Project 3 Report

Skylar Kantor, CSE 231, Fall 2021

Introduction

This project is a speed limit sign, designed to be manufactured cheaply for use in rural areas. The sign calculates a safe speed limit for the current weather conditions and displays it on a lighted display, which is surrounded by LEDs that flash when the calculated speed limit differs from the dry speed limit by more than five miles per hour. This prevents situations where inexperienced drivers or those new to the area misjudge the risks of wet or icy conditions on certain roads and could lose control of a vehicle, causing an accident. Since the project uses easily available components such as 7-segment displays and a DHT11 humidity and temperature sensor, it is ideal for rural towns that may not have the budget for implementing other measures to increase road safety in wet, icy, or snowy conditions.

Specifications

- Produces a safe speed limit for any temperature and humidity in the range of conditions the DHT11 can tolerate
- Defaults to the normal posted speed limit when there is an issue with the DHT11
- Continues to run indefinitely without intervention

Features

- Lighted 7-segment display for readability in any light level
- Flashing LEDs to warn drivers of a speed limit change
- Tolerates sensor failure
- Error indicator LEDs for easy debugging
- Reacts in two seconds or less to changing weather conditions

Project Requirements

Since the project uses flashing LEDs, those are configured using bitwise driver control. The 3 indicator LEDs on the Nucleo are also configured with bitwise driver control. As LEDs are a binary on/off output, controlling the 3 indicator LEDs and 2 pairs of flashing LEDs on the display made sense to do with bitwise control, since that is how it was done in class examples and previous projects.

The project also has many threads attempting to access the temperature, humidity, and speed limit data. Since some threads access only the speed limit data, some access both the speed limit and temperature/humidity data, and others only access the temperature/humidity data, two mutexes are used, one for the temperature/humidity data and one for the speed limit, so that even if a thread is, for example, storing the current temperature and humidity from the sensor, another thread can display the most recently calculated speed limit, but the thread that is taking the temperature and humidity to calculate the new speed limit must wait until the sensor thread completes before accessing the shared data.

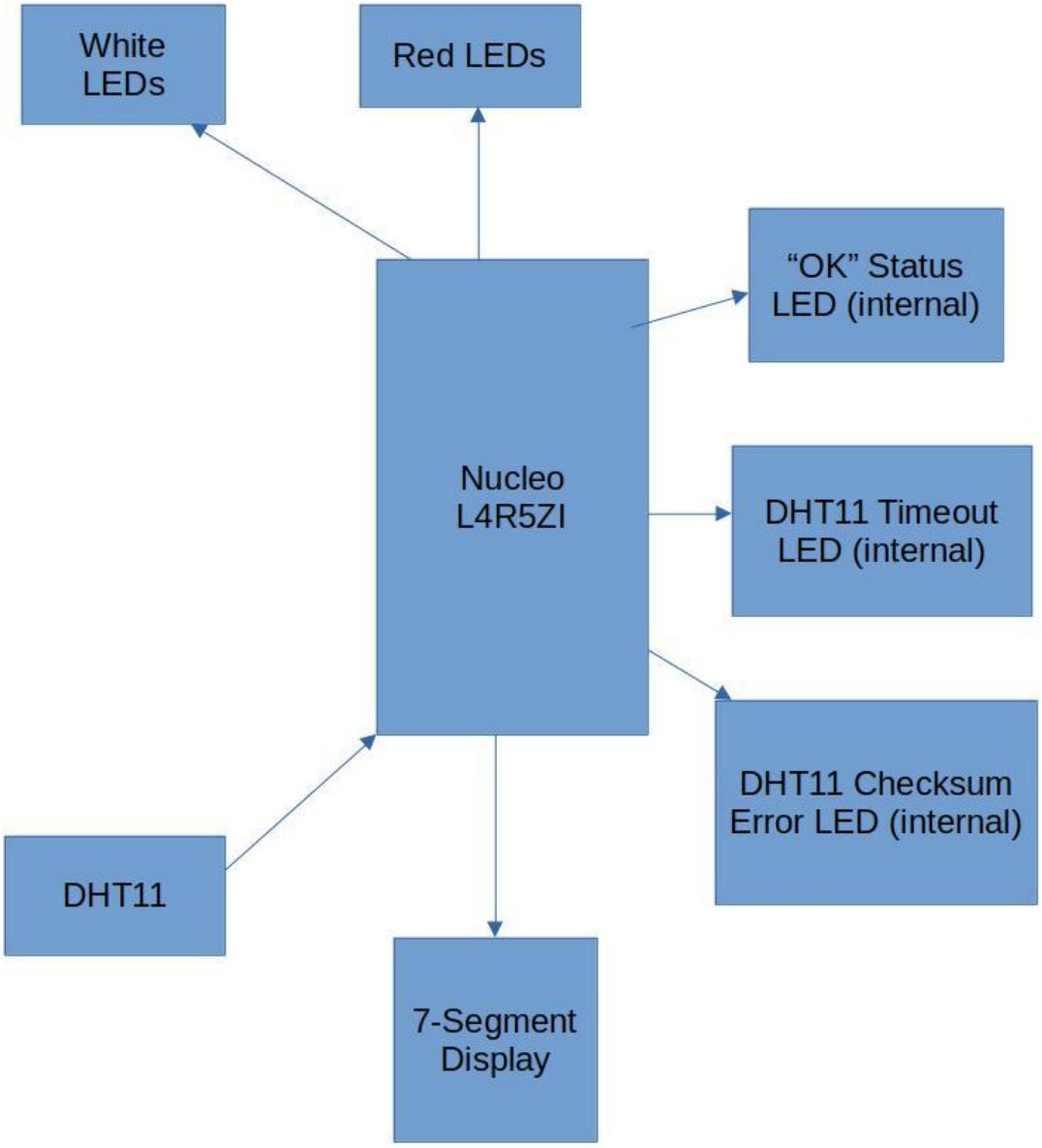
Since the DHT11 sensor can only be read every two seconds[1], a Ticker was used to ensure that the sensor was read as often as allowed, but not more often than once every two

seconds. The Ticker creates interrupts, that are then handled by an ISR that stores the temperature and humidity to shared variables, that are then used to calculate the speed limit. Two other Tickers ensure that as soon as the sensor is read, the new speed limit is calculated and then displayed on the 7-segment display.

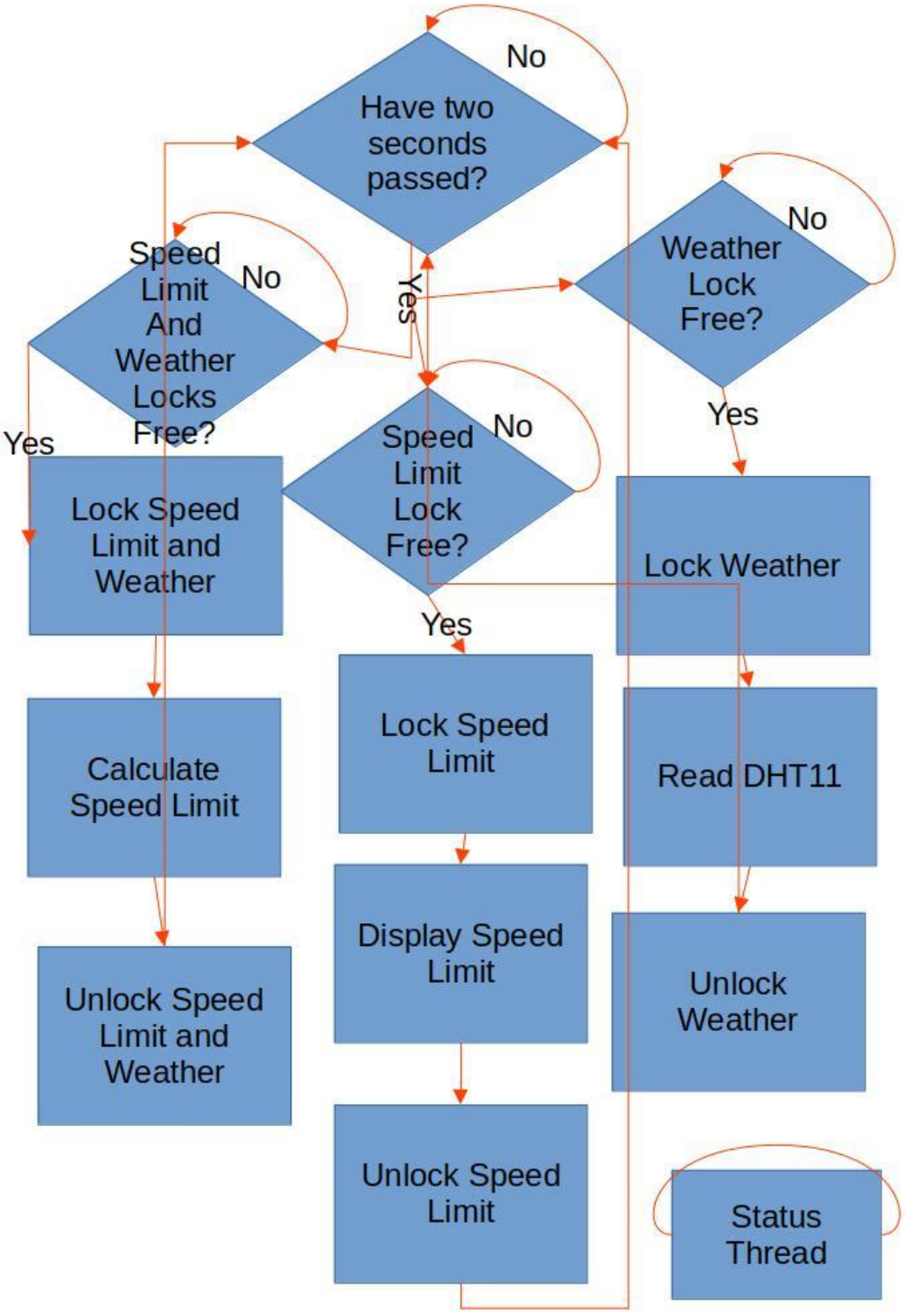
Using a mutex in an ISR requires the use of an EventQueue, which depends on threads, so each action done by the sign has its own thread. There is one for reading from the DHT11, one for writing to the display, and one for calculating the speed limit. There is also a separate thread that is not tied to an EventQueue, that checks the status of the DHT11 and sets the indicator LEDs on the Nucleo board if there are any errors detected.

This system is safety critical, and so the watchdog is set to ensure that the system doesn't take more than the time between two readings from the DHT11 before resetting to a safe state. The watchdog timeout is set to that time (two seconds), and then is fed at the end of each task done by the system (after reading the sensor, after writing to the display, and after calculating the speed limit). This means that each task should not take more than two seconds, which should ensure that the speed limit is updated frequently enough not to cause a safety issue.

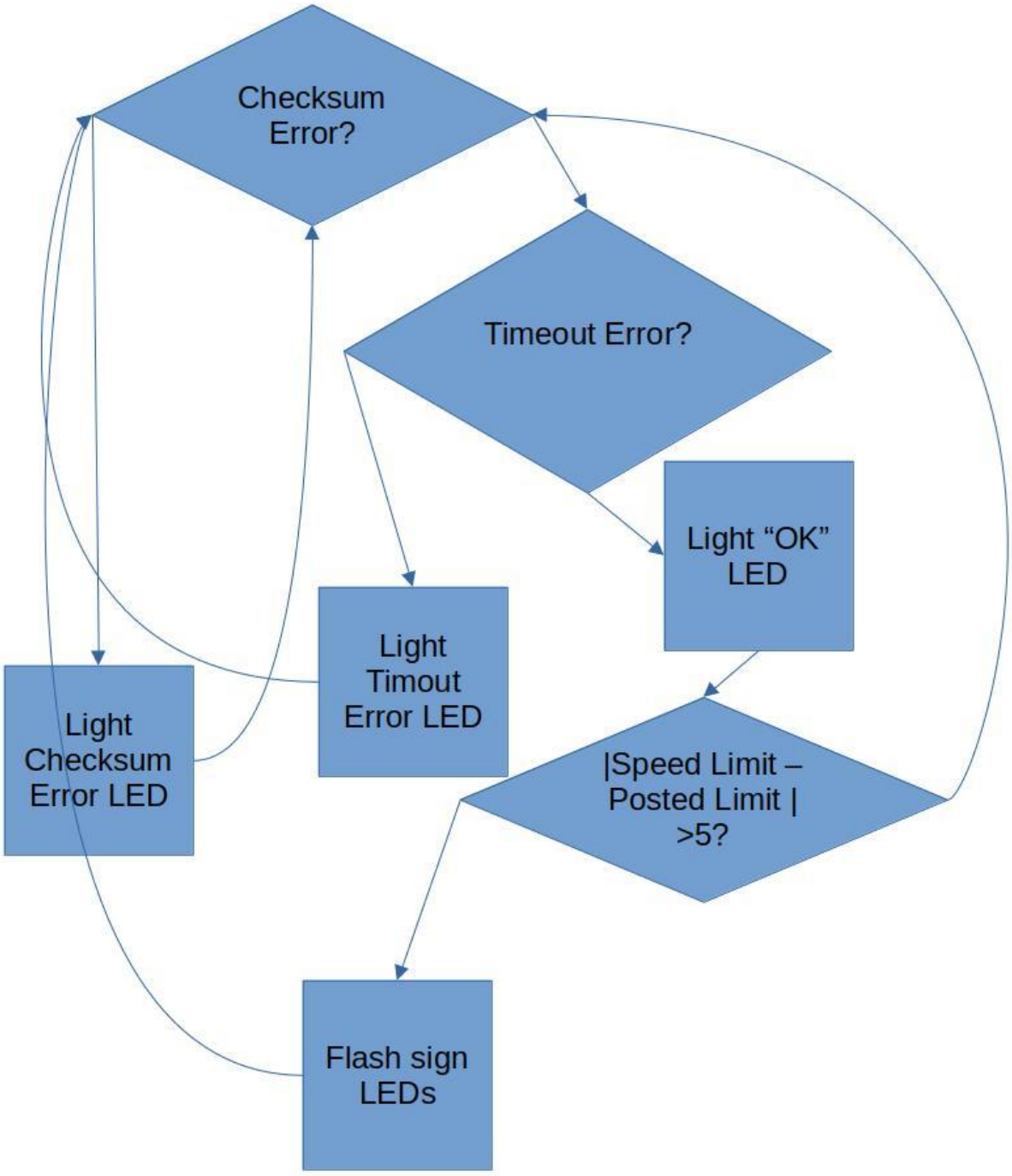
Block Diagram



Functionality Diagram



Status Thread Detail

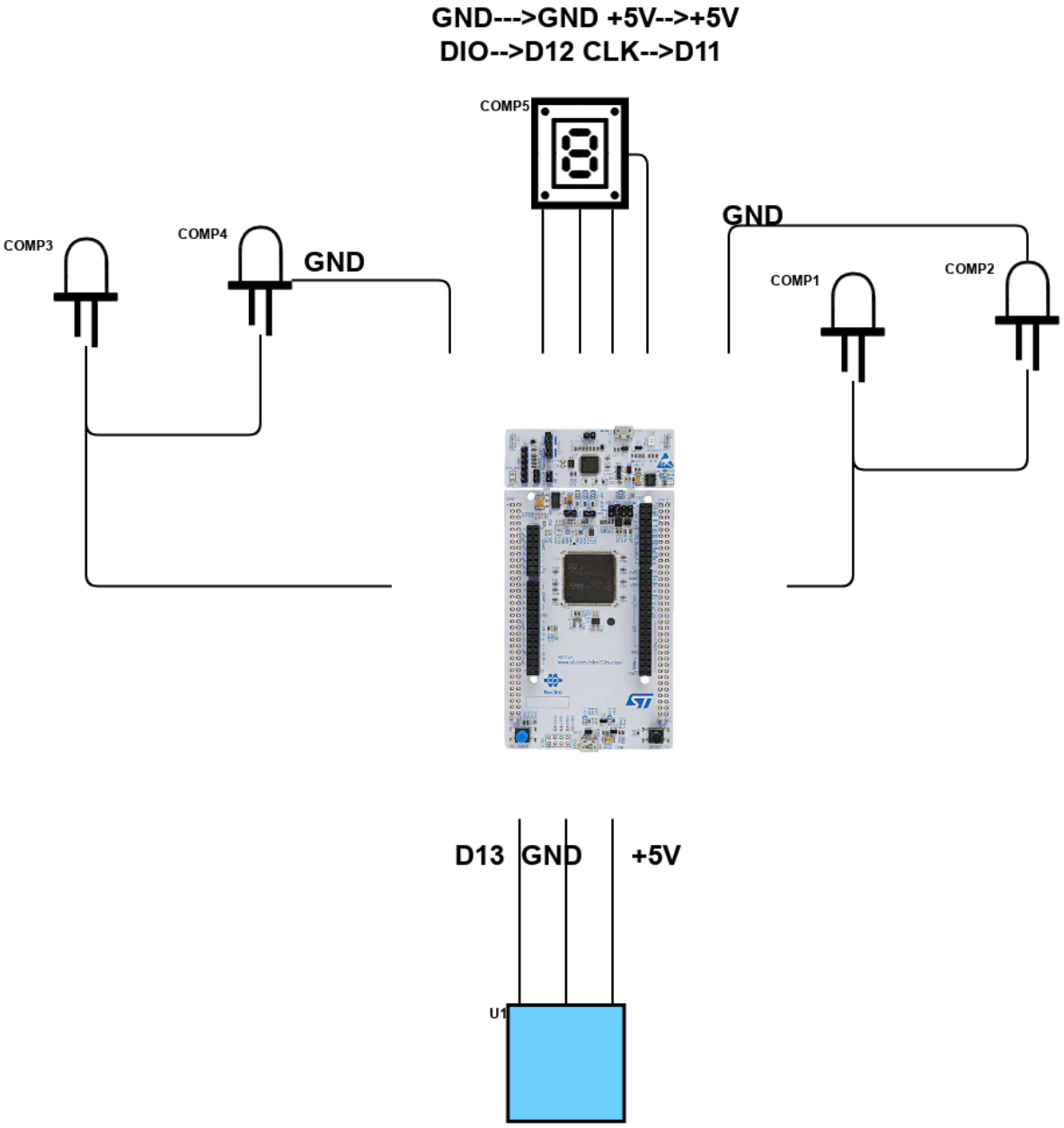


Bill Of Materials

- Two red LEDs
- Two white LEDs
- 7-segment display
- Breadboard
- DHT11
- Nucleo L4R5ZI
- Jumper wires

Instructions

Schematic



Building The System

1. Connect the Ground (blue) rail of the breadboard to the pin on the Nucleo marked “GND”
2. Connect the 5V (red) rail of the breadboard to the pin on the Nucleo marked “5V”
3. Put two red LEDs next to each other on the breadboard, ensuring that the long lead on one is in the same row as the short lead on another
4. Connect the first end of the LED chain to the ground rail, and the other to D15
5. Do the same for the white LEDs
6. Connect the first end of the LED chain to the ground rail, and the other to D14
7. Connect the 7-segment display to the Nucleo
    a. The Ground pin on the display is connected to the GND pin on the nucleo
    b. The 5V pin is connected to the 5 volt rail on the breadboard
    c. The CLK pin is connected to D11
    d. The DIO pin is connected to D12
8. Connect the DHT11 to the Nucleo
    a. Connect the power pin to the 5V rail on the breadboard
    b. Connect the ground pin to GND
    c. Connect the DIO pin to D13

Using The System

This system simply needs to be placed in the desired location for the signage, and once powered, will run and display the calculated speed limit indefinitely. Ensure that the display is in a location where it is readable, and the humidity and temperature sensor is off the ground and protected from rain.

Test Plan

Let the sign run and ensure that the green LED remains lit, and the speed limit is displayed properly. In testing, the speed limit stayed at the posted limit, and the checksum and timeout error LEDs were intermittently lit.

Future Considerations

The current iteration of the system cannot actually detect rain, only that the current temperature is at or near the dew point. It also cannot detect icing conditions, or fog, or any other weather that may impact

sight distance. In the future, additional sensors such as moisture sensors could be added to determine road conditions more accurately. In addition, the DHT11 used seemed to be unreliable, and a different manufacturer may be used. Also, the current method of calculating the dew point [2] is only accurate above fifty percent relative humidity, so a pressure sensor could be included to determine dew point at low humidity.

References

1. D. Driving, “Driving in wet conditions,” Defensive Driving, 03-Aug-2017. [Online]. Available: https://www.defensivedriving.com/blog/driving-in-wet-road-conditions/. [Accessed: 10-Dec-2021].

2. M. G. Lawrence, “The relationship between relative humidity and the dewpoint temperature in moist air: A simple conversion and applications,” Bulletin of the American Meteorological Society, vol. 86, no. 2, pp. 225–234, 2005