

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 9-11 on input capture, PWM, motors and control
 - Textbook Chapter 10 on control systems,
 - Data sheets for L293, GP2Y0A21YK, HC-SR04, and Ping))).

- Starter files**
- See Lab 6, plus `PWMDua1_4C123` project (dual PWM output)

Background

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 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 several 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 such that there is a wall either on the left or on the right. Tracks will be designed so that the robot can successfully make the turns without backing up.

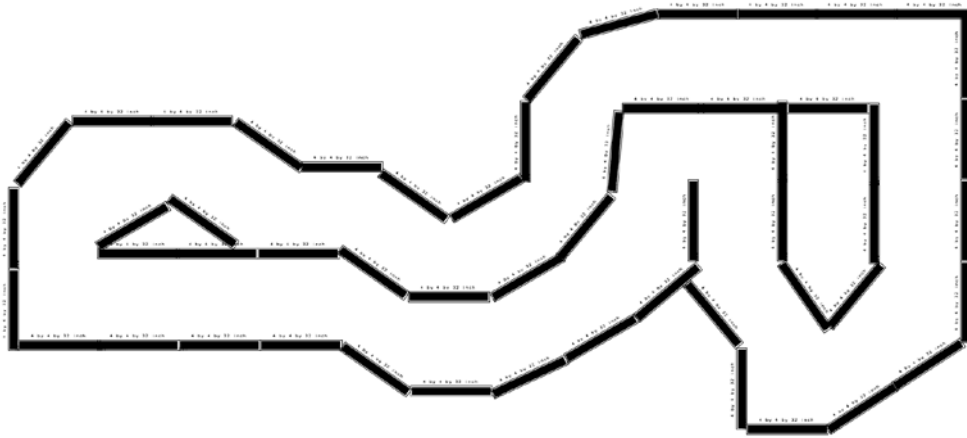


Figure 7.1. Design of Track 7. Milestones are labeled with blue tape on the wood.

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. Up to two robots will begin side by side in pairs or alone at a milestone as chosen by the TA-referee. The TA-referee will tell you which milestone to start at 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 a 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

$$\text{Score} = \mathbf{M} + \mathbf{S}$$

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 robot baseplate 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, and then 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 (Lab 7B), qualifying, and finals (Lab 7C). During the *pre-qualifying* round, which you will demonstrate to any TA or instructor during any regularly scheduled lab on or before Thursday April 28th, 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 26th. 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 3rd. 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 the ETC T-Room/Student Lounge as indoor backup location in case of rain) 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 seeding. Machines will be able to qualify after their regularly scheduled qualifying time (ask the TAs and Professor for a time), but will be given a lower seeding and a grade penalty.

The *final* competition will take place on Thursday, May 5th during the class period. By 2pm, Thursday 5/5, 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 as described in the instructions). 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 5/5, 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 built on a provided robot platform as shown in Figure 7.2. A 11.625 x 7.5 inch plywood board serves as the base plate that holds all robot components, including a servo-driven front-wheel steering assembly, left and right rear wheels driven by DC motors, battery, and mounts for sensor and motor boards with one LaunchPad each connected by a CAN bus from previous labs. Input devices include up to four ultrasonic ranging sensors, four IR range sensors, and two switches. Motors include one servo and two geared DC motors. The *bounding box* is defined as the 3-D cuboid 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 projection of its bounding box onto the floor when viewed from the top. No hard component must extend beyond the 11.625 x 7.5 inch footprint of the base 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 *height* of the bounding box is 12 inches at all times during the competition. In other words, your robot cannot unfold so that it becomes larger than the maximum 11.625 x 7.5 x 12 inch cuboid. Your wheels must be on the ground at all times. You cannot go over a *wall* during the competition.

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.

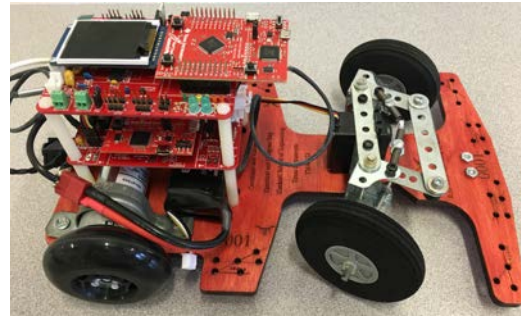
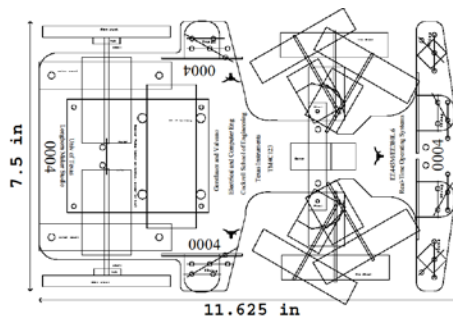


Figure 7.2. Robot footprint and robot platform.

The Judges Rule

- The instructor is the judge. The judge may alter or eliminate any rule, or add rules, at any time.
- 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 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

You will be allowed to purchase and use additional electronics and sensors as long as they are powered from the 11.1V battery of the robot platform. 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 checked out from the TAs. Teams may possess only one battery at a time. All robot platforms contain the same base components, except there is variability in the sensors and additional electronics you put on your robot. Electrical components from the checkout counter may be obtained (such as resistors, capacitors, transistors and IC chips). Other than what is part of the given platform, no batteries, servos, solenoids, wheels, 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 11.1 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 11.1V battery (e.g., no CO₂, butane, or gasoline may be used as a power source). You may not use another battery, even if it is lower voltage than 11.1V. 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 robot. Rubber bands may be used as structural components (e.g., holding things together), but not used as a power source.

The use of temporary binding components such as nuts, bolts, rubber bands, cable ties, string, and wire is encouraged, but please do not use methods that permanently alter the robot 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 robot (these items get nasty after sitting in the box for 12 months). You may solder wires to connectors and switches (but not to Ping and IR distance sensors), as long as permanent damage is avoided. Drilling additional holes in the baseplate and 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. 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, etc.), and structural components (such as wood, metal, plastic, shafts, gears, chains, pulleys, and slip couplings). Repeating from above, you cannot purchase or use additional batteries or motors, and you cannot permanently alter the given robot platform. 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 two PCB boards. 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 platform consists the following parts, partially donated by Tivoli

- Mechanical base
 - Various Meccano/Erector set pieces for base, mounts and steering assembly (~\$1-\$2 per piece to replace)
 - 1x Base plate, laser cut out of 12 x 24 x 0.25in plywood, www.rcplanet.com (~\$5)
 - 8x Nylon standoffs 4-40 hex 1-5/8" plus nuts and screws, Keystone Electronics (~\$5-\$6)
 - 1x 16 Quart Storage Box, 16-5/8 x 11-3/4 x 7 in, White Lid with Clear Base, Sterilite 16448012 (~\$4.50)
- Battery (www.all-battery.com)
 - 1x 11.1V 2200mAh 25C Li-Poly Battery Pack w/ Deans Connector, Tenergy (\$39.99 w/ charger)
 - 1x TN267 LiPO/LiFe Battery Pack Balance Charger, Tenergy
- Rear-wheel drive train
 - 2x MOT1050, Geared DC motor, Merkle-Korff S1627B (\$7.95 each)
 - 2x Aluminum Scooter Wheel Adapter for 5mm Shaft, #2673, www.pololu.com (\$4.95 each)
 - 2x Scooter/Skate Wheel 70x25mm, black, #3272, www.pololu.com (\$2.49 each)
- Front-wheel steering
 - 1x 157067, 14500 RBT geared servo motor (plus servo parts) (\$17.95 each)
 - 2x 400961, 3 in Tires, Neoprene Foam Tread (\$7.25 per pair)
- Ultrasonic range sensors
 - either 4x Ping))) sensors, Parallax, www.parallax.com (\$29.95 each)
 - or 4x Module HC-SR04 SuntekStore, # 14006829 (\$3.41 each)
 - 4x 30cm 2.54mm Female to Female Connect Jumper Wire
 - 4x Ultrasonic mounting bracket (\$1.27 each on eBay)
- IR sensors
 - 4x GP2Y0A21YK 80cm IR range sensor, SEN39046P (\$13.95 each including Grove cable)
 - The field of view is 10-cm wide cylinder in front of them (not cone-shaped)
 - They operate differently positioned horizontally versus vertically
 - 4x Grove connector cable for IR sensor

Additional components available for checkout from the TAs

- Various pieces from Meccano/Erector Special Anniversary Edition set (643 pieces)
 - Axles, drive belts, pulleys, large and small metal pieces, many nuts and bolts
- 2x SMS-189, mini-snap-action switch or SMS-174, mini-action roller switch

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
- 2x 1-1/4-in dia Galvanized 2-Hole Pipe Straps (motor mounts)



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 "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 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.

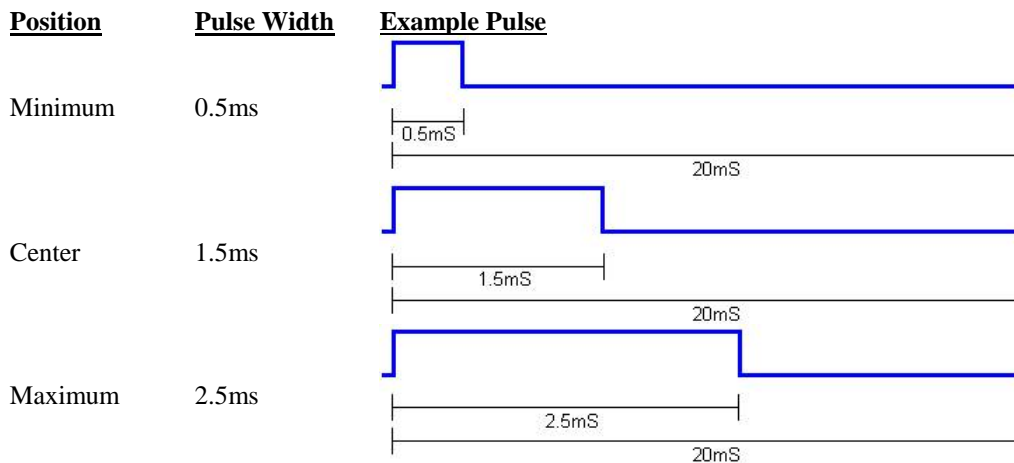


Figure 7.6. The timing constraints of the **HS-303** servo (<http://www.brookshiresoftware.com/>) not drawn to scale.

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. The servo outputs on the motor board are powered by an additional 7805 voltage 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/>).

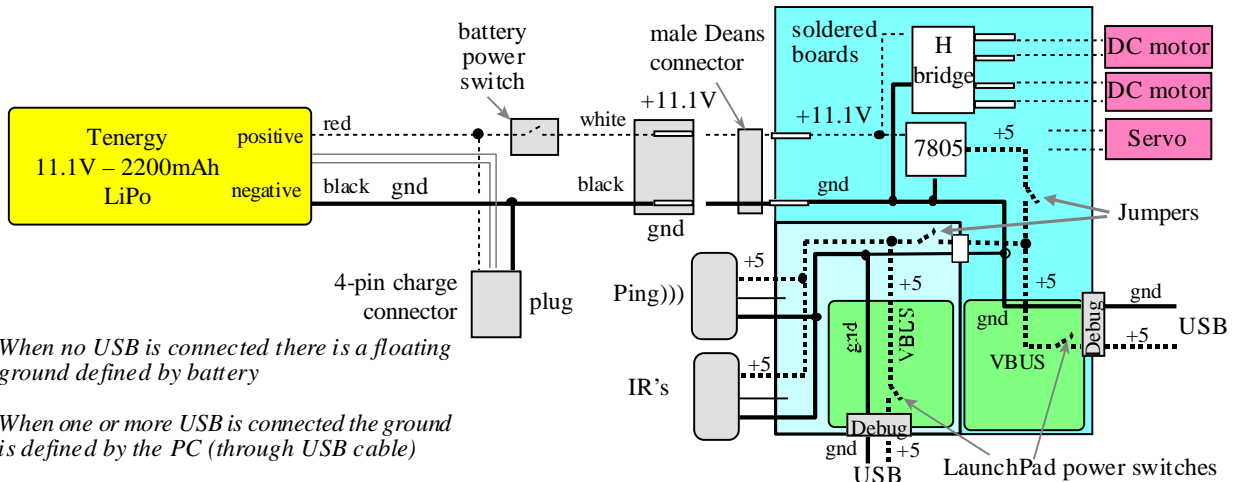
LiPo Battery Pack

The 11.1V 2200mAh Li-Polymer 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 4-pin TP-style connector is used for charging the battery with the provided TN267 balance charger. Please only use the provided charger. Make sure the charger is set to "LiPo" mode before attaching the battery and plugging in the wall wart. A red LED means it is charging and a green LED that the charge is complete. Do not power the robot or microcontroller while charging. The 2-pin female, polarized Deans connector with red (11.1V) and black (ground) wires is used to connect the battery to the motor board via an in-line power switch. Do not connect 11.1V directly into your microcontroller board, and **never over discharge the battery**. Figure 7.7 shows how power is routed in the robot. There must be exactly one +5V source for each board. On the sensor board, JP1 allows the board's +5V power to come from an external source (J13) versus USB. On the motor board, JP2 allows sourcing the boards' +5V power from the battery via a 7805 voltage regulator. The servo has a second 7805 regulator and thus its own +5V signal. For battery operation, all +5V signals (except servo +5V) are connected to each other and the battery by closing the JP1 jumper on the servo board, the JP2 jumper on the motor board, and wiring up the 5V/GND connectors of the two boards. There must then be exactly one power source for this +5V signal:



- +5V comes from battery via 7805, both LaunchPad power switches are off (SW3 set to "DEVICE")
- +5V comes from sensor LaunchPad USB, battery switch is off, motor LaunchPad power switch is off
- +5V comes from motor LaunchPad USB, battery switch is off, sensor LaunchPad power switch is off

One voltage source and regulator is sufficient to power two ARM boards, (Ping))) and IR distance sensors. The ARM boards create the +3.3V power from the +5V power connected to the +5V/VBUS signal. All grounds must be connected. The +3.3V from the two LaunchPads are not connected.



Turn off both LaunchPad power switches when connecting both battery and USB.
If battery is not connected, then turn on exactly ONE LaunchPad power switch.

Figure 7.7. Power connections for the robot (large solid is ground, large dotted is +5V, small dotted is +11.1V).

During debugging, you need to connect one or both USB to the PC(s). There are three options: If you do not need +11.1V 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. If you need +11.1V 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. Alternatively, remove one or both jumpers and always power the corresponding LaunchPad(s) from USB (with or without the motor, i.e. with battery switched on or off).

One way to test the battery is to first drain the energy from the battery with a 10 to 20 ohm load. LiPo batteries have an initial drop and then a uniform voltage until they are almost empty. So, once the voltage drops to 10V, we can consider the battery empty. Make sure the test load can handle the $I=11.1/R$ current that will be flowing. Again, make sure to never drain the battery completely (below 9V) or it will be damaged. Second, fully charge the battery. Third, discharge the battery again measuring the time-integral over the voltage and hence current 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 2200mAh specification, but you should get at least 50%. The robot should operate for hours on one charge.

7A. Preparation (due Wednesday 4/13)

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 names, last names, phone numbers and 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

1) Design and build the mechanical aspects of the robot. Decide about the type, number and placement of sensors you want to use. A drawing will be required for the final report, but at this time only a rough sketch is sufficient. I.e., just enough detail for the TA to understand your basic approach, but not enough detail for someone to replicate.

2) Design and implement the low-level software layer for movement and steering control on top of Lab 6 motor, servo and sensor drivers/interfaces. *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.

3) Write a simple high-level main program to test and debug 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?

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

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

A) Objectives (not required for 7A)

B) Hardware Design

- 1) Rough mechanical sketch of the robot (Procedure 1)

C) Software Design (documentation and code)

- 1) Low-level software layer for movement and control (header and code files) (Procedure 2)
- 2) High-level test program to evaluate movement and steering (Procedure 3)

D) Measurement Data

- 1) Give the robot speed and turning accuracy (Procedure 3)

E) Analysis and Discussion (none)

7A. Checkout (show this to the TA by Wednesday 4/20, or Thursday 4/22 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) Design an initial data flow graph of the hardware-software system. Include the motors, sensors, interface drivers, interrupt service routines, and strategic global variables.

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

4) Design the interface (write the header file) to the low-level software layer(s) for movement and control.

7B. Procedure

1) Design and build the electronic interfaces connecting any additional high-level sensors to the microcontroller (e.g., bumper switches in addition to Lab 6/7A sensor interfaces). Please thoroughly test all circuitry and interfaces before connecting to the microcontroller. We expect you to use the 11.1V battery during competition, but you may use an external regulated supply during testing.

2) Design and implement 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.

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

4) Write a second high-level main program to test and debug 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 to be included in the final 7C lab report and submission)

A) Objectives (not required for this 7B)

B) Hardware Design

1) Electrical circuit diagram for any additional sensor interfaces (Procedure 1)

C) Software Design (documentation and code)

1) Low-level device drivers for the sensor interfaces (header and code files) (Procedure 2)

2) Mid-level software layer for collision detection (header and code files) (Procedure 3)

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

D) Measurement Data

1) Power supply current for various operations (Procedure 4)

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

E) Analysis and Discussion (none)

7B. Checkout, Pre-qualifying round (show this to the TA by Thursday 4/28)

The robot should be able to run for 180 seconds getting a positive score travelling at least two milestones. Show your debugging monitors (Lab 7B, Procedure 5) 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 4/26.

7C. Preparation (none)**7C. Procedure**

- 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) Develop (design, implement, test and debug) the high-level competition algorithm. Write the C code for the competition algorithm using abstractive methods. There should be 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 and submission, due Friday May 6)

- A) Objectives (1/2 page maximum)
- B) Hardware Design
 - 1) Final mechanical drawing of the robot (Procedure 1)
 - 2) Final electrical circuit diagram for the sensor interfaces (7B Procedure 1)
- C) Software Design (documentation and code)
 - 1) Low-level device drivers for the motor interfaces (header and code files) (Lab 6 and 7A Procedure 2)
 - 2) Low-level device drivers for the sensor interfaces (header and code files) (Lab 6 and 7B Procedure 2)
 - 3) Low- and mid-level layers for movement, control, and detection (7A Procedure 2 and 7B Procedure 3)
 - 4) High-level competition algorithm (7C Procedure 2)
 - 5) Final data flow graph
 - 6) 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?
 - 4) 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)
 - 1) Classify each team member including oneself as:
 - worked harder than average (explain),
 - worked an average amount,
 - worked less than average (explain)

7C. Checkout, 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 has 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 25. 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 28, 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 26. You can qualify with any TA. There will be no checkouts later than April 29th 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 Lab 7B report or Lab 7C 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 May 3rd, 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 5th. The last component is the Lab 7C report (worth 100 points), which is due by 5pm, Friday May 6th. This report will be graded in the usual manner.

50 points for Lab 7A checkout, Wednesday 4/20 or Thursday 4/21

+20 bonus for early Lab 7B checkout by 4/26

50 points for Lab 7B checkout, Thursday 4/28 (2 milestones and get a positive score)

50 points for qualifying round for getting at least 2 milestones and get a positive score
Tuesday 5/3 (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 5/6

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

Robot parts

We 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, and erector pieces. We will dispense components as long as we have enough stock to allow all teams to have equal access. Unfortunately, we do not have enough batteries or distance sensors for groups to have more than their allotment.

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

<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/>