Final Project

Electrical Engineering 474 Lab 5
University of Washington

Name	Student ID
Denis Jivaikin	1432129
Novin Changizi	1434956
Ting-Yu (Jacky) Wang	1333301

INTRODUCTION

This lab focuses on the development of a vehicle that utilizes both the Bluetooth chip as well as startup scripts to allow the BeagleBone to automatically execute vehicle firmware. In addition to the core project, the group has decided to implement lane detection feature for the vehicle which enables the vehicle to drive within a given lane.

HARDWARE OVERVIEW

The setup is similar to that of lab 4 (Bluetooth, motors and H-bridges). In addition, a camera and a Raspberry Pi controller is used to perform lane detection. The result from the lane detection is transferred from the Pi to the BeagleBone via UART. The hardware interconnection is as follows:

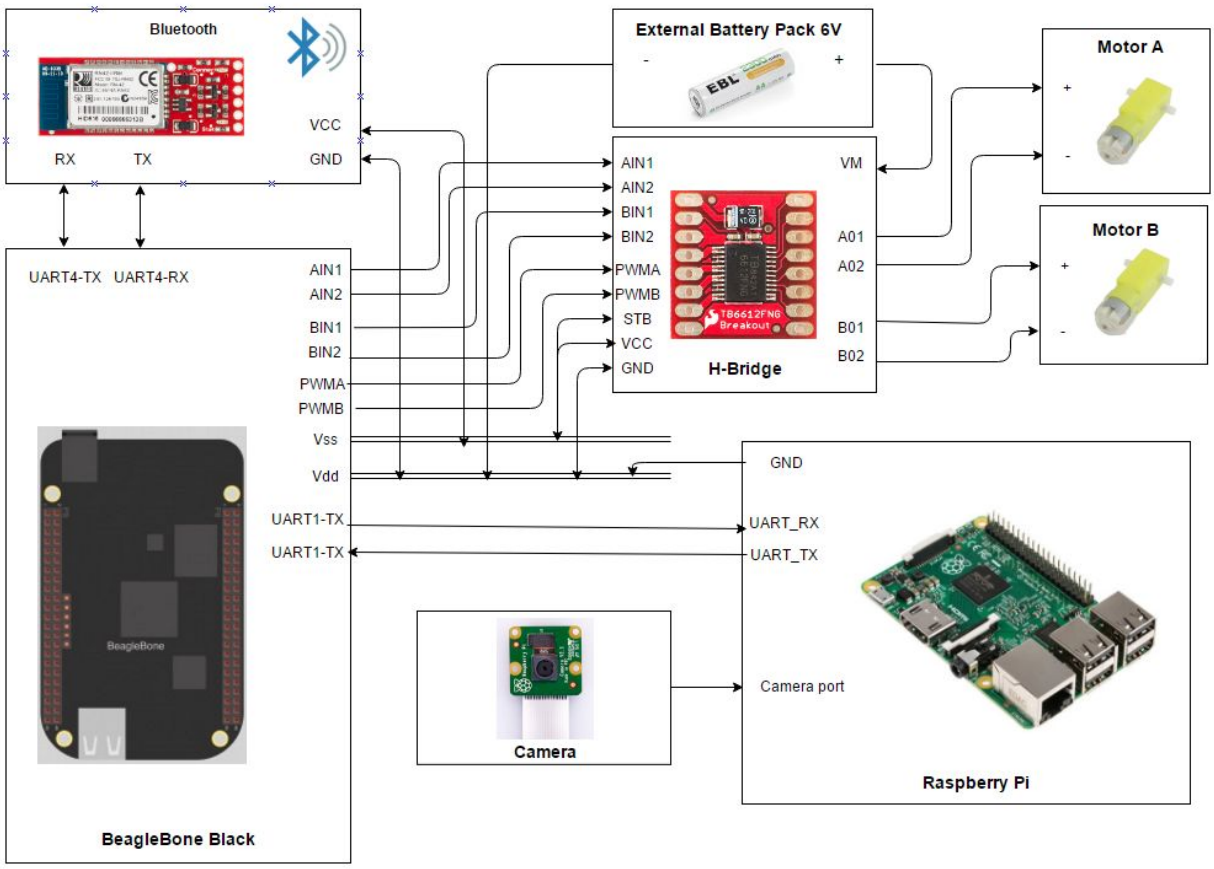

Figure 1: Hardware interconnection block diagram

SOFTWARE OVERVIEW

To properly control the tank, two executables were developed *motor* and *uart_drive*.

motor:

- Exports gpio and pwm pins and changes the directions with previously used functions: export_gpio_pins, export_pwm_pins, and change_dir. Duty cycles and frequencies of pwm pins are also configured
- Exports and samples ADC pins (proximity sensors)
- Creates a fifo that allows communication between the tank and the remote controller
- Remote Control Mode
    - In a while(1) loop constantly reads the fifo for user input. If the user specifies an action, prints it to the screen and calls a corresponding function: forward(...), backward(...), left(...), right(...), stop(...). In the action functions input pins of the H-bridges are toggled in different combinations to get the tank to move in desired directions
- Line Following Mode
    - In this mode our robot uses the Raspberry Pi and the OpenCV library to process the Raspberry Pi Camera's input (the camera is mounted in front of the robot at a 45 degree angle) and finds the line that it is trying to follow if it is in the frame. Then it makes a decision about what the robot should do (forward, left, or right) in order to follow that line (or stop if there is no lines in frame), and sends that decision over UART to the BeagleBone to dispatch to motor
    - The line detection in line.c does the following in a continuous loop:
        - Capture camera input
        - Perform Gaussian Blur to smoothen edges
        - Perform Canny edge detection that finds edges by looking for neighboring pixels with high pixel value difference. Produces a binary image where white represents edge, black represents non-edge
        - Perform Hough line detection on the result. This looks for linear contiguous pixels of the same color (in this case white edge pixels)
        - Averages the resulting lines
        - Finds the horizontal distance of this resulting line to the center of the image, uses the slope and some math
        - If this line was to the right of the center past a threshold variable, then correct the tank position by dispatching right command
        - Else if it was to the left past that threshold dispatch left
        - Otherwise keep going straight since no correction was necessary
        - If no line was found in the frame then simply dispatch stop
        - (Could improve this by taking line angle into account as well)

uart_drive:

- Calls a python script UART_SETUP.py to setup UART4 and UART1 for use on BeagleBone
- Opens the devices (UART4) and modifies the settings of serial communication such as: baud rate, number of bits transferred, delays, control flow, and parity bit choice
- Writes the initialization string to Bluetooth (UART4) to enter the configuration mode and change the name of the device
- Uses while(1) loop to constantly read data from UART and write it to the pipe created by *motor* for communicating with Bluetooth.

uart_drive_pi:

- Opens the devices (UART1) and modifies the settings of serial communication such as: baud rate, number of bits transferred, delays, control flow, and parity bit choice
- Uses while(1) loop to constantly read data from UART1 and write it to the pipe created by *motor* for communicating with Pi.

To test the Bluetooth (UART4) communication an Arduino terminal app on an android phone was used to send commands through. In addition, a startup bash script was written to invoke both the aforementioned executables whenever the BeagleBone is powered up. This effectively allows the tank to be operating without being plugged into a computer.