Feasibility Model Design

F2019 – Edit this document into a deliverable.

|  |  |  |  |
| --- | --- | --- | --- |
| Lab Section: | 1 | Group: | 22 |

# System-Level Design

Our ECE 298 projects start with a conceptual architecture, like the block diagram in Figure 1a). Specific example in Figure 1b). **Replace this figure with a high-level block diagram of your system.**

## 

## Project Design Requirements

In PD 21 you learned about engineering requirements. they fall into three major categories, as follows:

1. **Functional requirements** are quantities that specify the performance of a design. They are related to the functions of the design, identified as answers to the question, "What does it do?" For example, a functional requirement for a coffee maker may specify the time required to brew a pot of coffee, a DC power supply may specify its maximum voltage, and a vehicle alarm system may specify how much noise it makes when it is set off
2. **Non-functional requirements** specify characteristics of the design that are not performance based. Theses are typically features or qualities that are desirable to the client. as
3. **Constraint requirements** place limits on the design space, and often reflect budget or other project limitations. For example, cost, weight, and noise.

The basic form of most of these requirements is the same: a short description, followed by a relationship (equals, less than, or greater than) and a value.

**State three to five major Functional Requirements that your project must meet to successfully solve your problem statement.**

1. Use two distance sensors, one on the front and the back of the skateboard, in order to detect objects at a distance less than or equal to 400cm**.**
2. When the device is first turned on there needs to be a user setup mode. The setup mode will allow the user to set the distances that they want the LEDs and sound to be activated. The minimum distance that can be used is 3cm and the maximum distance 400cm.
3. The rear-facing sensor will be set by the user to have three proximity thresholds. Each set range will have a different LED turn on to indicate how close objects behind the user are.
4. The forward-facing sensor will be set by the user to have two proximity thresholds. Each proximity threshold will have a different beep frequency. This will be used for when a pedestrian gets too close to the front of your board to warn them that you are there.

## Project Sensors and User Inputs

* List the types of sensors and user inputs you may require (light, sound, temperature, magnetic field).

The types of sensors that we will require are multiple ultrasonic distance sensors for the front and back off the bike. For user inputs, we will need the user to set the distances they want to use during the user setup process by using the two push buttons on the MCU.

* For each sensor and user input, list how you will connect it to the MCU, including additional interface components, if needed.
* To connect the distance sensor to the MCU we will connect two I/O pins to the trigger and echo of the sensor. The pin connected to the echo sensor will be in series with resistors to ensure the voltage returned to the pin is correct. We will also connect the 5V pin to the VCC pin of the sensor. Finally, the last pin of the sensor will be connected to a common ground set by the MCU.
* The user input we are using are the push buttons that are already connected to the MCU.

## Project Actuators and Indicators

* List the types of actuators and indicators you may require (e.g. light, sound, mechanical motion)

The types of actuators and indicators we require are red/yellow/orange/green LEDs, an audio piezo, and the LCD display.

* For each actuator and indicator, list how you will connect it to the MCU, including additional interface components, if needed.
* To connect the LEDs to the MCU we will connect an I/O pin to a resistor that is in series with the led. The other side of the led will be connected to the common ground.
* To connect the audio piezo to the MCU we will connect one I/O pin to the red wire of the speaker and have the black wire of the speaker connected the common ground.
* The LCD display is already connected to the MCU.

## Project MCU Peripherals

* List the resources inside the MCU that could be used to implement your project (e.g. ADC, timers, interrupts, GPIO functions).
* GPIO functions to connect the sensors, actuators and indicators to the proper pins for communication with the MSP430FR4133.
* Timers to create delays.
* PWM to output a signal to the audio piezo.
* Push buttons for user inputs of the distances for setup.
* LCD Display to display the distances during setup and use.
* List parameters that the software running on the MCU might require.

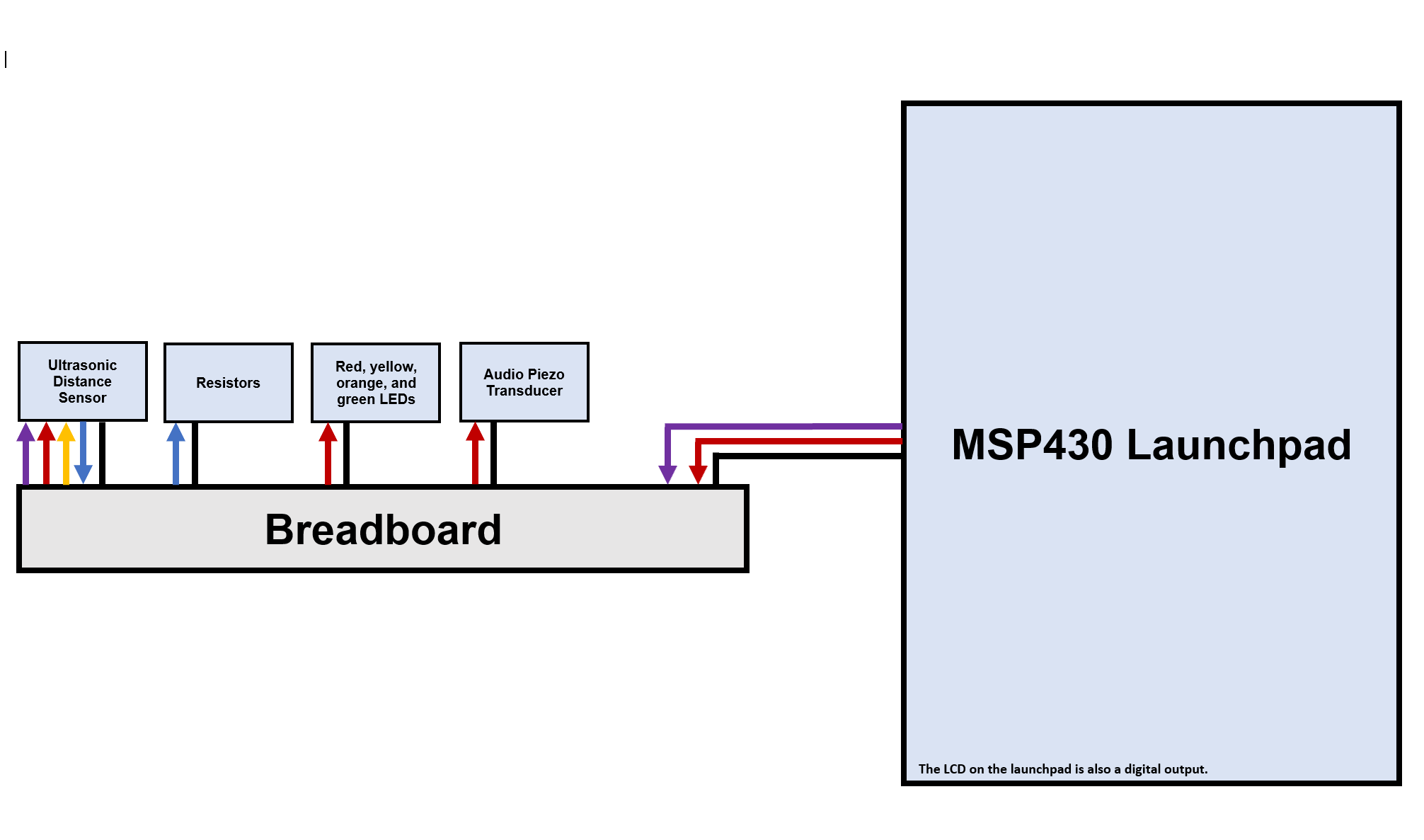
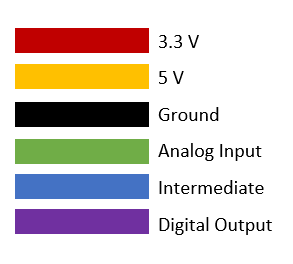
The parameters that the software on the MCU might require are a timer for calculating the distance, GPIO pin values, push button values to know when they are pressed, and the LCD display for displaying the distances.

## Project Testing Methodology

* For each sensor, user input, actuator, indicator, and MCU peripheral listed above, state how you will verify that each one is functioning as expected (a table may be helpful)
* State how you will validate that each Project Design Requirement has been met

# Feasibility Model Diagram and Software Flowchart (High-Level)

A simplified example is shown in Figure 2 and Figure 3. **Replace these figures with high-level block diagrams of your system.**



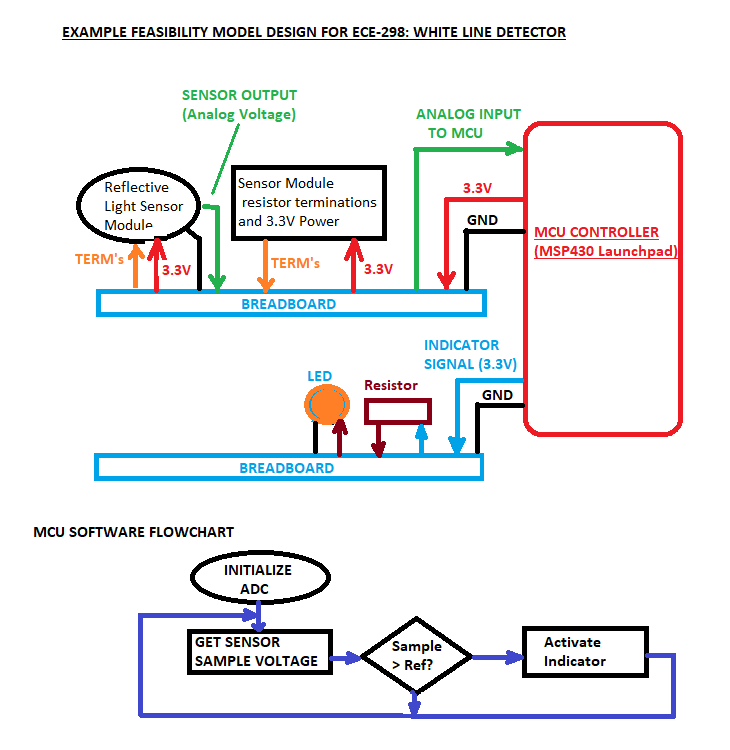


Figure 3: Simple Sketch of a Software Flowchart

## Initial Bill of Materials

* List what modules and components (including quantities) are needed from the ECE 298 Parts spreadsheet for your Feasibility Model Design

|  |  |
| --- | --- |
| **Modules/Components** | **Quantities** |
|  | 1 |
|  | 1 |
|  | 1 |
|  | 1 |
|  | 1 |
|  | 1 |
|  | 1 |