Parking-Maid

Project Design Plan

Ruoqing Fu (rf2475), Po-Yen Chou (pc2624), Hao-Yu Hsieh (hh2539)

Milestone1:
To make motors, ultrasonic sensors and temperature sensor work individually.

Milestone2:
- To set up our own FPGA environment that performs the same as that used in lab
- To piece together all the modules and platforms.
- To establish the communication between hardware and software

Milestone3:
- To implement distance sensing techniques to improve the accuracy of distance calculation
- To implement the automatic parking algorithm

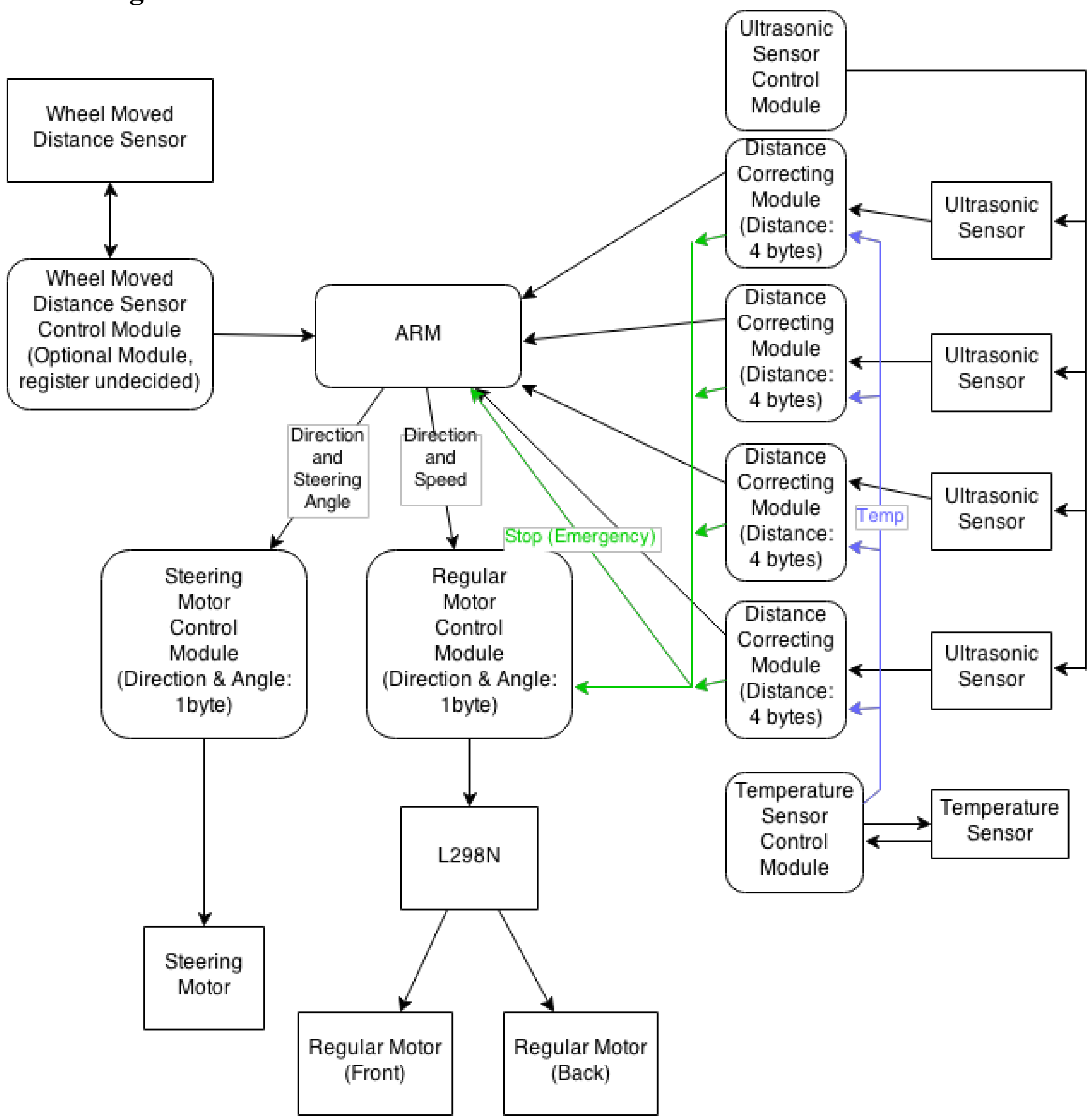
Current Progress:
- Successfully built the LEGO chassis
- Successfully verified the functionality of ultrasonic system
- Successfully verified the functionality of regular and steering motors.
- Successfully combined all above modules into one single module and can be controlled by switches and buttons on FPGA.

Module List and Descriptions:

Design Figure:

In our design, we use 4 ultrasonic sensors to detect the distance, 2 regular motors and 1 steering motor to control the movement of the car.

Block Diagrams:



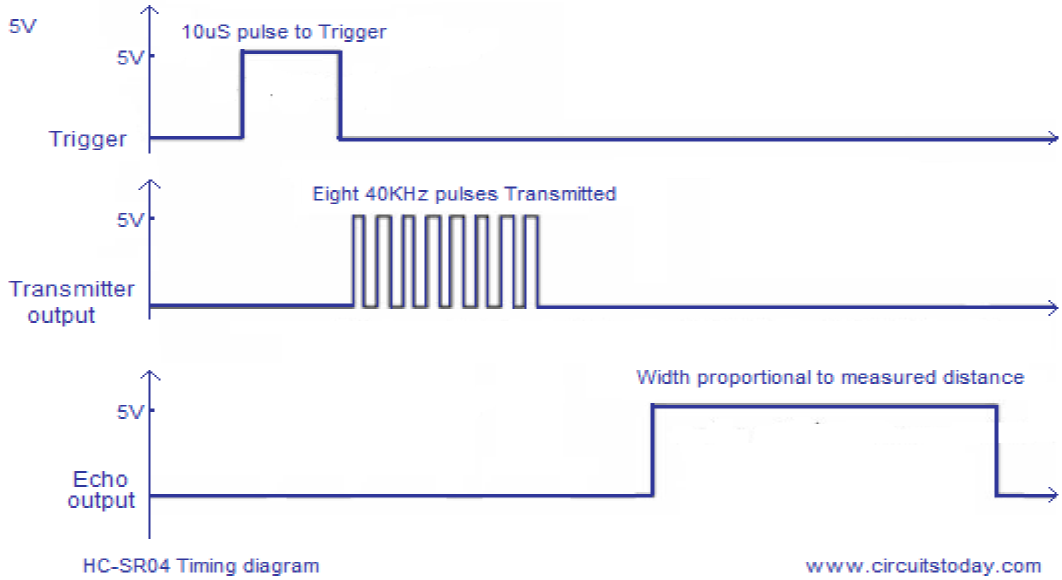
In general, we use ARM/Linux as our control center. It's used to as the following:
1. Collect data from peripherals
2. Use car speed to correct distance received from Distance Correcting Module(DCM), since the speed of a moving car will affect the correctness of the distance. (DCM only correct distance using temperature.)
3. Compute the parking algorithm
4.. Control the peripherals, such as motors and ultrasonic sensor configuration.

Each peripheral is described as follows:

Ultrasonic Sensor Control Module & 4 Ultrasonic Sensors:
They are planned to be installed on the car to obtain the dynamic distance between the parking

space and the car. Our parking tool is mainly focusing on parallel parking on the right. So two of them are placed on the right side of the car and the other two are placed on the front and back respectively. (If we have more time, we may also implement parking space parking, which may need up to 8 ultrasonic modules)

Ultrasonic sensor timing diagram:



For Hc-sr04 Ultrasonic sensor module, we only need to input a short 10 µS signal to the trigger pin, then the module will send out eight 40KHz pulse sound and receive the echo signal. From the echo pin, we will then receive a signal, of which the high level time is the time interval between sending and receiving the ultrasound. So we can calculate the distance from the formula: distance = (high level time)* (sound speed) / 2

Temperature Sensor Module:
The temperature sensor will work with ultrasonic modules to produce an accurate distance output because ultrasonic modules vary by temperature.

Temperature Sensor:

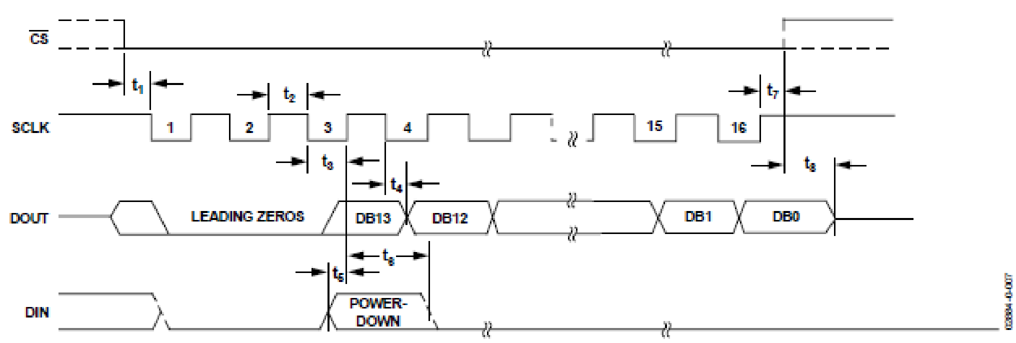


Figure 14. Serial Interface Timing Diagram

Temperature sensor will work with ultrasonic module to achieve more accurate distance calculation. The sensor is communicated through SPI. The ADC built in the sensor will produce a 14-bit output to represent the external temperature. CS is an active-low signal to control the start of temperature measurement. When CS is pulled low, Dout will become effective after two clock cycles. DIN is an optional signal and will only be used to write the part in standby mode. As shown in the timing diagram, CS has to arrives at least 5ns early before SCLK and leaves 5ns after DOUT completes to meet clock setup time and hold time requirement. For our project, we will enlarge the setup and hold time from 5ns to 25ns and trigger the CS at the closest positive edge. Currently we use the temperature sensor built in FPGA, but may use an external temperature sensor in the further, since the one on FPGA might be affected by the heat generated by FPGA.

Distance Correcting Module:
Because the accuracy of ultrasonic may be affected by different temperatures, we use this module to correct the distance obtained from ultrasonic sensors.

Regular Motor Control Modules & Steering Motor Control Module:
Regular motor is used to move forward and backward; Steering motor is used to make turns.

L298N:
A hardware that can convert PWM signals generated by FPGA to stable DC voltage that can control the regular motors. We use this hardware because we need different DC voltage levels to control motor speed.

(Optional) Wheel Moved Distance Sensing Module:
The mechanism is to take the consideration of other factors that may produce distance calculation error, such as car sliding in bad weather or helping ultrasonic sensors calculate the distance. We haven’t decided how to implement this module. It may be unnecessary to implement if we find a better way to reduce the calculation error.

Emergency Stop From Ultrasonic Sensors to Regular Motor Control Module:
This path enables the car to brake for emergency and directly connects to Motors without going through the arm processors, which could be delayed. In usual cases, motors should only be controlled by ARM/Linux.