Lab 7 Formula0001 Racing Robot

The robot materials were funded in part by a grant from Tivioli.

Goals
- Design a robot that can move forward/backward, and turn left/right,
- Interface motors, and sensors to two microcontrollers,
- Implement pulse-width modulation using output compare to adjust power to the motors
- Employ input capture interrupts to measure distance,
- Write low-level device drivers for the motors and sensors,
- Develop a high-level control system,
- Use communication skills to work effectively as team.

Review
- Text book Chapters 8 on interfacing motors and sensors
- Lectures 9-12 on input capture, PWM, motors and control
- Textbook Chapter 10 on control systems,
- Construction guide for the erector set,
- Data sheets for L293, GP2Y0A21YK, HC-SR04, and Ping))).

Starter files
- See Lab6.sch

To win this competition, your team must combine mechanical, electrical and computer skills. Each robot has one switch (in addition to the reset switch) that the students will push to start the competition when instructed by the TA-referee. The **reset** switch (the normal hardware reset) should cause all motors to stop and will be used for safety purposes. You will push your **start** switch once at the start of the competition signifying the start of the 180-second competition. You are allowed **up to three** repair events on your robot during the 180-second competition. Otherwise, the robot must run autonomously. There can be no wireless input commands or buttons pushed during the competition. Both microcontrollers must run using your RTOS with blocking semaphores developed in previous labs. It must also employ digital control in an appropriate manner. Examples of appropriate control are travelling at a constant speed, travelling at a constant angle to the wall, and travelling at a constant distance to the wall. Example control algorithms include incremental, proportional-integral and fuzzy logic. If you are not sure whether or not robot satisfies this requirement ask you TA.

The Track

There are seven example tracks posted on the class web page. You can view these tracks by looking at the pdf versions, and you can create your own track by editing the PCBartist PCB files. Figure 7.1 shows one of the tracks, which is made with 54 pieces of wall. The *wall* pieces are 32 to 33 inches long, made from cedar wood about 3.5 inches tall and 3.5 inches wide. The width of track including straight portions and turns will vary from 29 to 35 inches. Each lap will be divided into 5 to 10 segments, demarked by the milestones and signified with blue tape on the walls. Robots will be started in pairs, with a wall either on the left or on the right.

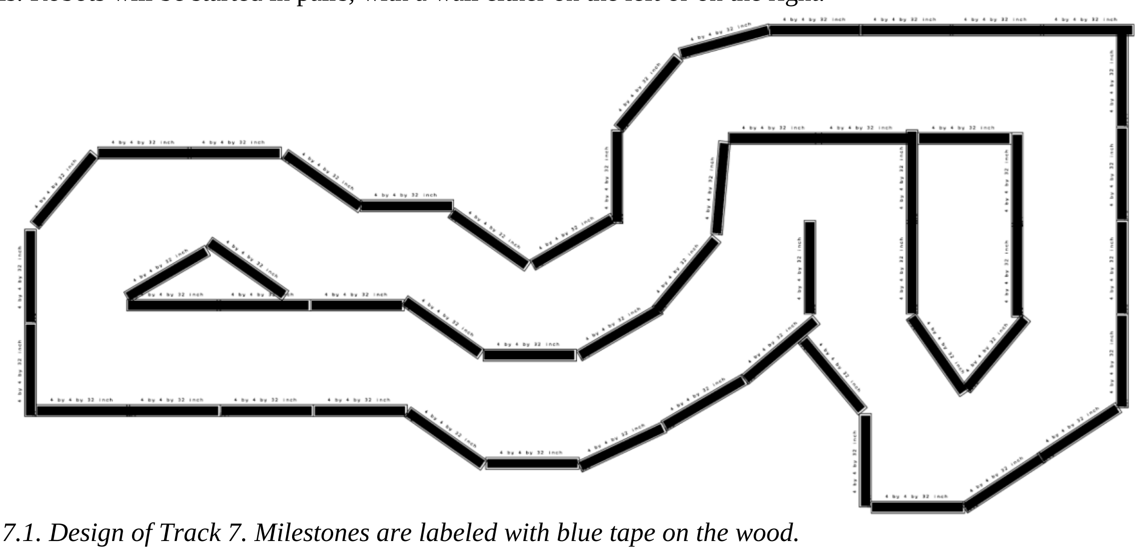

Figure 7.1. Design of Track 7. Milestones are labeled with blue tape on the wood.

The Race

One, two, three, or four robots will race at the same time. Your score will be based on both speed and accuracy. Your starting location will not be known until race time. Two robots will begin side by side at a milestone as chosen by the TA-referee. The TA-referee will tell you which milestone to start up and which direction to travel.

The students will have 60 seconds to place their machines onto the track from the time the TA-referee calls them to set up. The front-most part of your robot must be at the milestone line at the start, but its lateral position and orientation are up to you as long as you are on your half of the track. When the teams are ready, the TA-referee says, "Go", and each team will push its **start** button to activate its software. If a robot fails to start as expected, the TA-referee will award one *false start*, and the race will be restarted. Two *false starts* constitute a disqualification. The remaining robots will be allowed to continue. The powered portion of a competition will last 180 seconds. Your robot must stop all moving functions at the end of 180 seconds, and any robot that continues to move after 180 seconds will lose two points.

One member of the team will be designated as the mechanic. This member will be allowed to make **up to 3** repairs during the race. If your robot needs a fourth repair it will be disqualified and not allowed to continue. Examples of repairs include but are not limited to: 1) reorienting the robot because it is going the wrong way; 2) moving a robot because it is stuck up against a wall; 3) reconnecting a loose wire; 4) fixing a mechanical malfunction; and 5) resetting the software. Repairs must be made in such as way as to not impede the progress of the other robots. Because of the speed limit in pit row, all repairs incur at minimum of a 15 second penalty. A team member shall count down 15 seconds out loud during the repair. After the repair, the robot must be placed back on the track at a position behind where it was picked up and not within 2 feet of another robot.

Each milestone is signified by tape on both sides of the track, creating a virtual line across the track. You will start with the front of your robot at one of these virtual lines. Let **M** be the number of milestones passed, and **S** the stop penalty. **S** will be **0** if your robot stops after 180 seconds. However **S** will be -2 if your robot does not stop after 180 seconds, and **S** will be -2 if your robot is disqualified. Your score is calculated as

Score = **M + S**

For example, if your robot passes 7 milestones but does not stop moving after 180 seconds, then your score will be 7-2=5. To achieve a milestone, any part of your Frisbee disk must cross the milestone line. *You can achieve a milestone only once per lap.* "Going the wrong way" is defined as a robot making significant progress in the wrong direction. If the mechanic thinks your robot is "Going the wrong way", he or she should pick it up, **wait 15 seconds,** rotate it into the proper direction, put it down at least 2 feet from other robots.

The TA-referee will disqualify a robot if it does not appear to be making progress or if the TA-referee thinks you are using repairs inappropriately. The TA-referee will disqualify a robot immediately if a robot presents physical danger to spectators, itself, or the opposing robot. The motors have plastic gears and will break if you attempt to drive your robot through a wall. Disqualified robots will be removed from the track and placed in a position of shame. However, disqualified robots get credit and penalties according to the scoring criteria up to the point of disqualification. For example, if your robot crosses one milestone (**M**=1), and then it gets disqualified, your score will be -1.

The Strategy and Sportsmanship

The basic goal is to move quickly and accurately around the track. The robot is supposed to move in an intelligent manner. Your TA will not allow unintelligent robots to compete. If you are not sure what intelligent means, run your strategy by your TA. No pieces may be intentionally dropped by your robot. The opponent's robot or the track may not be altered in any way. Pieces unintentionally dropped by your robot will be removed by the TA-referee. Good strategy, knowledge of your position/orientation, and reliability will be important. Because sensors can be unreliable, an effective solution must be robust, so that the robot acts in an appropriate manner even when presented with inaccurate sensor data. Good sportsmanship must be followed.

The Competition Format

The contest has three phases, which are pre-qualifying, qualifying, and finals. During the *pre-qualifying* round, which you will demonstrate to any TA or instructor during any regularly scheduled lab on or before Thursday April 30th, your robot must get a positive score and at least two milestones in order to move on to the qualifying round. There will be 20-point grade bonus if you can pre-qualify on or before April 28th. Pre-qualifying and qualifying rounds will be run with one robot on the track at a time. The number of runs will depend on the number of robots ready to race and the time remaining in the lab period. The pre-qualifying rounds are run on a first-come-first served basis.

Pre-qualified machines will then race twice in the *qualifying* round, occurring during the Tuesday lab and class periods on May 5th. Each robot team will be assigned by the TAs to a particular Tuesday lab/class period for the qualifying competition. The TA will decide the race order during the qualifying runs. Qualifying competitions occur on the ECJ outdoor plaza (with an indoor backup location in case of rain, TBD) using a track different from pre-qualification and different from the finals. The accumulated scores of these qualifying races will be used for seeding into the final competition. The total number of positive points scored during the qualifying rounds will be used as a tie-breaker in the seedings. Machines will be able to qualify after their regularly scheduled qualifying time (ask the TAs and Valvano for a time), but will be given a lower seeding and a grade penalty.

The *final* competition will take place on Thursday May 7th during the class period. By 2pm, Thursday 5/7, the race track for that day will be selected from one of the tracks posted on the web site, and it will be built on the ECJ plaza (or the backup location). The race direction will be determined by coin flip at the start of each race. You will be allowed to run tests on the race track from 3pm to 5pm. Get a good night sleep before the race (i.e., charge the battery for 8 hours). After every able robot has raced three times, the 4th 5th 6th and 7th place robots will race once more. The winner of this race and 1st 2nd and 3rd place robots will compete in one final race to determine the overall winner. The winner of this final race will be crowned champion. In case of a tie, additional race-offs will be held. Depending on the number of qualifying robots, some robots with a high seeding may race with fewer competitors. One possible schedule with 13 robots (1=top to 13=bottom) is as follows:

Thursday 5/7, ECJ plaza
- 5:05 Robots 10,11,12,13
- 5:10 Robots 1,2,3,5
- 5:15 Robots 4,6,7,8
- 5:20 Robots 9,10,11,12
- 5:25 Robots 2,3,5,13
- 5:30 Robots 4,6,7,8
- 5:35 Robots 1,9,12,13
- 5:40 Robots 8,9,10,11
- 5:45 Robots 2,5,7,8
- 5:50 Robots 1,3,4
- 6:00 Race-off with 4th through 7th place for a spot in the finals
- 6:10 Final race

The TA chooses the four starting locations.
Teams then place robots on track with highest seeded robot choosing first.

The Robot

Each team will choose a name for its robot. Please limit your name to 20 characters including spaces. Robots will be constructed using materials from the robot kit. A standard 9.5-inch diameter Frisbee will be used as a base, which should be approximately level to the floor so the center of the bumper (i.e., the point it would contact the wall or another robot) exists from 1.25 to 1.5 inches from the floor. The purpose of the Frisbee is to prevent two robots from getting tangled together. Do not cut any part of the circular bumper (i.e., there should be a continuous bumper around the robot). Input devices include four ultrasonic ranging sensors, four IR range sensors, and four tactile touch sensors. Motors include one servo, four geared DC motors, two high-RPM low-torque DC motors, two stepper motors and one solenoid. The *bounding box* is defined as the 3-D cylinder with 90 degree angles that should contain all the hard components of your robot. Any object with a well-defined volume is defined as hard, including grabbers, sensors, and wheels. The *footprint* of your robot is defined as the floor of its bounding box, as shown in Figure 7.2. The footprint must equal a standard circular Frisbee at all times during the competition. Soft components of your robot such as rubber bands, string, and piano wire (whiskers) may extend 2 inches beyond the footprint. These soft components must be designed in such a way to minimize the chance of tangling with another robot. Tangled robots will result in a double disqualification. The maximum wheel diameter is 3.25 inches (to limit torque to the motors). The maximum *height* of the bounding box is 12 inches at all times during the competition. In other words, your robot cannot unfold so that is becomes larger than the maximums of 9.5 inch diameter, and 12 inch high. Your wheels must be on the ground at all times. You cannot go over a *wall* during the competition.

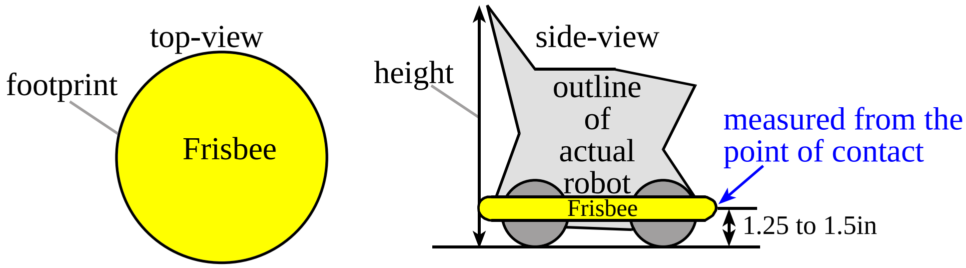

Figure 7.2. Definitions of footprint and height.

Three Laws of EE445M/EE380L.6 Robotics:

1. A robot may not harm a human being, or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given to it by TA-referees.
3. A robot must protect its own existence and the existence of the other robots, as long as such protection does not conflict with the First or Second Law.

The Judges Rule

- The instructor is the judge.
- Contestants should ask the judge about possible designs or strategies that may be questionable under any of the rules.
- Contest rules and procedures, or even the game, may have to be altered during the semester. As much notice as possible will be given.
- The judge may alter or eliminate any rule, or add rules, at any time.
- The TA-referees will decide any discrepancies in the contest play.
- All decisions of the TA-referees are final.

Restrictions

Teams are encouraged to seek advice and help from any source, but the design, construction, and software must be exclusively their own work. All entries must be solely controlled by two ARM microcontroller boards. You will be allowed to compete with a robot controlled by one ARM board, but a grade reduction will occur. Other than the repair events, there can be no human intervention once the game begins. A robot that is moved by a team member other than the repairman during a game will be disqualified for that round. Also, team members touching the walls during play risk disqualification of their robot. Teams may qualify only one robot, and only that robot may race in the preliminary and final rounds. Modifications to your mechanical, electrical or software components following qualification are encouraged. Repair and reprogramming are allowed during the competitions only if time permits. A robot cannot be designed to destroy the physical structure of its opponent. In particular, robots are not allowed to destroy their opponent's microprocessor board, wheels, power, or sensors. Jamming the opponent's sensors with light or sound is not allowed. However, passing a slower robot is an obvious strategy. No parts or substances may be deliberately dumped, deposited, or otherwise left to remain on the track surface. A machine that appears to have been designed to perform such a function will be disqualified. Pieces that accidentally fall off robots will be removed from the arena during a game by the TA-referee at his/her discretion. No adhesives or sticky substances (such as glue or tape) may be applied to any part of your robot, the opponent's robot, the floor, or the walls. Any machine that appears to be a safety hazard will be disqualified from the competition.

Robot Structure Restrictions

Even though the kit has a lot of components, you will be allowed to purchase and use additional electronics and sensors as long as they are powered from the one 8.4V battery supplied in the kit. Additional components can be borrowed from the professor as long as there is sufficient quantity for all teams to have a similar quantity. There are boxes of random structural components (erector pieces), which may also be borrowed starting 1 week after you receive your kit. Teams may possess only one battery at a time. All kits contain approximately the same components, except there is variability in the touch sensors. Electrical components from the checkout counter may be obtained (such as resistors, capacitors, transistors and IC chips). Other than what is in the kit, no batteries, servos, solenoids, or motors may be used (because this will give all robots the same mechanical capabilities). If a motor or sensor is broken or damaged, it will be replaced. You may not employ devices to increase the voltage above the 8.4 volts provided by the battery (e.g., no DC-DC converters). You can have no electrical, mechanical, chemical, or nuclear power source other than the provided 8.4 V battery (e.g., no CO_2, butane, or gasoline may be used as a power source). You may not use another battery, even if it is lower voltage than 8.4V. You may use lubricants on the metal parts (not in the motors or sensors), as long as all components are cleaned prior to returning the kits. Rubber bands may be used as structural components (e.g., as tires or holding things together), but not used as a power source. For example, the motor can be used to transfer energy from the battery to a spring or rubber band. The spring or rubber band may then cause a flipper, a leg, or an arm to move or return back. In other words, humans cannot preload potential energy into a spring or rubber band prior to the competition.

The use of temporary binding components such as nuts, bolts, rubber bands, cable ties, string, and wire is encouraged, but please don't use methods that permanently alter the pieces such as tape, hot glue, epoxy, or welding. Since tape is so nasty to clean up, the use of tape is not allowed. Please discard all rubber bands before returning the kit (these items get nasty after sitting in the box for 12 months). You may solder wires to the connectors and switches (but not to Ping and IR distance sensors), as long as permanent damage is avoided. Drilling additional holes in the erector pieces is allowed, as long as the original usage of the piece is not altered. You may alter your own structural pieces (wood, plastic, metal) as you wish. The Frisbee base may be altered in any way you wish as long as it still acts like a bumper, preventing robots from getting tangled. You may use your own Frisbee as long as it is the same size as all the others. Purely decorative items may be added, within the size limits and good taste.

The $50 Rule

Each team is allowed to purchase additional parts up to a $50 limit not including taxes or shipping. These additional parts may include electrical components (such as sensors, transistors, amplifiers, motor drivers etc.), and structural components (such as wood, metal, plastic, wheels, shafts, gears, chains, pulleys, and slip couplings). Repeating from above, you cannot purchase or use additional batteries or motors. Components from the checkout or the instructor's office do not need to be logged and do not count towards this $50 limit. You do not need to measure the cost of readily available supplies such as nuts, bolts, rubber bands, cable ties, string, paper, cardboard, wire, your ARM boards, or your protoboards. To purchase an additional item, it must be readily available to all teams at that price. *Readily available* means, if it is purchased, it can be received 7 days later. Free components may be obtained as long as they are logged and equally available to all teams. The robot name, description, cost, and source must be first approved by a TA or instructor, then emailed to the instructor, titled *The $50 Rule*. All extra items must be logged within 5 days of its purchase. If you purchase a part, log it, then decide not to use it, the part does not count towards your $50 limit. After the component is purchased, the receipt must be shown to your TA (with date and cost). Any machines found with added parts that have not been documented in this fashion will be disqualified.

Each robot kit contains the following materials, partially donated by Tivioli

- Amazon.com, Erector Special Edition Anniversary Set, 643 pieces, $109.99 each
  Case, DC toy motor, wheels, axles, drive belts, pulleys, large and small metal pieces, many nuts and bolts
- BG Micro, www.bgmicro.com,
  1x BAT1060, 8.4V/3.8AH, NiMH battery, $7 each (with charger $3) (no longer in stock)
  2x MOT1050, Geared DC motor, Merkle-Korff S1627B ($7.95 each)
- Ultrasonic range sensors
  either 4x Ping))) sensors, $29.95, Parallax, www.parallax.com
  or 4x Module HC-SR04 SuntekStore, $3.41, # 14006829
- Sparkfun, www.sparkfun.com
  4x GP2Y0A21YK IR range sensor, SEN-00242 ($13.95 each)
  The field of view is 10-cm wide cylinder in front of them (not cone-shaped)
  They operate differently positioned horizontally versus vertically
  4x Jumper wires, SEN-08733 ($1.50 each)
- All Electronics Corporation, http://www.allelectronics.com, Phone 1-800-826-5432
  either 4x SMS-189, MINI-SNAP-ACTION SWITCH, $0.50 each
  or 4x SMS-174, MINI-ACTION ROLLER SWITCH, $1.25 each
- Jameco, 1-800-831-4242, www.jameco.com
  1x 157067, 14500 RBT geared servo motor (bag of servo parts), $17.95 each
  2x 400961, 3 in Tires, Neoprene Foam Tread, $7.25 per pair
  1x 358336, Ball and caster, $6.00 each
- Mounting Hubs two for a 4mm shaft, two for a 5mm shaft (4mm shaft drilled to a 3/16 inch diameter)
  either 4x Jameco 386118, Mounting Hub, $7.99 per pair
  or 4x Robot Market Place, Part# 0-MHUB06, http://www.robotmarketplace.com, $8.49 pair

Electronics available for checkout on the second floor

IRF522, N CHANNEL MOSFET, drives up to 8A, $0.40 each
TIP120, NPN TO-220 DARLINGTON, $0.65 each
1N914, Switching diode
L293B, H-bridge driver, $1.15 each
7805 voltage regulator use with Servo

DC Motor specifications (Merkle-Korff S1627B)

240 RPM at 12V
No load 180 RPM at 8.4V, 250mA
4.8 mm (3/16 inch) diameter shaft

HS-303 - Economy Standard Servo

For more information on servos, search the web site: http://www.brookshiresoftware.com/
Servos are a popular mechanism to implement steering in robotics. Ranging from micro servos with 15oz-in torque to powerful heavy-duty sailboat servos, they all share several common characteristics. A servo is essentially a motor for which you set the desired position or angle. The servo "knows" two things: where it is (the actual position) and where it wants to be (your desired position). When the servo receives a desired position, it attempts to move the servo horn to the desired position. The task of the servo, then, is to make the actual position equal to the desired position. The first step to understanding how servos work is to understand how to control them. All timing and electrical characteristics described here have been experimentally determined from a "HS-303 HiTec" servo. The servo is controlled by three wires: ground (black), power (red), and command (yellow). Power is usually between 4V and 6V and should be separate from system power (as servos are electrically noisy). Even small servos can draw over an amp under heavy load so the power supply should be appropriately rated. Though not recommended, servos may be driven to higher voltages to improve torque and speed characteristics. Servos are commanded through "Pulse Width Modulation," or PWM, signals sent through the command wire. Essentially, the width of a pulse defines the position. For example, sending a 1.5ms pulse to the servo, tells the servo that the desired position is 90 degrees. In order for the servo to hold this position, the command must be sent at about 50 Hz, or every 20 ms.

Position	Pulse Width
Minimum	0.5ms
Center	1.5ms
Maximum	2.5ms

Example Pulse

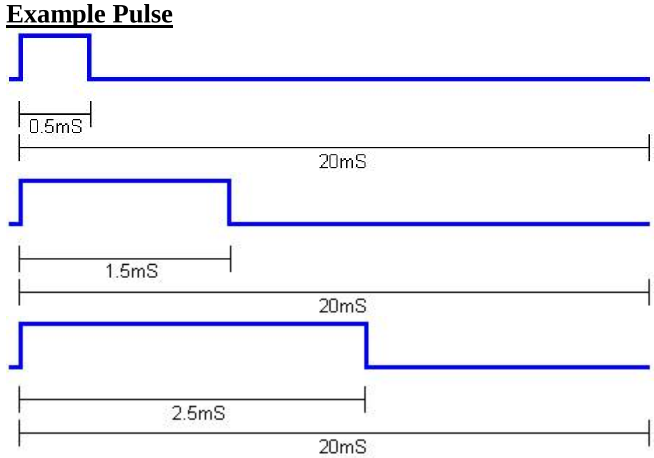

*Figure 7.6. The timing constraints of the **HS-303** servo (http://www.brookshiresoftware.com/) not drawn to scale.*

If you were to send a pulse longer than 2.5 ms or shorter than 0.5 ms, the servo would attempt to overdrive and damage itself. Once the servo has received the desired position (via the PWM signal) the servo must attempt to match the desired and actual positions. It does this by turning a small, geared motor left or right. If, for example, the desired position is less than the actual position, the servo will turn to the left. On the other hand, if the desired position is greater than the actual position, the servo will turn to the right. In this manner, the servo "zeros-in" on the correct position. Should a load force the servo horn to the right or left, the servo will attempt to compensate. Note that there is no control mechanism for the speed of movement and, for most servos, the speed is specified in degrees/second. Indeed, one of the primary tasks of electronics is to synthesize servo speed control by stepping through a series of positions. *If you plan to use a servo, please add an additional 7805 regulator, creating a separate +5V source just for the servo.* If you connect the servo to the +5V used by the microcontroller, then the servo will cause transient power losses to the microcontroller, causing software resets.

Operating Voltage	4.8V		6.0V	
Torque	42 oz-in	3.3 kg-cm	49 oz-in	3.7 kg-cm
Speed @ 60 degrees	19 sec		15 sec	
	Standard		Metric	
Size L x W x H	1.6" x 0.8" x 1.4"		41 x 20 x 37mm	
Weight	1.9 oz		48.5g	

Table 7.1. HS-303 HiTec" servo specifications (http://www.brookshiresoftware.com/).

BAT1060 – NiMH Battery Pack

The 8.4V/3.6AH NiMH battery pack has two connectors. The most important design factor is to prevent the motor current (which can be over an amp) from passing through the microcontroller boards (either power or ground). The polarized RC-type connector (like the picture on the right) is used for charging the battery. It takes about 8 hours to fully charge the NiMH battery, using the 12V wall wart (with the matching RC-type connector), but please do not power the robot or microcontroller while charging. ***Please do not charge for more than 8 hours.*** The red (8.4V) and black (ground) wires must be connected to a soldered circuit board for powering the DC and stepper motors. Do not connect 8.4V directly into your microcontroller board. Please double-check the polarity of each connector before soldering it to the PCB. Figure 7.7 shows one way to route power. If you use a servo, it will need a second 7805 regulator, and thus have its own +5V signal. If you use the servo its regulator and interface need to be designed and soldered onto this circuit board as well. On the soldered board, all +5V/VBUS signals (except servo +5V) are connected. There must be exactly one power source for this +5V/VBUS signal:

- +5V comes from battery via 7805, both LaunchPad power switches are off
- +5V comes from left LaunchPad USB, battery is off, right LaunchPad power switch is off
- +5V comes from right LaunchPad USB, battery is off, left LaunchPad power switch is off

Be very careful that the DC motor or servo currents do not travel through the microcontroller boards. All motor interfaces should be placed on the soldered board. One 7805 is sufficient to power two ARM boards, Ping))) and IR distance sensors. The ARM boards create the 3.3V power from the +5V power you connect to the VBUS signal. For each LaunchPad please do connect VBUS to +5V. All grounds must be connected. However, do not connect together the +3.3V from the two LaunchPads. You may use one or both +3.3V power in your interfaces.

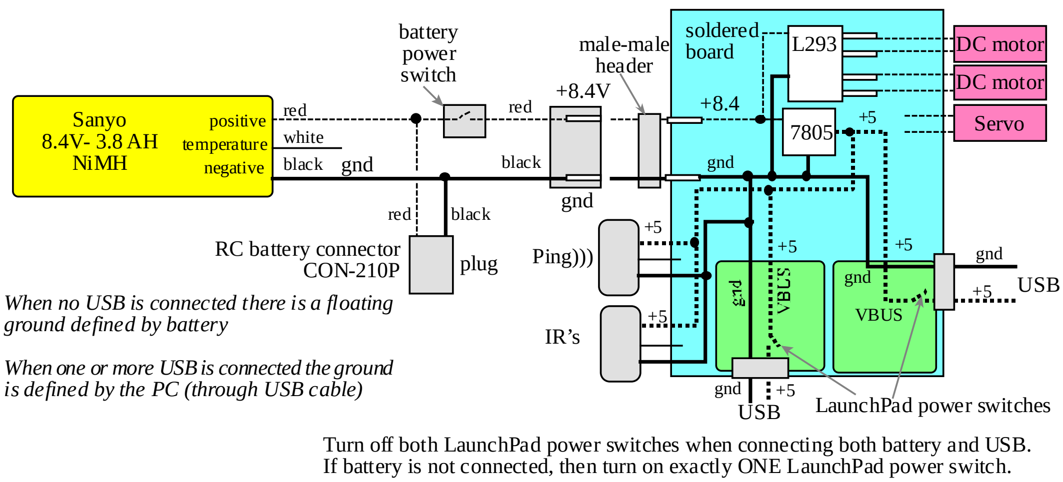

Figure 7.7. Power connections for the robot (large solid is ground, large dotted is +5V, small dotted is +8.4V).

During debugging, you need to connect one or both USB to the PC(s). There are two options. If you need +8.4 V to power the motors: turn off both power switches on the two LaunchPads and turn on the battery power. Now, you can connect either or both USB cables to computer(s) and debug. If you do not need +8.4V to the motor: turn on exactly ONE power switch on ONE LaunchPad and turn off the battery power. You can connect one or both USB cables to computer(s) and debug.

One way to test the battery is to first drain the energy from the battery with a 10 to 20 ohm load. NiMH batteries have a uniform voltage until they are almost empty. So, once the voltage drops to 5V, we can consider the battery empty. Make sure the test load can handle the I=8.4/R current that will be flowing. It should take hours to drain the battery. Second, charge the battery for 8 hours (no more than 8). Third, discharge the battery again measuring the voltage across the load. If you use a variable resistance load, such as the motors, you should measure both voltage and current. You may not get the full 3.6 amp-hour specification, but you should get at least half this amount.

7A. Preparation (due Wednesday 4/15)

1) The robot design team with 3-5 members was established as part of Lab 6. The members do not have to be in the same lab section or have the same TA as the previous labs. The ideal team has at least one member with strengths in the areas of mechanical design (e.g., pulleys, belts, motors, and rack-and-pinion steering), electrical interfacing (e.g., transistors, currents, back EMF, servo interfacing, and sensor interfacing), power management (e.g., maintaining constant power to the microcontroller while operating the motors), software design (e.g., device drivers, juggling multiple time-critical tasks, making it fit into the 32k compiler-limited space), high-level control, and project management (e.g., conflict resolution, report generation, and keeping on schedule). Part of your final project grade will be confidential peer evaluations, so choose your team wisely, then make a commitment to get this project finished. Please turn in the following:

- Robot name
- First and last names of all team members
- Home phone numbers of all team members
- Email addresses of all team members
- Select at least two times each week for an official team project meeting (without the TA)
- List all the regularly scheduled lab hours that it is possible for all your members to meet weekly with a TA. We will select one of these hours to be when your preparation and demonstrations will be due.

7A. Procedure (do this during your lab period)

1) Design and build the mechanical aspects of the robot. It must move, turn, and carry the electronics. Make a rough mechanical sketch of the robot showing how it moves, turns, carries, and senses. A detailed drawing will be required for the final report, but at this time only a rough sketch is required. I.e., just enough detail for the TA to understand your basic approach, but not enough detail for someone to build a duplicate.

2) Design the electronic interface between the steering and power motor(s) and the microcontrollers. We expect you to use the 8.4V NiMH battery, but you may use an external regulated supply during testing. Please test the circuitry before connecting to the microcontrollers. Snubber diodes must be used for all devices having an inductive load. Please consider the required current when designing the interfaces. Use a multimeter to measure the actual voltages and currents across the motors. Watch the power signals on the scope during times when motors are turned on and off to verify a constant power supply line to the microcontrollers. Please thoroughly test all interfaces before connecting to the microcontroller. Include low-level hardware interface like the H-bridge.

3) Show how the system will be powered.

4) Design the low-level software drivers for the movement and steering motors. *It is a requirement that the robot use some sort of controller feedback (sensor input->motor output).* You can control period, speed, angle to wall, or distance to the wall. If you want to control another parameter, please get approval from your TA.

5) Write a simple high-level main program to test the movement and steering. Measure the maximum speed of the robot. Experimentally determine the best way to make turns. Calculate the accuracy of the turning algorithm. I.e., if you say turn +90, how many degrees does it actually turn?

6) Debugging a complex system like this is very hard. Please design features into the robot to assist in debugging.

7A. Deliverables (exact components of the lab report, to be included in the final 7C report)

A) Objectives (not required for this 7A)

B) Hardware Design

1) Rough mechanical sketch of the robot (Procedure 1)
2) Electrical circuit diagram for the motor interfaces (Procedure 2)
3) Power supply circuitry (Procedure 3)

C) Software Design (printout of these software components)
1) Low-level device drivers for the motor interfaces (header and code files) (Procedure 4)
2) High-level test program to evaluate movement and steering (Procedure 5)

D) Measurement Data
1) Give the voltage and currents of each of the motors used (Procedure 2)
2) Give the robot speed and turning accuracy (Procedure 5)

E) Analysis and Discussion
none

7A. Checkout (show this to the TA by Wednesday 4/22, or Thursday 4/23 depending on lab session)

You should be able to demonstrate to the TA the driver motors spin under software control both with the robot on the ground moving and lifted off the ground. If you control distance to wall, demonstrate your robot can follow a straight wall. You should demonstrate the robot moving and turning. Explain the debugging features of your device. From a software design perspective, the OS is at a low-level. Therefore, the OS needs to be designed in from the beginning, and not added on at the end. All three demonstrations (7A checkout, 7B checkout, and the competition) must employ the RTOS running on both microcontrollers.

7B. Preparation (do this before your lab period, but don't turn it in)

1) Develop a high-level plan of how your robot will compete. Your algorithm must involve abstractive methods. I.e., it must be layered, with a high-level algorithm separated from the low-level details of how the machine operates. A finite state machine is one example of how this abstraction might be implemented. You are of course allowed to develop your own approach, as long as there is a clear mapping from the high-level algorithm to the eventual C code. For example, if you decide to implement a finite state machine, draw the state graph for competition algorithm. If you use fuzzy logic, draw the fuzzification and defuzzification graphs.

2) Develop an initial data flow graph of the hardware-software system. Include the motors, sensors, interface drivers, interrupt service routines, and strategic global variables.

3) Develop an initial call graph of the software system. Include foreground and background modules and their linkage (which ones call which).

4) Write the header file for the low-level sensor software driver.

7B. Procedure (do this during your lab period)

1) Design the electronic interfaces connecting the high-level sensors to the microcontroller (e.g., bumper switches plus integration of Lab 6 sensor interfaces). Again, please test the circuitry before connecting to the microcontroller.

2) Design the software drivers for the detection of the walls, to avoid damaging the motors (robot safety is a high-priority process). Your robot must stop **(and back up)** if it hits a wall.

3) Design the software drivers for the sensors, creating software modules with prototypes for public functions in the header file and implementations of both public and private functions in the code file.

4) Write a second high-level main program to test the movement, steering and sensors. This program should demonstrate most of the middle-level building blocks that will be required for the competition. Monitor the power supply current during various operations (stopped, moving and turning).

5) Provide debugging monitors so you can experimentally verify the robot can tell how far it is from the walls.

7B. Deliverables (exact components of the lab report, to be included in the final 7C report)

A) Objectives (not required for this 7B)

B) Hardware Design
1) Most recent electrical circuit diagram for the motor interfaces (7A Procedure 2)
2) Most recent power supply circuitry (7A Procedure 3)
3) Electrical circuit diagram for the sensor interfaces (Lab 6 and 7B Procedure 1)

C) Software Design (printout of these software components)
1) Low-level device drivers for the motor interfaces (header and code files) (7A Procedure 4)
2) Low-level device drivers for collision detection (header and code files) (Procedure 5)
3) Low-level device drivers for the sensor interfaces (header and code files) (7B Procedure 2,3)

4) High-level test program to evaluate movement, steering and sensing (7B Procedure 4)

D) Measurement Data

1) Power supply current for various operations (7B Procedure 4).

2) Accuracy of the positioning system, knowing where it is (7B Procedure 5)

E) Analysis and Discussion (none)

7B. Checkout is the pre-qualifying round (show this to the TA by Thursday 4/30)

Show Lab 7B Procedure 5 to your TA. The robot should be able to run for 180 seconds getting a positive score travelling at least two milestones. Show your debugging monitors demonstrating the robot is able to know where it is relative to the track walls. Explain how you tested the system. There will be a 20-point bonus for qualifying on or before the last lab section on Tuesday 4/28.

7C. Preparation none

7C. Procedure (do this during your lab period)

1) Make detailed mechanical drawings and/or photos of the robot showing how it moves, turns, carries, and senses. There should be enough detail so the robot could be duplicated.

2) Debug the high-level competition algorithm. Write the C code for the competition algorithm using abstractive methods. It should have a clear mapping from the high-level algorithm to the C code.

3) Draw the final data flow graph of the hardware-software system. Include the motors, sensors, interface drivers, interrupt service routines, and strategic global variables.

4) Draw the final call graph of the software system.

7C. Deliverables (exact components of the lab report, due Friday May 8)

A) Objectives (1/2 page maximum)

B) Hardware Design

1) Final mechanical drawing of the robot (7C procedure 1)

2) Final electrical circuit diagram for the motor interfaces (7A Section 2)

3) Final power supply circuitry (7A Section 3)

4) Final electrical circuit diagram for the sensor interfaces (7B Section 1)

C) Software Design (printout of these software components)

1) Low-level device drivers for the motor interfaces (header and code files) (7A Section 4)

2) Low-level device drivers for the sensor interfaces (header and code files) (7B Section 2)

3) High-level competition algorithm (7C Section 2)

4) Final data flow graph

5) Final call graph

D) Measurement Data

1) Scores during qualifying and preliminary competitions

E) Analysis and Discussion (2 page maximum). In particular, answer these questions

1) What is the effect of time delay in your control system?

2) What sensors would you need to develop a more effective passing strategy?

3) If you hit the wall a lot, how could you have changed the design to be more effective? If your robot can travel 3 milestones without hitting a wall, you can skip this question.

F) Post-mortem concerning team member interactions (attached to the report)

1) Each team member evaluates each other team member including oneself

Simply list one or two weaknesses.

Simply list two or three strength characteristics.

2) Major failures in the way the team interacted (if any)

3) Major successes in the way the team interacted

G) Peer Review (each student submits independently and confidentially directly to the TA)

Classify each team member including oneself as:

- worked harder than average (explain),
- worked an average amount,
- worked less than average (explain)

7C. Checkout is the qualifying and final rounds (show your robot to the class on demo day)
You should pre-qualify, compete in the qualifiers, and then run the finals.

Lab 7 as a total of 300 points (counts for 3 labs.) Your Lab 7 grade will have five components. The first component is the demonstration of your microcontroller-controlled robot that moves (7A checkout). The robot must be controlled using a well-defined control algorithm such as incremental PI or fuzzy logic running in real time with real-time feedback. There will be no Lab7A report or Lab7B preparation. You will be awarded 50 points if your robot moves under microcontroller feedback control and no points if you fail to demonstrate motion by April 24. There will be no later checkouts of this part. You may demonstrate the motion to any of the TAs, or even the instructor himself. The second component is the demonstration that your robot is able to run for 180 seconds getting a positive score travelling at least two milestones. (7B checkout, i.e. pre-qualification). You will be awarded 50 points for Lab 7B checkout by April 30, which includes demonstrating to the TA your debugging features. There will be 20-point bonus if you can do Lab 7B checkout on or before April 28. You can qualify with any TA. There will be no checkouts later than May 1st for this part. You may demonstrate the Lab 7B checkout to any of the TAs, or even to the instructor himself. The third component is qualifying round, where your robot must achieve two milestones and get a positive score. There will be no Lab7B report or Lab7C preparation. The fourth component is the scoring during the qualifying phase of the competition. The full 50 points will be awarded to teams that score high enough to be in the upper 50% seeding. Fewer points will be awarded for those teams in the bottom 50% seeding, depending on the specific scores and performances of the robot. After Tuesday 5/5 8pm, there will be late checkouts of the qualifying round. Pre-qualifying and qualifying rounds will be run with just one robot. The final competition will be 5-6:30pm Thursday May 7th. The last component is the Lab 7C report (worth 100 points), which is due by 5pm, Friday May 8th. This report will be graded in the usual manner.

- 50 points for Lab 7A checkout, Wednesday 4/22 or Thursday 4/23
- +20 bonus for early Lab 7B checkout by 4/28
- 50 points for Lab 7B checkout, Thursday 4/30 (2 milestones and get a positive score)
- 50 points for qualifying round for getting at least 2 milestones and get a positive score
  - Tuesday 5/5 (as specified by your TA)
  - Robots run one at a time with average score over 2 or 3 runs (depending on time)
  - Late checkouts of the qualifying round get the lowest seeding
- 0 to 50 points for seeding in the qualifying round,
  - 50 points for upper 50%,
  - 25 points for next 25%,
  - 0 points for the lowest 25%
- 100 points for the Lab 7C report (Lab 7A 7B 7C deliverables) due 5pm Friday 5/8

The race winners only have to write parts F and G of the final report

Robot parts

I have additional parts that can be assigned to your kits, such as servos, optical sensors, switches, DC motors, springs, belts, cable ties, rubber bands, heat-shrink, wire, erector pieces, PN2222, IRF540, IRF522, FQP27P06 and 7805 regulators. I will dispense components as long as I have enough stock to allow all teams to have equal access. Unfortunately, I do not have enough batteries or distance sensors for groups to have more than their allotment.

My inventory procedures are limited, so please take good care of your kit, and return the kit in a manner that will make next year's competition enjoyable for the next team using your kit.

http://www.robotstore.com/ http://www.robotmarketplace.com
http://www.sparkfun.com/ http://www.parallax.com/
http://www.towerhobbies.com/

Surplus parts

http://www.bgmicro.com/ http://www.allelectronics.com/

Full-service parts

http://www.jameco.com http://www.digikey.com http://www.mouser.com

Part search engine

http://octopart.com/

Baggy of parts for Labs 6 and 7 (do not need to return, but return any parts you do not use)

____ 1, Jameco Part no. 206587, 6.8 x 4.3 inch protoboard
____ 2, Jameco, #231546 0.156 in header (for two DC motors)
____ 2, MCP2551 CAN driver chips
____ 2, 100 ohm resistors (should have been 120 ohms)
____ 1, 2 by 40 male headers, snap into four 2 by 10 headers for two LaunchPads
____ 1, 2-pin 0.1in header for power in from battery (do not cut any battery wires)
____ 1, 7805 regulator (8.4V in, 5V out)
____ 2, 4.7uF tantalum caps for regulator
____ 8, 1N914 diodes for motor driver interface, 1N914B-ND
____ 1, 16-pin socket (can be used for L293, get L293 from checkout and return it at the end of the semester)
____ 1- standard Frisbee (purchased at Wal-Mart for $1.00)