CS437: Internet of Things
Lab 1 Part 1

Name: Michael Edukonis
NETID: meduk2
Late Days Used: 0
Video Link: https://youtu.be/SyrBTmAsbu4

Design Considerations
January 26, 2023

Assembly:
All parts assembled according to lab specifications and kit instructions. Some feedback was noted from other customers and students regarding less than desirable run time from the standard set of two 3.7v 18650 batteries. Multimeter test shows that the battery holder for the car is wired in series giving a nominal output from two batteries of 7.4v @ 5000 mAh at a weight of 75 grams. This configuration provided approximately 5-6 minutes of run time before reduction in power is noted on the picar-4wd hat LED output.

The Raspberry Pi 4b is configured on the bottom with the picar-4wd hat on top. This reduces airflow to the raspberry pi board with some noticeable heat coming from the assembly, but no adverse effect was noted while running python scripts for this lab. Picar-4wd code includes functions for returning the current cpu/gpu temps which remained well below the max 80c as documented in raspberry pi documentation (Raspberry Pi Ltd).

Due to an abundance of caution partly due to the elevated price of raspberry pi boards, heatsinks were installed which helped to further reduce operating temperatures. During handling of the assembly, the picar-4wd hat underside shorted itself to the main heatsink causing a failure to a 3.3v voltage regulator. The surface mount regulator was replaced, and testing resumed with some electrical tape applied to the picar-4wd hat underside to insulate it.

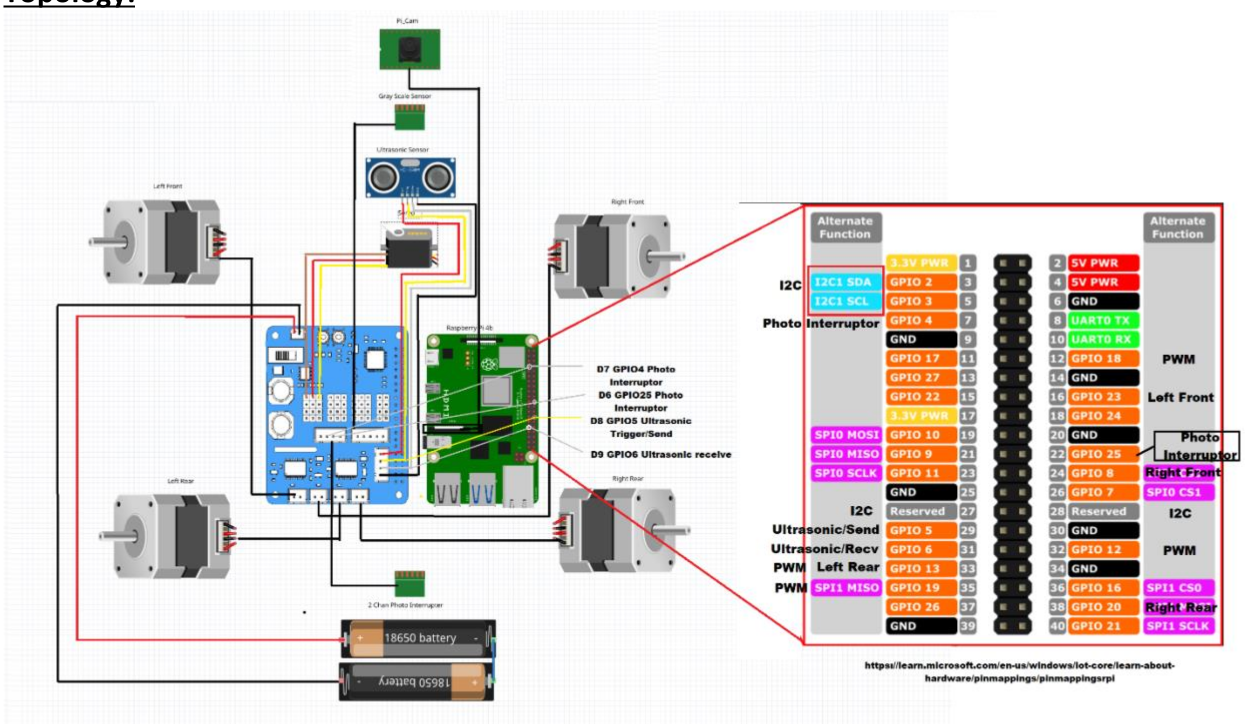
The pi camera was mounted to the front most bumper of the car with plastic m2.5 standoffs and screws. The 6-inch cable was not sufficient to connect the pi cam to the raspberry pi board so a 12-inch length cable was utilized successfully. Pi-camera will be utilized in part 2 of this lab.

Wiring:
Through the picar-4wd hat, various devices are wired to the general purpose I/O pins of the raspberry pi. Topology attached.

Pin 3	GPIO2	I2C
Pin 5	GPIO3	I2C

Pin 7	GPIO4	Photo Interrupter
Pin 12	GPIO18	Hardware PWM
Pin 16	GPIO23	Left Front Motor
Pin 22	GPIO25	Photo Interrupter
Pin 24	GPIO8	Right Front Motor
Pin 27	Reserved	Secondary I2C
Pin 28	Reserved	Secondary I2C
Pin 29	GPIO5	Ultrasonic Trigger/Send
Pin 31	GPIO6	Ultrasonic Recv
Pin 32	GPIO12	Hardware PWM
Pin 33	GPIO33	Left Rear Motor (also hardware PWM)
Pin 35	GPIO19	Hardware PWM
Pin 38	GPIO20	Right Rear Motor

Topology:



Software:

The raspberry pi 4b utilizes a 32gb microSD card with the latest raspberry pi OS (bullseye rel: 9/22/2022). wpa_supplicant and ssh files were added to activate wireless connection and remote capability upon boot. Once booted and logged in, update/upgrade was run, smb server was also installed and configured for ease of moving files back and forth across the network.

Picar-4wd code was reviewed, installed, and examples were tested as is to verify that all assembled components were working properly. Tests were successful. Despite lack of comments, analysis of picar-4wd code revealed lower level functions that would be useful for writing friendlier, user defined functions to operate the car.

The goal of this lab is to write a program in python, utilizing the picar-4wd software as a starting point, that uses the ultrasonic sensor to detect obstacles that come within several centimeters of your car's front bumper. When the car gets close enough to an obstacle to meet the criteria of software, the car will stop, choose another random direction, back up and turn, and then move forward in the new direction.

The code 1st instructs the ultrasonic sensor to scan it's surroundings at 9 intervals across a 180 degree range: 90, 60, 45, 20, 0, -20, -45, -60, -90 degrees. The degree intervals are achieved through movement of servo to which the ultrasonic sensor is attached. The ultrasonic sensor takes a measurement at each stop. These instructions and the results are handled in the scan() function.

A list of measurements to surrounding objects (if any) is generated. Measurements in centimeters is returned. If nothing is within range, a -2 is returned. The software takes these measurements and converts them in the same list to a "1" for object near or a "0" meaning nothing nearby. This gives the software a very rough "visual" of it's surroundings. The next instructions will slice the list into areas (left, center, right) and decide if anything is in the way. There is a slight deviation from the lab spec for object detection. Turning is based on three scenarios:

1. If an object is detected to the left, the car should turn away from the object to the right.
2. If an object is detected to the right, the car should turn away from the object to the left.
3. If an object is detected head on, the car will adjust it's position as specified in the lab requirements. It will backup, pick a random direction, execute a turn to that direction, and then move forward in the new direction after re-scanning it's environment.

Citations:

Picar-4wd software
SunFounder, picar-4wd, (2020), GitHub repository, https://github.com/sunfounder/picar-4wd

Raspberry Pi Ltd. (2012, January). Raspberry pi documentation. Raspberry Pi Configuration. Retrieved January 23, 2023, from https://www.raspberrypi.com/documentation/computers/configuration.html #over-temperature-warning-80-85c

Contributors:

No other contributors on this report.

Code and documents for this lab: https://github.com/medukonis/picar