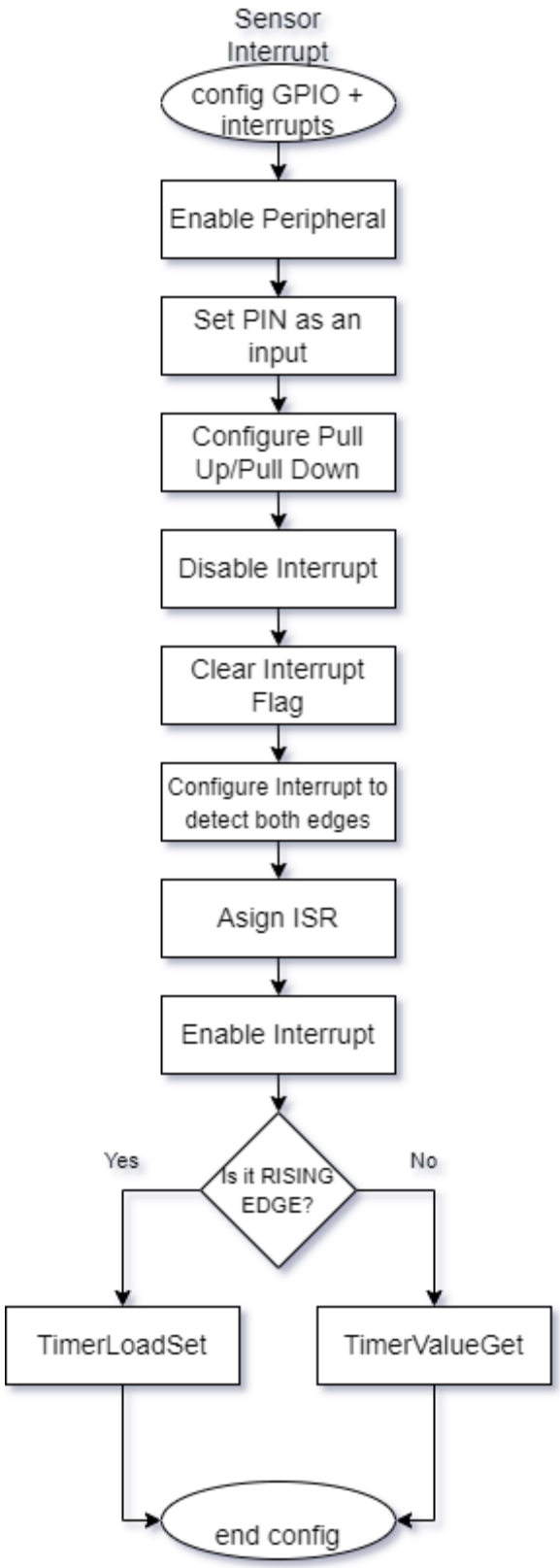
All tasks have been able to be implemented, and the system performs as expected.

Flowcharts

To have a clear and general view of the path that the code had to follow, a flowchart was drawn.

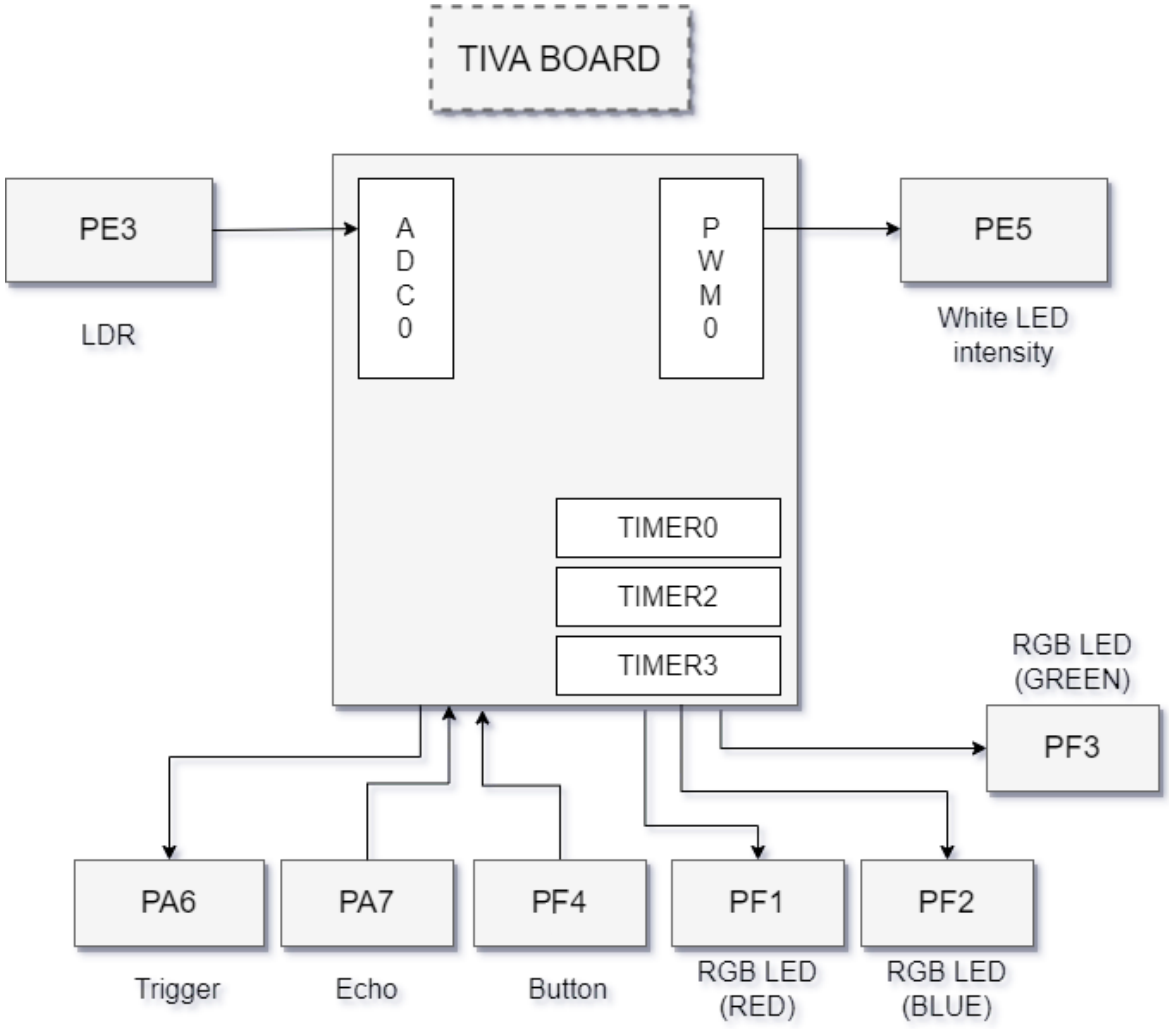



Timer Echo
config timer 1s
Enable Peripheral
Set clock source
Configure Mode
Set Prescaler
Set Load
Register ISR
Enable Interrupt Source
Start Timer
end config
Timer Trigger
config timer 20
Enable Peripheral
Set clock source
Configure Mode
Set Prescaler
Set Load
Register ISR
Enable Interrupt Source
Start Timer
end config
Timer 1s
config timer 1s
Enable Peripheral
Set clock source
Configure Mode
Set Prescaler
Set Load
Register ISR
Enable Interrupt Source
Start Timer
end config
Button
config GPIO + interrupts
Enable Peripheral
Set PIN as an input
Configure Pull Up/Pull Down
Disable Interrupt
Clear Interrupt Flag
Configure Interrupt
Asign ISR
Enable Interrupt
Change Mode
end config

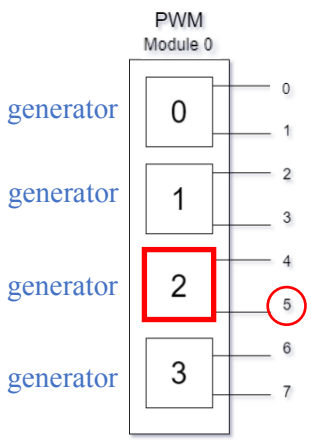

Sensor Interrupt
config GPIO + interrupts
Enable Peripheral
Set PIN as an input
Configure Pull Up/Pull Down
Disable Interrupt
Clear Interrupt Flag
Configure Interrupt to detect both edges
Asign ISR
Enable Interrupt
Is it RISING EDGE?
Yes
No
TimerLoadSet
TimerValueGet
end config

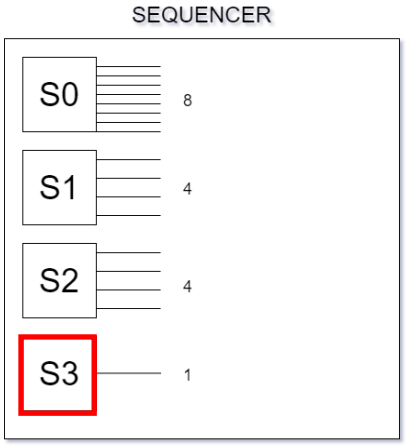
Block diagram

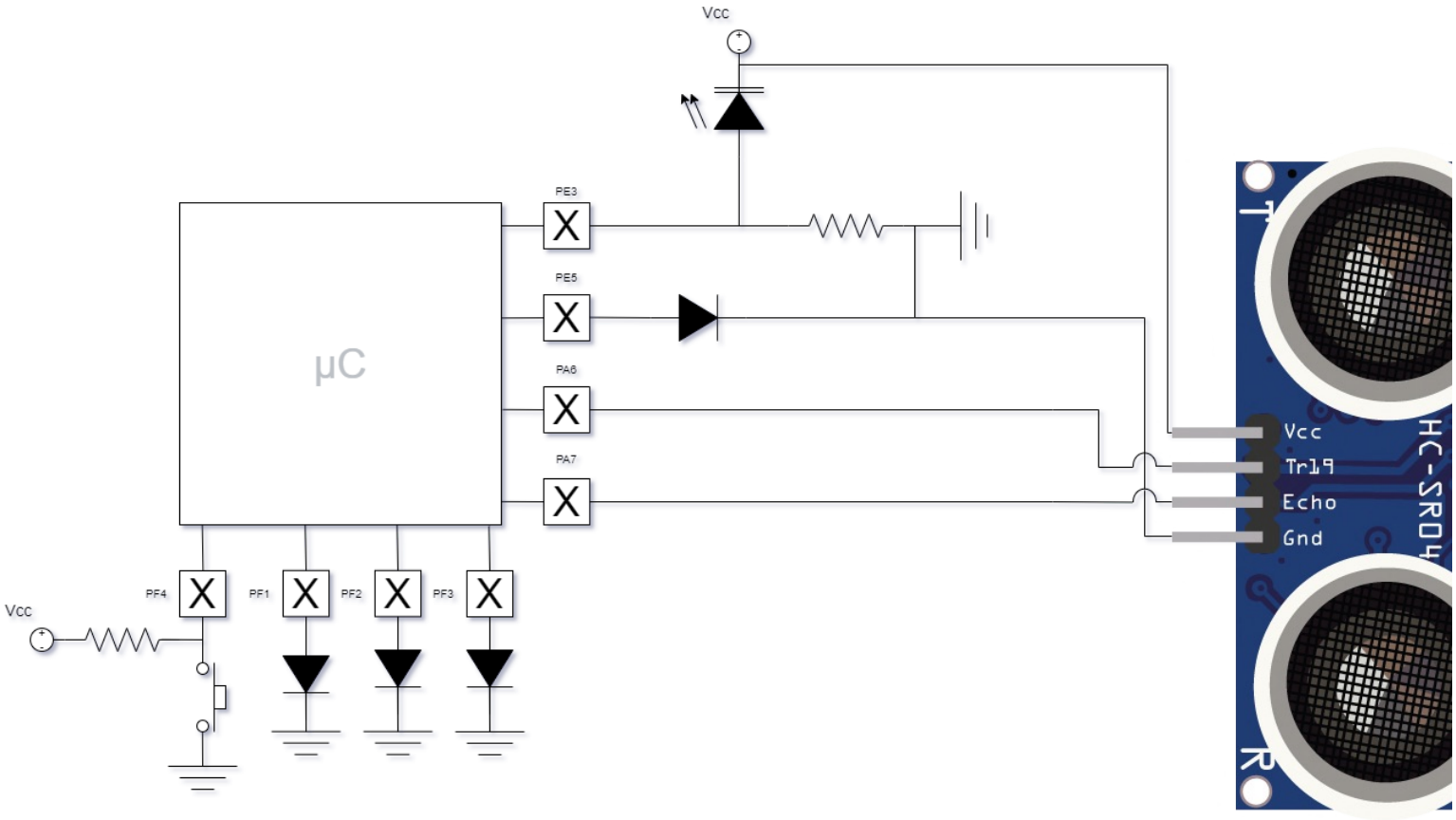
A block diagram of the system has also been drawn:



Electronics schematics







Conclusion

Throughout this project, valuable knowledge in microcontroller programming was acquired, one skill that will be important in the future. Understanding a new programming language, C, was the most challenging aspect of this project, but ultimately, was the most rewarding one. Overall, it was a different and entertaining lab session.