ME218c Smart Product Design

ME 218c Spring 2014 Project
Watch Our Robots Levitate During Championship University Presentations (WORLD CUP)

Project Preview on May 23, 2014 6-10 PM in SPDL.
Grading Session on May 27, 2014 6 PM-10 PM.

Goal:

Based on your performance in the ME218a project, you and your team, along with the rest of the ME218c class, have once again been contracted by ESPN0xFF, this time to develop a team of robotic soccer players. Not the usual fare of rolling 'bots driven by instructions from an unseen intellect, your players will be tele-operated by members of the audience and float above the playing field on a cushion of air. Due to concerns about head injuries in last season's JOUSTBALL, the network has decided to monitor for harsh impacts and potential injuries in this season's soccer competition.

Purpose:

The underlying purpose of this project is to provide you with an opportunity to gain experience in integrating all that you have learned in the ME218 course sequence, with an emphasis on the new material in ME218c.

The Task:

Design and build a Pneumatically Levitated Athletic Yacht-Emulating Robot (PLAYER) and companion Communication-Oriented Assistive Control Hardware (COACH) (aka controller).Your COACH must be able to operate any of the soccer PLAYERs in the class and conversely your PLAYER must be controllable from any of the COACHs built by other teams.

Specifications

General:

1. Each team will construct a PLAYER and a COACH.
2. The PLAYERs are floating vehicles designed with the ability to manipulate an SPDL supplied "soccer ball"(softball sized WIFFLE ball).
3. The COACHs are the wireless remote controllers for the PLAYERs.

Basic Game Play:

4. A match will be a competition between 2 teams. Each team will consist of 6 PLAYERs and 3 COACHs. At any point in time, only 3 PLAYERs per side will be active and controlled by the 3 COACHs from that team. The remaining 3 PLAYERs per side will begin in the On Deck Regions for their side.
5. The goal of the game is to score as many goals as possible within an 8min. match.
6. Players will deplete their 'Energy Store' at a rate that depends on the setting commanded to the lift fan (see LiFKM below). While in the On Deck regions, the 'Energy Store' will slowly recuperate.
7. When a player's Energy Store drops below 10% it **must** be moved to the On Deck region and swapped out for another player currently recuperating in the On-Deck region. To avoid injury, PLAYERs with low energy must not exert themselves too much; they are prohibited from kicking.
8. An active player must be 'Tagged Out' of the game before one of the On-Deck players can be activated.
9. To 'Tag Out', the PLAYER must physically tag a paddle at the back of an open On-Deck region on their side of the field. The field will then broadcast a message (*TagDetected*) to all PLAYERs and COACHs to announce that a PLAYER has tagged a particular On-Deck location (R1-4 or G1-4). All PLAYERS must translate and relay these *TagDetected* messages to the on-board LiFKIM (see below). When the COACH wishing to 'Tag Out' its PLAYER receives the *TagDetected* notification from the field, it must send a message (*TagOut*) to its PLAYER telling the PLAYER to 'Tag Out' at the specified On-Deck location. When the PLAYER receives the *TagOut* command it will translate and relay that message to the LiFKIM. If, within the previous 2 seconds the LiFKIM has received a *TagDetected* message for the same location as the *TagOut* command, it will shut down the lift fan, begin recuperating and return a message indicating its status as On Deck.

10. If, at any one time, a team has more than 3 PLAYERS active, the match will be forfeit.

11. If, during play, the ball enters and remains in the On-Deck region, a referee will move the ball to the center of the field to allow play to continue.

12. PLAYERs must remain hovering except when in the On-Deck regions.

13. Players may not hold the ball.

The Playing Field:

14. The playing area will be a rectangular area 25'x10'.

15. The outer perimeter will be bounded by wooden walls 1.5" high with plastic 'chicken-wire' walls to a height of 2' to keep the ball on the field.

16. The Goal Line and Center Line will be marked by 1" black tape.

17. .The Goal openings will be 24" high.

18. The On-Deck regions will be 2'x1.75' each on either side of the Goals.

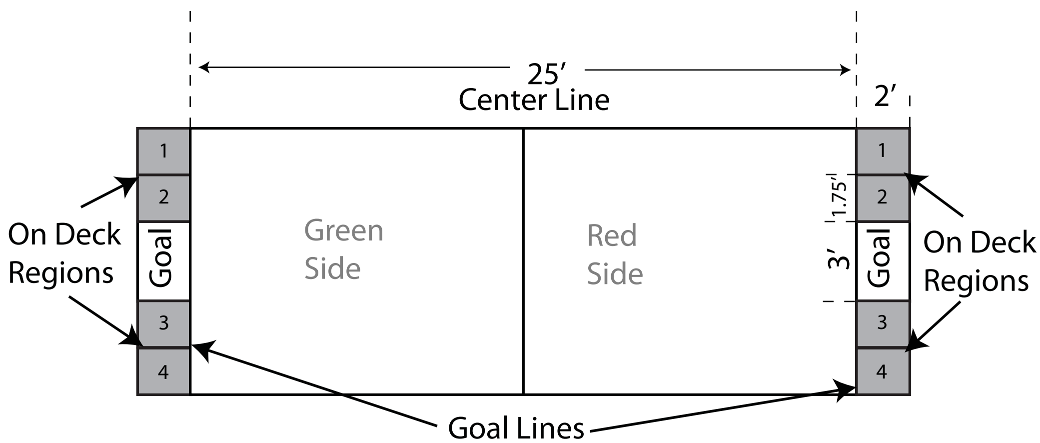

Fig 1 Playing Field

The PLAYERs:

19. Each PLAYER must be capable of moving under its own power around the playing field (described above).

20. PLAYERs must be battery powered by NiCd or NiMH batteries and operate without a tether.

21. Remote control of PLAYER functions must be achieved via a COACH using the provided RF hardware (XBee24 modules).

22. The PLAYER must hover, and the only source of lift allowed is the lift fan provided by the SPDL.

23. Each PLAYER must carry an SPDL supplied Lift Fan Kontroller with Impact Monitor (LiFKIM). The LiFKIM will be powered (from your player) with no more than 15V and will provide the power to control the lift fan. **Do Not connect the lift fan directly to batteries.** The lift fan is not rated for the battery voltage.

24. Propulsion and steering systems are free with the caveat that they may not directly apply force to the ground. An exception to this rule allows for dragging an element against the floor to brake or steer. This will be acceptable as long as the contact mechanism would, if deployed while the PLAYER was not moving, create approximately equal drag in all directions and is not capable of supporting the vehicle.

25. Each PLAYER must provide an electro-mechanical display of its Energy State that will be clearly visible to the operators of the COACHs and the audience. Whimsy in the form of the display is encouraged.

26. The PLAYER must fit entirely into one of the On-Deck regions and be no more than 12" tall at the start of the game..

27. Each PLAYER must provide a mechanical method to 'Kick' the ball on command from the COACH.

28. Each bot must have an impact bumper which sits at a height of 1.5 inches off the playing surface and surrounds the entire perimeter of the bot with gaps no greater than $\frac{1}{2}$". This bumper must be capable of sustaining repeated impacts from other bots.

29. Each PLAYER must wear a "jersey" clearly displaying its number to the COACHs and the audience.

30. The PLAYER must not kick when the LiFKIM reports that the energy level is less than 10%.

31. Each player must have a switch that will specify if it is playing for the Red or Green side in a particular match.

32. The perimeter of the PLAYER must be largely convex. Whether or not a perimeter is largely convex will be determined by moving a straight-edge around the perimeter at the height of the bumper. If the contact between the straight-edge and the perimeter resolves to 2 or more points (rather than a single point), then the maximum distance between the straight-edge and the perimeter of the vehicle must not exceed 1/(distance between contact points).

The COACHs:

33. Each team will design and construct a COACH that will relay commands from a human operator to a PLAYER, and receive and display status information from the PLAYER.

34. The COACH must be capable of displaying to the operator an indication of active communication with its associated PLAYER.

35. The COACH must provide a method for the operator to designate on which side (Red or Green) the COACH is playing and which PLAYER (1-13) the operator would like to control.

36. The COACH must include a display of the energy level of the LiFKIM on the PLAYER to which it is currently connected.

37. COACHs must be battery powered, and shall have sufficient battery capacity for at least 8 hours of continuous operation. The report should show documentation and calculations to support meeting this requirement.

38. COACHs must be untethered, portable by one person and occupy a volume no larger than 2.5'Wx2.5'Dx5'H.

39. Input to the COACH should involve at least 3 sensing modalities (e.g. position, force, audio, acceleration, etc.). Use of unusual interface methods is encouraged.

40. The actions required by the user of the COACH to issue commands to the PLAYER should be inventive and interesting for the audience to watch. Use of actions that make the operator look and feel foolish is encouraged.

41. COACHs should be intuitive to operate, and/or have sufficient visual instructions that a typical spectator (even a non-engineer) would be able to learn its controls within the time span of a single game round.

The Lift Fan Kontroller and Impact Monitor (LiFKIM):

42. The LiFKIM is a self-contained subsystem that will control the power to the lift fan, monitor the energy level of the player and monitor for impacts with other players that might temporarily incapacitate a player.

43. The LiFKIM will have lift control and TagOut commands and Active/On-Deck and Energy status reports that will be available via commands issued over the interface described in the LiFKIM documentation.

44. Communications with the LiFKIM must take place through a PIC16F690 programmed in assembly language.

Radio Communications:

45. Communications between the PLAYERs, and COACHs, will take place over an SPDL-supplied 802.15.4 radio (Xbee24) using the Non-Beacon API mode of operation.

46. Any COACH should be capable of controlling any PLAYER.

47. Once a game begins, communication will take the form of bi-directional communications between a PLAYER and its bound COACH.

48. Each PLAYER and COACH will be assigned a unique ID in the form of the source address of each SPDL-supplied radio.

49. The details of the communications protocol will be defined and specified by a Communications Committee, which will consist of one member from each project team. The specification must be in a written form and with sufficient detail that someone sufficiently skilled in ME218 material could implement it.

50. In order to better balance the workload and learning among team members, each of the following tasks must be completed by a different member of the team: serve on the communications committee, implement communications on the PLAYER, and implement communications on the COACH.

51. The class communications protocol must include a procedure for validation of communication between the PLAYER and COACH. The COACHs must provide a visual indication of when a functioning communications link between the PLAYER and COACH exists.

52. The COACH may issue commands to a PLAYER at a rate no greater than 5 Hz.

53. The PLAYER may issue responses to a COACH at a rate no greater than 5 Hz.

General Requirements:

54. At a minimum, either the COACH or the PLAYER must contain two actively communicating processors. There is no class imposed upper limit on the number of processors employed.

55. The microcontroller that interfaces with the LiFKIM must be programmed entirely in assembly language.

56. You are limited to an expenditure of **$200.00/ team** for all materials and parts used in the construction of your project. Materials from the lab kit or the Cabinet Of Freedom do not count against the limit. All other items count at their fair market value.

57. A project logbook must be maintained for each group. An on-line blog is appropriate to meet this requirement as long as it is made available to the teaching staff for review. This book should reflect the current state of the project, planning for the future, results of meetings, designs as they evolve etc. The project logbook will be collected at irregular intervals for evaluation.

58. A report describing the technical details of the system will be required. The report should be of sufficient detail that a person skilled at the level of ME218c could understand, reproduce, and modify the design. The report must be in website format, and be suitable for posting on the SPDL site.

59. PLAYERs based substantially on purchased vehicle platforms are not allowed.

60. All projects must respect the spirit of the rules. If your team is considering something that **may** violate the spirit of the rules, you must consult a member of the teaching staff.

Safety:

61. Both the PLAYERs and the COACHs should be safe, both to the user and the spectators.

62. No part of the PLAYER or COACH may become ballistic.

63. The teaching staff reserves the right to disqualify any device considered unsafe.

Check-Points

Design Review:

On **05/06/14** between 10am & 3:00pm we will conduct a design review, one team at a time. Each team should prepare a few images showing your proposed designs for both the PLAYER and the COACH. You will have 5 minutes to walk us through your ideas. The focus should be on system level concepts, **not detailed hardware or software**. We will spend the balance of the time-slot giving feedback and asking questions. You will present these via a Skype or Google Hangout (TBD) with Ed & the TAs.

First Draft of Communications Standard:

Due by 5:00 pm on **05/07/14**. Ed will provide feedback on the specification as input to the second draft.

Communications Standard:

Due by 5:00 pm on **05/09/14**. This is the working draft of the communications standard.

First Check-Point:

On **05/13/14**, you must demonstrate

1) The ability of the PLAYER to receive and correctly decode and respond to commands from the COACH (simulated inputs are acceptable at this time).

2) That your PLAYER platform has been built, and is capable of bearing the approximate weight of all the necessary components it will carry when complete. It is encouraged, but not required, to demonstrate working propulsion and steering subsystems.

3) The ability to communicate with the LiFKIM and report the LiFKIM status.

The final working version of the communications standard is due. No further changes are allowed to the standard. This protocol will be evaluated with respect to its completeness and suitability for the proposed system. **Note:** this is a functional evaluation only. The focus should be on demonstrating **functional** hardware and software.

Second Check-Point:

On **05/20/14**, you must demonstrate the ability to communicate all required functionality between your PLAYER and COACH. This will include commands from the COACH to the PLAYER and all status messages from the PLAYER to the COACH.

Project Preview:

At the Project Preview on **05/23/14**, each team must demonstrate (in addition to the 1st & 2nd check-points' functionality)

1) The ability to successfully send and execute the drive, steering and kicking commands (including the actuation of all electromechanical outputs) from an operator of the COACH to the PLAYER.

Grading Session:

During the Grading Session on **05/27/14**, each team will be required to demonstrate the ability to successfully participate in a game. This will include

1) Establishing communications between your PLAYER and COACH, between your PLAYER and the COACH from another team and between your COACH and a PLAYER from another team. Successful communication will require the display of PLAYER status information and the correct response to controls.

2) Demonstrating the ability to communicate with the LiFKIM and display the Energy Status on both the PLAYER and the COACH.

3) Navigating a PLAYER from the initial position and successfully kicking the ball into a goal.

4) Successfully perform a 'Tag Out' operation and take control of another On Deck PLAYER.

Public Presentation:

This will take place on **05/28/14** starting at 6:00 pm in the Peterson Atrium. At this event, members of the public will be allowed to act as operators of the COACHs.

Report:

Draft due on **06/02/14** by 4:00 pm. The final version (with revisions incorporated) is due by 5:00 pm on **06/06/14**.

Evaluation

Performance Testing Procedures:

One or more of the team members will demonstrate the PLAYER and COACH during the first & second check points and project preview. Members of the teaching team will operate the PLAYER and COACH during the grading session.

Grading Criteria:

- ☐ **Concept (15%)** This will be based on the technical merit of the design and coding for the machine. Included in this grade will be evaluation of the appropriateness of the solution, as well as innovative hardware, software and use of physical principles in the solution.
- ☐ **Implementation (15%)** This will be based on the prototype displayed at the evaluation session. Included in this grade will be evaluation of the physical appearance of the prototype and quality of construction. We will not presume to judge true aesthetics, but will concentrate on craftsmanship and finished appearance.
- ☐ **First Check Point (10%)** Based on the results of the performance demonstrated on 05/13/14**.**
- ☐ **Second Check Point (10%)** Based on the results of the performance demonstrated on 05/20/14**.**
- ☐ **Preliminary Performance (10%)** Based on the results of the performance demonstrated during the Project Preview.
- ☐ **Performance (15%)** Based on the results of the performance testing during the Grading Session.
- ☐ **Report (10%)** This will be based on an evaluation of the report. It will be judged on clarity of explanations, completeness and appropriateness of the documentation.
- ☐ **Report Review (5%)** These points will be awarded based on the thoroughness of your review of your partner team's report. Read the explanations, do they make sense? Review the circuits, do they look like they should work?
- ☐ **Log Book (5%)** This will be evaluated by the evidence of consistent maintenance as well as the quality and relevance of the material in the log book.
- ☐ **Housekeeping (5%)** Based on the timely return of SPDL components, cleanliness of group workstations as well as the overall cleanliness of the lab. No grades will be recorded for teams who have not returned all loaned materials.

Gems of Wisdom from Prior Generations

- Get the radio working with the 'E128 first.
- Do not continue working until the wee hours of the morning unless you absolutely have to because errors propagate when tired. A fresh look at things in the morning will save you a lot of pain at night. Sleep is not a crutch, it is a necessity.
- Put some time into your first prototype. You might be surprised how many things you throw together for testing purposes make it into your final project.
- Label or color-code your connectors so that it's easy to plug them into the right place. Connectors that can only be hooked up one way (such as Molex) prevent undesirable incidents like reversing the voltage and ground connections and frying components in the process.
- When building networks, add nodes one at a time to better track down "bad nodes".
- Debugging LEDs are useful for getting feedback on the operational state of PICs.
- A "power central" board is a good thing to have, particularly if you're dealing with multiple supply voltages. This makes the circuitry cleaner, and can save you from supplying your PIC with 37 volts.

- Think twice before planning to provide PWM for motors with PICs (at least the one with 20 pins). You will need to take care of output compares and timers also.
- You will need to leave some pins on PICs (especially those with only 20 pins) open for debugging.
- Don't hesitate to add another PIC and SPI communication. It's really easy.
- Using shift registers for debugging can also be a helpful trick to obtain more information, but it is not good for timing issues.
- Try working during the day (seriously!). Debugging is way easier with a clear head.

- The radio boards runs on 3.3V not 5V. Design your circuits accordingly and be ready to convert between the two voltages.
- Jameco is NOT OPEN on weekends. Don't postpone your trip until Saturday – you will be sorely disappointed.
- Don't be dead set on a theme at the beginning of the project. Let the project theme develop as you move through the project. You'll be surprised how many great ideas pop up as you go along.
- Hot glue down all soldered wire connections. You'll lose a lot of time tracking down an error that may end up being a loose/broken wire.
- Write all functions as non-blocking code – no matter where they fit into the flow of the program.
- Just because two points on a circuit look like ground when probed doesn't mean they are connected.

- Test circuit as it will be implemented in final form, as well as fully integrated.
- Test in environment in which hardware will be used (radios outside, with appropriate distant in between).
- Testing our radio pair in the presence of other active radio pairs revealed problems that didn't exist when we test alone.
- It's easy to make a design with bad ergonomics which make it impossible for the user to perform the task. Prototype/try out the user scenario yourself as early as possible.
- Keep circuit diagrams up to date as you make them.
- Use lab notebook so that all information is at one place and teammates can have easy access to it.
- Take a lot of pictures as you go.
- Remember to HAVE FUN.

- If you are having intermittent problems (e.g. it works only some of the time) check your connections – eLiFKIMly those connecting your various circuits to a common ground.
- Modularize as much as possible – test all of the components separately before integrating
- Build and test all of your circuits and sensors on breadboard before you make them hard mounted on perfboard.
- When moving your circuits from breadboard to perfboard, rather than dismantling your breadboards, leave your working breadboards intact and buy new components and build entirely new circuits on the perfboard. That way, if something goes wrong once everything is built, you will always have a backup copy of your circuits on the breadboard that you know worked before you integrated everything.
- Isolate your circuits onto individual perf-boards (rather than having a giant perf-board with all of your circuits). Makes it much easier to take them out to debug them.
- A nice pair of wire snips (flush cutters) and wire strippers makes wirewrapping and circuit building in general much easier.
- Do your circuit calculations to make sure you have enough/not too much voltage/current/power
- Have plenty of spare parts ready to go in case something blows at the last minute
- Always take the time to test on the actual competition field at the actual competition location
- Make sure at least two people of the group understand or at least have an idea of each component – mechanical, electrical, software. Doesn't have to be the same two people, but it insures that if someone's missing, that the group isn't stuck.
- Be friendly with other teams – you never know when you're going to need help.

- Test your wireless communication outside and at range!
- Test your components for interoperability with everyone else's before game day.
- Source an off-the-shelf housing/controller and gut it. That way you can focus on the electronics and not the mechanical design.
- If you can avoid having to spend time building something by buying an equivalent part, do it!
- Remember your banksel commands and save yourself hours of debugging.
- Make things accessible (i.e. batteries, boards, DIP sockets, etc.) so you don't have to unscrew things when you need to test, power cycle, or reset things.
- Learn to use assembler macros. They clean up your code visually.
- Don't use macros when you can use a function-type subroutine!

- Size does matter. The bigger or larger the motions involved in your controller, the more entertaining it will be to watch.
- Do not bury your wireless antenna in a box. Try to keep it out in the open.
- Buy a good pair of wire strippers, preferably ones that can strip 30AWG wire. Your fingers will thank you.
- Try to avoid having to scramble together parts or code for a check-off. This means keeping on top of the project schedule. If you don't, you will end up throwing away a lot of hours on setups that will not make it to your final design.
- PICs are notoriously difficult to debug. Either source an MPLAB ICD2 (or equivalent), or finish your circuitry early, before writing the bulk of your code. You will find that changing even small things in assembler can cost you hours.

- If a change causes things not to work the first thing you should check is if the code is in the correct bank. It is always a good idea to use a bank select command at the start of every routine rather than assume you'll know where it is.
- Build and test the code in small manageable pieces. If a lot of changes are made at once and the new program doesn't work, it is very hard to isolate the problem without a lot of work.
- Use the debugger. Running routines through the debugger to see what will happen will save lots of time and effort. Getting a routine to work in the debugger usually allows you to assume problems that come up in actual testing are hardware rather than software related.
- When you're tired and everything starts to fail don't forget to check the batteries.
- If you're tired and everything starts to fail and it's not the battery consider going home and looking at it again the next morning rather than changing a lot of code. Often it is some small little change you overlooked and are too tired to notice.
- Make sure all data tables are in the correct location.
- Make use of calls and macros whenever possible to keep the code clean. This also makes repetitive actions easier to code and change.
- Make use of #defines for labeling pins and value as much as possible. This makes it very easy to see what pins are connected to what and allows for the easiest changes. Rather than searching for a specific port and pin throughout the code you only have to change one #define value.

- Don't believe anyone that tells you that the 218C project is less time consuming than the 218b project. It's not.
- Pick your battles early. Learning to program the PIC's and the Zigbees is a lot of work on its own. Trying to add other challenges can be tough.
- Move to solder boards or wire wrap boards as soon as you can. If you are developing simple hardware that you understand well, don't be afraid to solder it on a board. Troubleshooting bad connections on a breadboard is a waste of your time.
- Allocate your pins and subsystems early. A spreadsheet that shows all of your pins is very handy.
- Practice on the course as soon as possible to test your operable range.

- PICs are apparently not designed to be inserted backwards. We recommend against doing this.
- Don't spend more than a few hours debugging SPI code before debugging all of the related hardware.
- Start by making a schedule for the project and include any outside events like vacations, graduations, etc. to avoid surprises later on.
- Black objects left in the sun tend to melt any hot glue that is exposed. This is detrimental to the project's structural integrity. It is therefore wise to a) avoid hotglue or more realistically, b) avoid leaving black hotglued objects in the full sun for extended periods of time.

- Although checkpoints are important, the key is to continue working on the final product, so at all times try to write code/build hardware that you will be able to use in the final product. Try to minimize writing special "check-off code" and building "check-off hardware" that you won't use later.
- Thinking very carefully about your electrical design/layout will save you lots of soldering time. By designing carefully, you'll optimize locations of every components, and you'll end up making a lot less solder joints/connectors/electrical boards, fewer corrections.
- Use Debugging Leds if using PICs. Reserve a few outputs so you can toggle the bits and see if you get into loops or states. This was really helpful when we were trying to figure out what was wrong with our code. Also, since we already used the SSP and Asynchronous communications outputs, we could not use printfs to the terminal.
- Check your #defines and labels. With PIC programming, you tend to have a lot of GOTOs and CALLs which means you need a lot of labels. Try to have a good system for labeling things and creating your constants and variables. We used CAP_UNDERSCORE for # defines and FirstCapitalLetter with no spaces for variables. Where we went wrong was creating "FORWARD" and "FORWARD_CMD" which we misinterpreted and messed us up for a long time.
- Talk to other people about the communication protocols and how they implement their code. It's hard to figure out the datasheets by yourself with no help from anyone.
- Debug code extensively prior to integration with other software/hardware elements.
- Utilize 7-segment display or LCD display for real-time debugging.

- Modularize mechanical systems so that simpler parts can be made earlier and used from the beginning of development.
- Develop a clear understanding of the communications protocol from the beginning of code development.
- Communicate well within your team so that some tasks are not overlooked, while others are duplicated.
- Learn some of the common problems with writing PIC code (such as dividing your long code into small sections using the "Maincode code" command).
- Bathe as frequently as possible; encourage others to do so as well.