Feasibility Model Design

F2019 – Edit this document into a deliverable.

|  |  |  |  |
| --- | --- | --- | --- |
| Lab Section: | 6 | Group: | 2 |

# 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.**

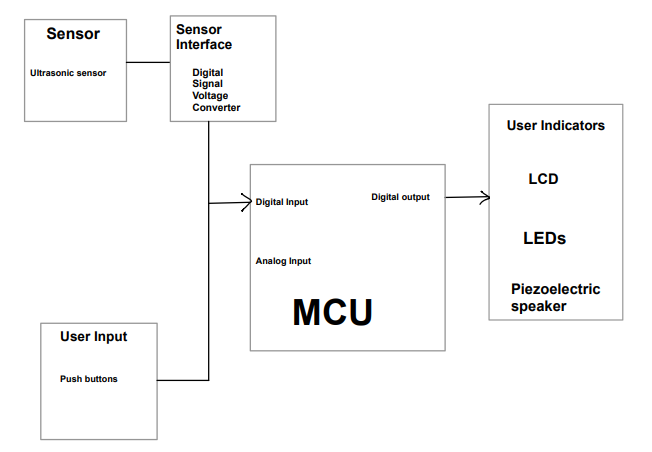


Figure 1b: Example using specific components and modules

## 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. For example, ease of use, ease of manufacturing, and use of recycled materials.
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. The project must display the output of the ultrasonic sensor readings, converting the sensor digital readings to a distance value to be displayed on the LCD.
2. The project must turn on a certain coloured LED corresponding to a distance value found from the front sensor value.
3. The project must also create 2 different sounds based upon the distance calculated from the back facing sensor value.
4. The project must also take in push button values that then correspond to setting up and configuring the MCU in either user mode or setup mode and the other push button should be able to set user threshold values.

## Project Sensors and User Inputs

* List the types of sensors and user inputs you may require (light, sound, temperature, magnetic field).
  + Ultrasonic sensor
  + Push Buttons
* For each sensor and user input, list how you will connect it to the MCU, including additional interface components, if needed.
  + Ultrasonic sensor is connected to the MCU using an interface component, the Digital Signal Voltage Level Converter which allows for voltage conversions between 3.3 V and 5 V.
  + Push Button which are attached through GPIO pins and input data that is interpreted on the MCU.

## Project Actuators and Indicators x

* List the types of actuators and indicators you may require (e.g. light, sound, mechanical motion)
  + LEDs x3 (Red, Orange, Green)
  + Audio transducer (piezoelectric speaker)
  + LCD
* For each actuator and indicator, list how you will connect it to the MCU, including additional interface components, if needed.
  + Each LED is connected with a resistance onto the GPIO pins.
  + Audio transducer connected to a GPIO pin.
  + LCD has native support on 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 pins; configuring as input/output pins, writing high/low turn LEDs on/off, buzz the piezoelectric speaker, and send trigger pulse to the ultrasonic sensor
  + GPIO interrupts to catch the rising and falling edge of the echo pulse from the ultrasonic sensor
  + Timer for measuring the width of the echo pulse sent from the ultrasonic sensor
* List parameters that the software running on the MCU might require
  + Ultrasonic sensor echo signal, push button inputs

## 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)

Sensor & user inputs:

|  |  |
| --- | --- |
| Push Buttons | Ultrasonic Sensor |
| Use push button to turn on LED and piezoelectric speaker | Use oscilloscope to view the echo response from the ultrasonic sensor and verify width increases as distance increases + display calculated distance in cm on LCD and see if it’s reasonable |

Actuator & Indicators:

|  |  |  |
| --- | --- | --- |
| LEDSs | Piezoelectric speaker | LCD |
| Intuitive – if they turn on they work | Intuitive – if it makes a sound it works | Display distance in cm from piezoelectric speaker on LCD |

MCU Peripherals:

|  |  |  |
| --- | --- | --- |
| GPIO pins | GPIO Interrupts | Timer |
| If LEDs and piezoelectric speakers work, GPIO pins working | Breakpoints in Code Composer Studio to confirm ISR is called | Breakpoints in Code Composer Studio and view the timer register to see if it is counting + if the distance displayed on LCD is accurate, the timer is working properly |

* State how you will validate that each Project Design Requirement has been met

|  |  |  |
| --- | --- | --- |
| Indicate distance to user | Inform user when something is close in the forward direction | Inform if something is too close in the backward direction |
| Display distance in cm on LCD | Buzz piezoelectric speaker from forward ultrasonic if it passes the thresholds | Turn on LEDs when the backwards distance passes specified thresholds |

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

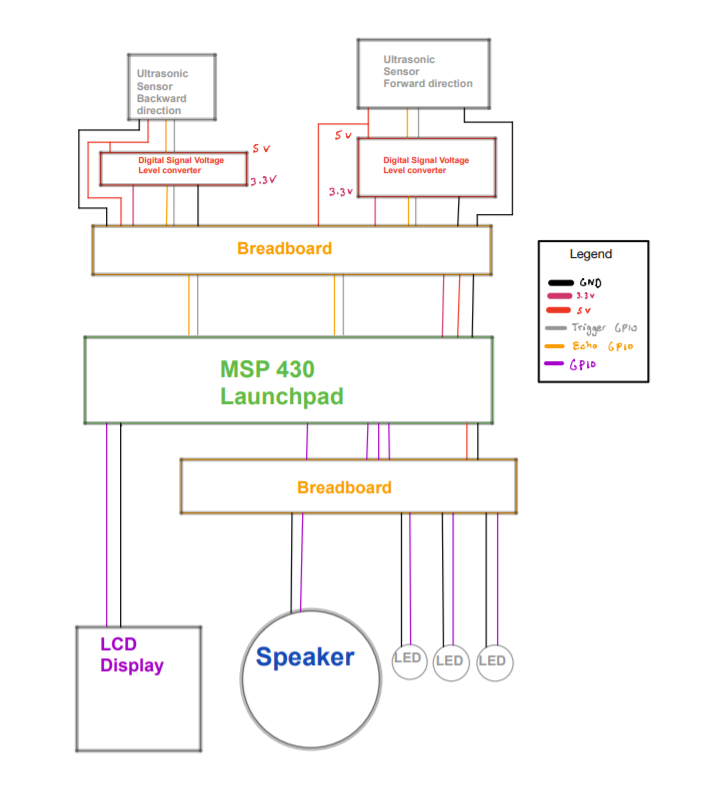


Figure 2: Simple Sketch of a Feasibility Model Design

## 

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PART NAME | Distributor Part No | ECE 398 DipTrace Part | RIGIDWARE SKU | RIGIDWARE PRICE (without TAX) | QUANTITY |
| MODULE - Ultrasonic Distance Sensor | Robotshop HC-SR04 |  | 4899825 | $2.40 | 2 |
| COMPONENT - AUDIO PIEZO TRANSDUCER 1-30V | "Digikey 433-1062-ND" |  | 4916287 | $0.85 | 1 |
| COMPONENT - RED LED - Diffused 5mm) | "Digikey 1497-1031-ND" | QPARTR5D34 | 4916296 | $0.40 | 1 |
| COMPONENT - ORANGE LED - Clear (5mm) | "Digikey 1497-1263-ND" | QPARTO5C34 | 4917122 | $0.40 | 1 |
| COMPONENT - YELLOW LED - Diff (5mm) | "Digikey 1497-1033-ND" | QPARTY5D34 | 4916429 | $0.40 | 1 |
| MODULE - Digital Signal Voltage Level Converter | Sparkfun BOB-12009 |  | 4923899 | $4.70 | 1 |