Lab 3 – Prototype Phase 1

ECE 298 – S2021

Overview

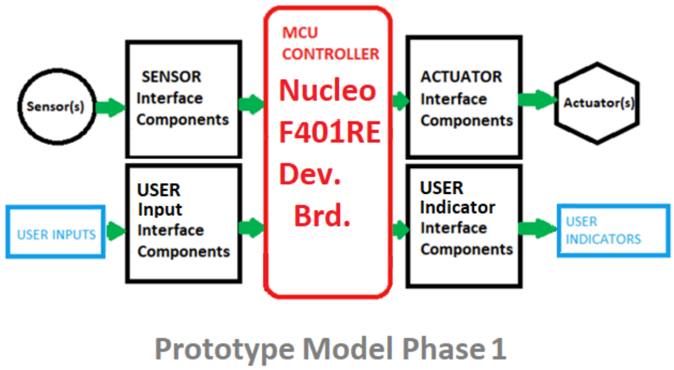


Figure 1 – Lab 3 system-level overview.

The design knowledge you gained from the Feasibility Model Phase is now used in the Prototype Model Phase. A large project may have many iterations of prototyping before a product is production-ready. Here we are aiming to complete one prototyping cycle, which will consist of using real-world parts and circuits for your design (Lab 3) and creating a printed circuit board (PCB) layout for it (Lab 4). The STM32 MCU will interface with and control your design. Proteus will simulate both the MCU's operation based on your compiled code (a digital simulation) in conjunction with your circuit components' simulation (an analog simulation). This kind of combined digital/analog design and simulation is called **mixed-signal**.

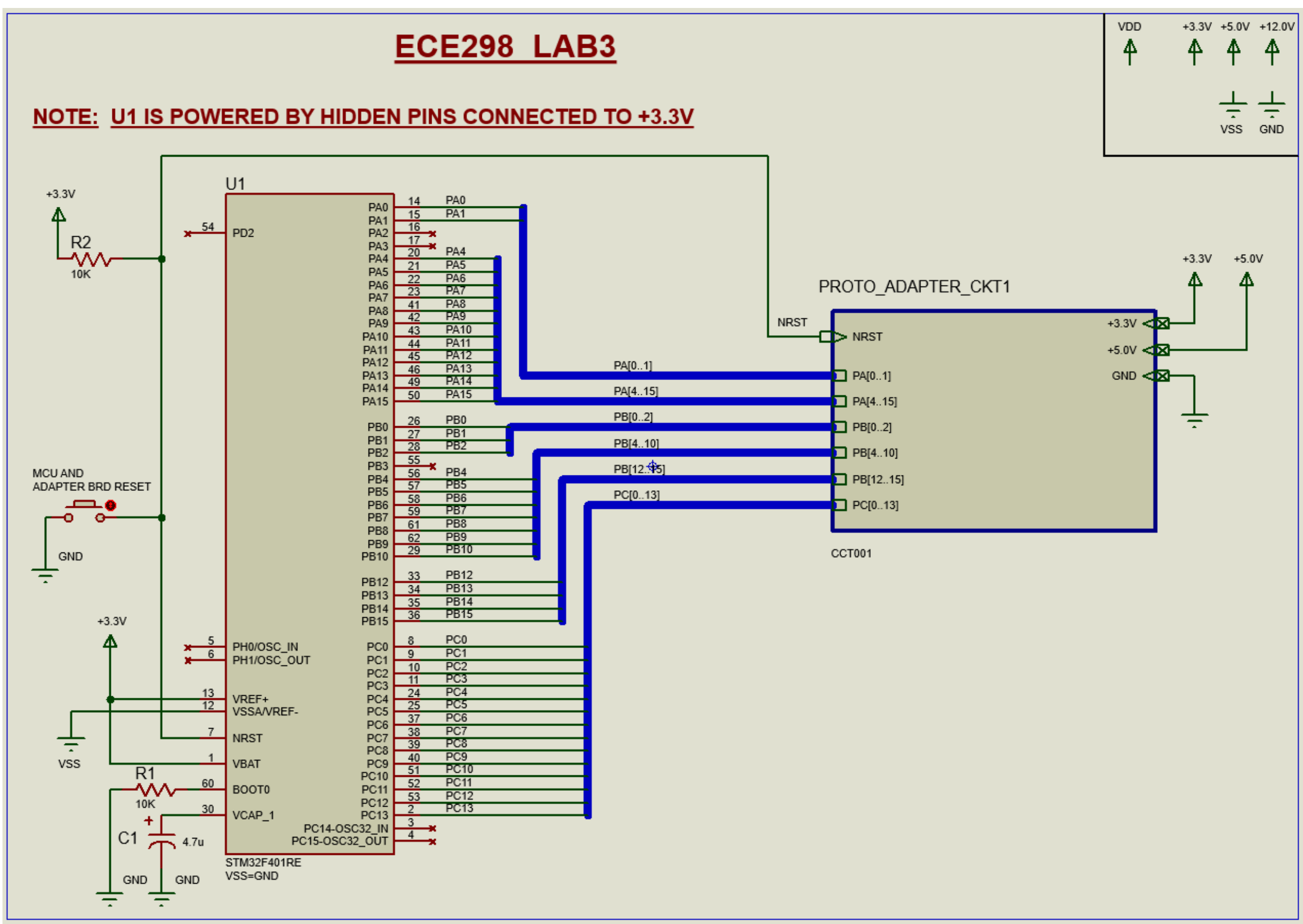


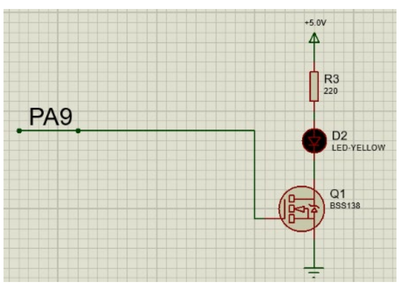
Figure 2 – Lab 3 Schematic. The PROTO_ADAPTER_CKT1 subsystem will hold your design.

Again, you will use the Proteus tools to create a Prototype Model schematic within the PROTO_ADAPTER_CKT1 subsystem, which is a hierarchical container for your previous schematics (Figure 2). You will upgrade your design to include all component models, connections, and embedded MCU code to represent your project (minus the PCB layout at this point).

Recall that the sensors and actuators' electrical attributes must be considered to interface them with the MCU pins reliably. In your whole design, all device and MCU pin specifications must be satisfied for "in-spec" operation to eliminate part electrical stress. For example, do not directly drive a 5 V pin with a 3.3 V pin or directly drive a 20 mA LED from a pin that can only supply 1 mA.

Copy your Lab 2 schematic work and paste it within the "child" sheet under the symbol called **PROTO_ADAPTER_CKT1**. Notice that all the port pins on the **PROTO_ADAPTER_CKT1** sheet are connected to the MCU pins through the "parent" sheet schematic. Replace any signals currently driven by Proteus generators by connecting specific MCU pins. For example, a 0 V to 3.3 V digital signal should come from an MCU pin configured as a GPIO digital output.

The MCU pins can provide various functions (General-Purpose I/O, SPI, etc.) but not all at once. You must be selective. Read the document on LEARN called "**ECE 298 W2021 Using the STM32CubeIDE with Proteus**", which explains how to use the STM32CubeIDE tool in selecting MCU pins for distinct functions. **There are some critical points in that document to link your compiled load file (.HEX) to your Proteus Lab 3 project.** The document provides a simple practice run of the embedded software/hardware integration process (Blink_Test).



Work through the Blink_Test example to see how to link your MCU code with your Proteus project simulation. You can follow the same procedure when you add your project's MCU pin choices. Start by selecting the pins for MCU peripheral resources (Timers, USART, etc.), then add your GPIO pin selections. This is because specific pins are used for the peripherals, but almost any pin can be a GPIO. Also, develop one kind of MCU peripheral function at a time to simplify debugging.

There are example STM32 starter files on LEARN to jumpstart your peripheral code development.

Deliverables

Submit a single .zip file that contains:

- A completed Lab 3 document template (PDF preferred), following the subsequent instructions
- Proteus Project that has your circuits successfully simulating with your MCU code
- STM32CubeIDE project directory that includes the .HEX file required by Proteus
- Any instructions the TAs and Instructors need to run the simulations to exercise all project requirements

Part 1 – Pin Mapping

Complete the table for all the MCU pins used in your design. You do **not** need to list those pins auto-assigned by the IDE configuration tool because of their use on the Nucleo board (e.g. PA5 = LD2 Nucleo LED).

- MCU Pin = the MCU pin name (P{A,B,C}{0 … 15})
- Pin Mode = the mode selected in the IDE's pin configuration tool
- Functional Description = what the pin will do in your design

Part 2 – MCU Resources

Complete the table for all the MCU resources used in your design. You do **not** need to list those resources auto-assigned by the IDE configuration tool because of their use on the Nucleo board (e.g. PA5 = LD2 Nucleo LED).

- MCU Resource = the MCU resource in the "Categories" view of the IDE's pin configuration tool (e.g., GPIO, RTC, TIM4, USART6, etc.)
- Functional Description = what the resource will do in your design

Part 3 – Test Cases

Generate simulations that show your combined MCU code + Proteus circuits working together to exercise all the operational modes necessary to meet your project's requirements:

- For each test scenario:
  - Add a subheading with the device name to make the document easier to navigate (use an appropriate Heading style)
  - Summarize the test case to explain what is happening and why, including which requirement it is exercising.
    - If it's not working as anticipated, present your best guess as to why and what changes you would make to correct the situation. This is a normal part of prototyping – it's really the whole point. No marks deducted if you identify and describe the presence of errors or weaknesses.
    - Consider both software/hardware correctness AND test case correctness (if you are exercising the system appropriately).
    - Be sure to highlight how your circuits account for the specifications of the MCU pins and devices to which they are connected. For example, is the MCU pin driving more current than it should, or has your interface accounted for that? Have analog voltages been scaled to an appropriate range for an ADC input? Etc.
  - Copy/paste the relevant project schematic sections
  - Copy/paste the simulation results (graph, scope outputs)