Truck Loading Using an Autonomous End-Loader

Functional Description and Complete System Block Diagram

By:

Kevin Hurley and Ryan Leman

Advisor:

Dr. Schertz

October 23, 2007

I. Introduction

The autonomous truck loading project will utilize an end-loader to locate and navigate to a load, scoop the load, and proceed to locate and navigate to a truck, at which point the end-loader will empty the bucket into the truck. This process will continue until it is deemed that the truck is filled. We will utilize encoders and infrared beacons to determine end-loader position and motor drive requirements to achieve the desired destination. A Silicon Labs development board will be used for overall system control. The final goal of the project will be to have a functional autonomous system fill the truck as quickly and accurately as possible, while maintaining a low system cost.

II. System Description

The system will consist of encoders or another form of velocity sensor, position sensor, likely infrared beacons, a Silicon Labs development board for system control, and the end-loader controlled via the microcontroller and a motor for both tracks and an additional motor for the bucket. There will be one encoder mounted on each of the two tracks and one infrared beacon on the truck and another at the load. The Silicon Labs development board will be mounted on the end-loader with a power source also mounted on the end-loader.

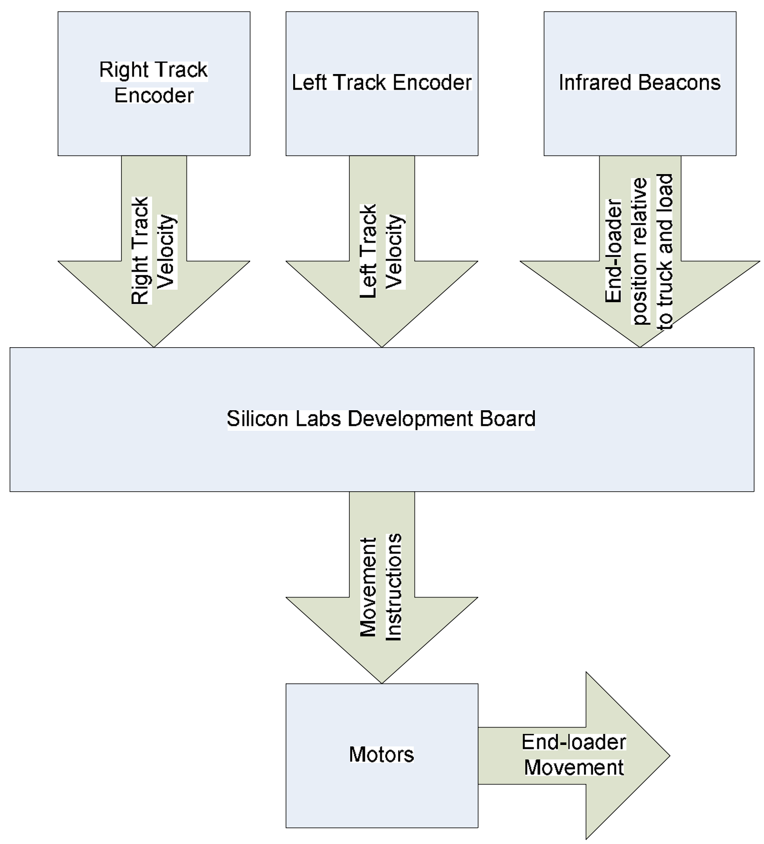

Figure 1: System Block Diagram

Subsystems

As figure 1 illustrates, there are four main subsystems. The subsystems are as follows: the encoders, the infrared beacons, the development board, and the motors.

1. Encoders

The encoders will be used as a form of velocity and position determination. They will allow the end-loader to determine how far the end-loader has moved based on the previous PWM in order to determine if a higher or lower PWM will be needed during the next timing cycle.

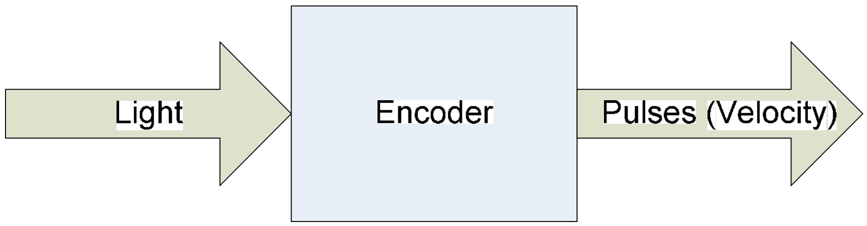

Figure 2: Encoder Subsystem Block Diagram

2. Infrared Beacons

The infrared beacons will be used to determine the locations of the end-loader relative to the truck and the load. This information, in conjunction with the encoder data, will allow the system to determine the PWM signal to be sent to the appropriate motors.

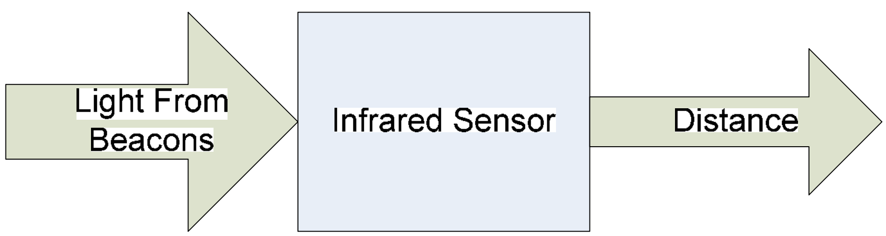

Figure 3: Infrared Subsystem Block Diagram

3. Development Board

At present, we intend to use a Silicon Labs development board to provide vehicle control, but will further investigate this to ensure that it will be adequate. The board will use the inputs from the infrared beacons and the encoders to provide motor control and to determine the appropriate operation to be undertaken. The development board will navigate to the load or truck and then operate the bucket when needed.

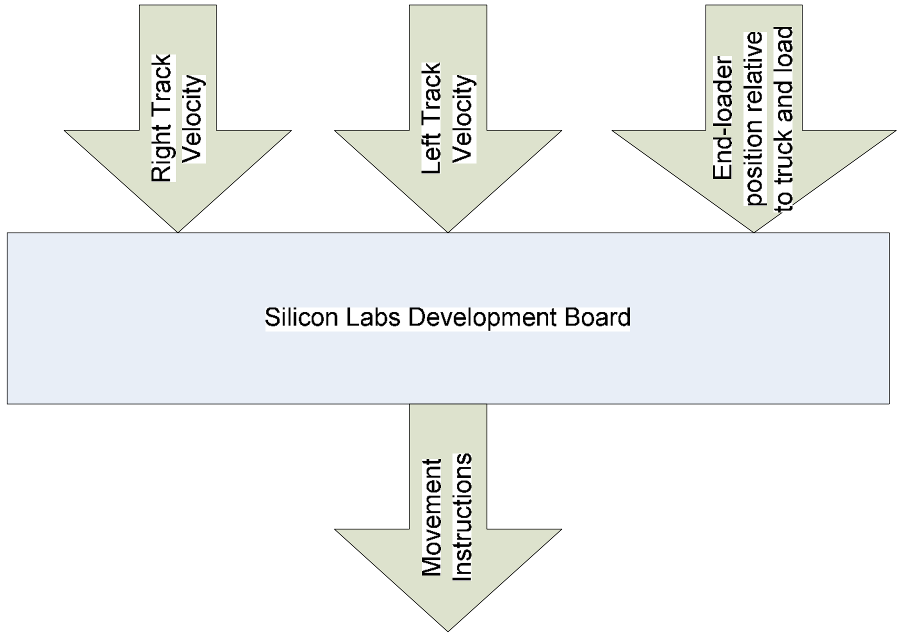

Figure 4: Development Board Subsystem Block Diagram

4. Motors

The motors will provide end-loader movement for navigating to the load and truck. There will be a separate motor to control the right and left tracks in order to provide turning. Both motors will also be able to operate in forward or reverse, independent of the other motor, giving us a smaller turning radius as well as the ability to drive in either the forward or reverse directions.

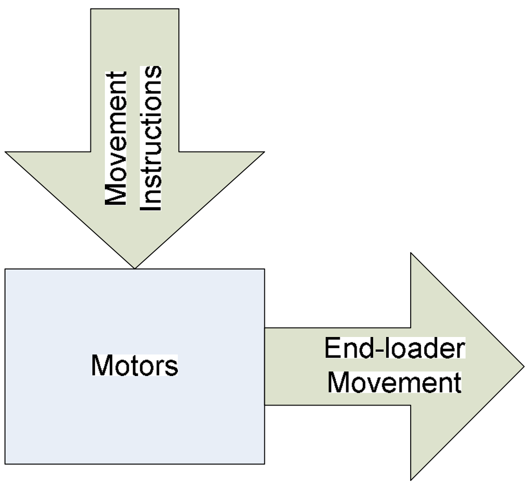

Figure 5: Motor Subsystem Block Diagram

Software Implementation

The overall software flowchart appears in Figure A-1. After setting up the interrupts and performing required initialization, the system will wait until our main timer has overflowed and set a flag indicating that the main loop should proceed. This will be done to ensure that control operations are handled at a constant period in order to provide sufficient data from our velocity sensor to determine the PWM drive required to each drive motor to move towards the load or truck.

After the main loop begins, the software will check to determine if the truck is full, likely to be defined as a certain number of times we have emptied a load into it. If the truck is deemed full, we will stop the end-loader and wait until a push button is pressed to indicate that the truck has been emptied and the end-loader can continue loading the truck.

If the truck is not full, the software will check the end-loader's position relative to the load and truck. It will also check how far the end-loader has moved based on the previous PWM in order to determine if a higher or lower PWM will be needed during the next timing cycle. If the bucket on the end-loader has already been filled with a load, the end-loader will need to navigate to the truck and empty the bucket into the truck once it arrives. If the bucket on the end-loader is not full, the end-loader will navigate towards the load and scoop up a load once it arrives there.

Development Procedure

After determining necessary hardware, we will begin vehicle construction and do limited testing to ensure functionality. Initially, we intend to construct a preliminary end-loader in order to determine if the current kit will be adequate, and if any additional hardware will be required. It will then be required to determine which sensors will provide necessary features and accuracy to maintain the lowest possible cost while still operating as needed. At this stage, we will begin implementing the control system via the microcontroller, using modular programming practices for easier software integration and debugging. The ultimate goal of the project will be to have a functional autonomous system to accurately scoop a load into the end-loader and fill the truck.

Appendix A

Figure A-1: Overall Software Flowchart