ELEC 327 Final Project Report - Team 19

Ethan Carr / Ishan Lokhandwala

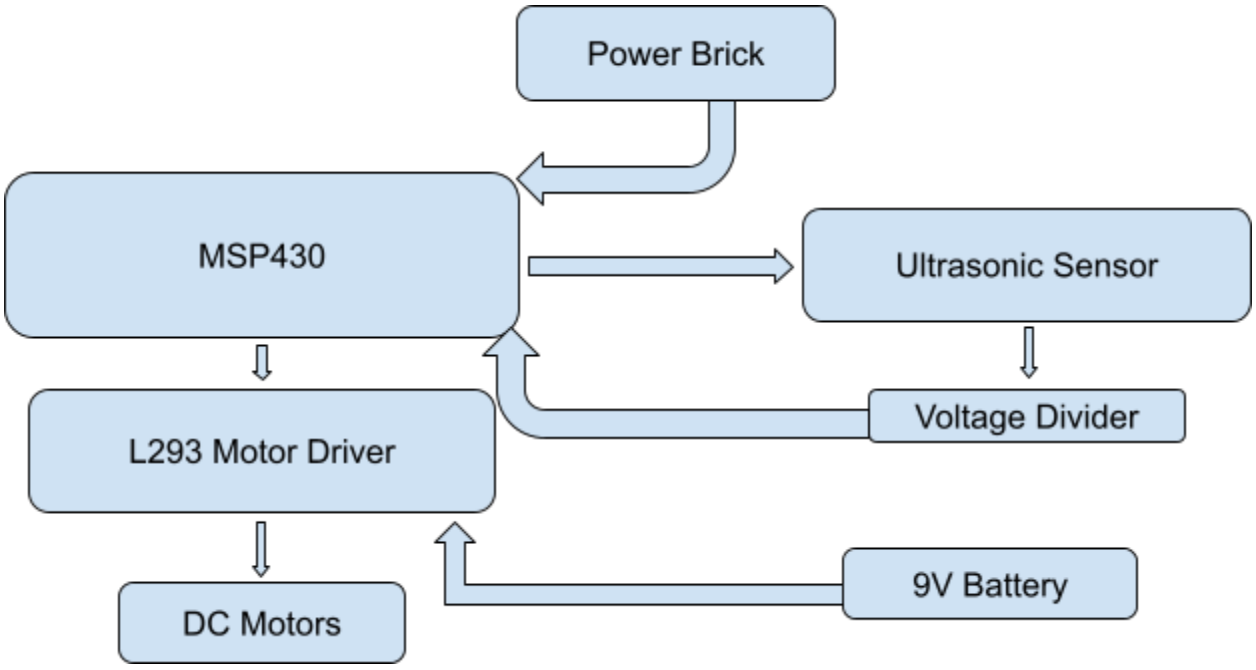
Spring 2024

Overview - This project is an implementation of a moving vehicle with obstacle avoidance. It uses a TI MSP430, a TI L293 motor driver, an ultrasonic sensor, and wheel motors. The vehicle works by sensing what is in front of itself, keeping track of the distance to the nearest obstacle. If the distance closes into a range of 15 cm or less, the vehicle will send a signal to the motors to stop to avoid collision. Upon moving the vehicle or obstacle such that the distance is now greater than 15 cm, the vehicle will send a signal back to the motors to resume its motion going forward. The overall code and implementation is modular with specific functions for analyzing and controlling the vehicle, allowing for a versatile platform for expansion if needed.

Code - The code has many different components. The beginning of the file has port, parameter, and other important definitions for the vehicle. One of the first functions in the code is PWM, which initializes the output directions for the MSP, specifically for motor control. It consists of an enable signal for the L293 on P1.2, and two signals for motor one, each at P1.0 and P1.1. There is code for an additional motor left over, but for our design, it was not needed. The next function, setMotorDirection, controls the motors and direction of spin. Initially, this function was used for multiple motor inputs, but as mentioned earlier, the use for multiple motors was not needed, since 2 motors could be driven from one set of signals. Within the function there is a statement to determine the direction of spin, which determines which out pin will be enabled or not. Next is the main function of the code. It initially defines some more constants to set up the vehicle. Important things to note are the pausing of the WDT, the initialization for button usage on startup, and configurations for the serial output and timer modules. Additionally, this where PWM() is called to initialize the values, as well as the interrupt enable for changing the execution path. Then the code enters a while loop, in which it calculates the distance measured by the sensor and prepares serial output. The distance is calculated by measuring the time difference between the trigger and echo pulses of the sensor, and dividing that value by the speed of sound, setting the distance in cm. The distance is then formatted to a char and read by the UART buffer. Finally, the end of the

file defines the two interrupts needed for controlling the vehicle, as well as a small serial output helper function. The two interrupts are the Timer_A1 interrupt, and the Port_1 interrupt, each having different use cases. The timer interrupt serves as a checkpoint for the MSP to check the distance read in by the ultrasonic sensor, with some predefined logic within the interrupt determining the next course of action. If the distance read is less than or equal to 15 cm or the button flag is disabled, the MSP will call setMotorDirection and switch off the motors, stopping the vehicle. If the above if statement does not trigger, then the else if block will check if the button flag is enabled, to which it will turn on the motors. The button flag acts as a “start” switch for the vehicle, and uses the Port_1 interrupt to activate the flag using P1.3.

Hardware - The vehicle uses very minimal hardware in its overall design. It is composed of a TI MSP430, a TI L293 motor driver, an HC-SR04 ultrasonic sensor, DC motor driven wheels, and various wires/resistors to connect everything together. A diagram of the set up is shown below:



ELEC 327 Final Project Report - Team 19

Ethan Carr / Ishan Lokhandwala

Spring 2024

The hardware breakdown is as follows: the code is flashed to the MSP430 via a laptop, and is then powered externally by a portable charger. The ultrasonic sensor's GND, VCC(5V), and trigger(P1.6) pins are all connected directly to the MSP430, but the echo pin is connected via a voltage divider, as it delivers a 5V signal, but the MSP430 can only handle up to 3.3V. The remaining voltage is diverted to GND. The MSP also drives the OUT(OUT3,4 to P1.0, P1.1) and EN(EN3,4 to P1.2) signals to the L293 motor driver, which control the motor state and output direction. It also serves as the GND for the motor driver. Power for the two DC motors is provided by an external 9V battery, which is connected to the VCC1 and GND pins of the motor driver.