FIRST Robotics Team 3667/PoHo RoBos
Port Huron Area School District 21st CCLC
Harrison Center
55 15th Street, Room 115
Port Huron, MI 48060

Figure It Out!

Fall 2011 Development Plan (Draft #2)

Goal: To develop (or improve) knowledge and skills. Includes both technology that may be of use in the 2012 competition (programming, sensors, materials, construction, tool skills, etc.) and operational that will benefit the team year-round. (shop, team management, logistics, fund-raising, marketing, etc.)

This is a *preliminary* list, and needs to be prioritized. We probably won't be able to do all (or perhaps even half) of it. But if something on the list jumps out at a mentor or student, let's do it!

1 Programming

1.1 Vision Recognition

THIS JUST IN (10/6/11): This year's KoP will include a Microsoft Kinect!!! See here: http://youtu.be/hlgA2nlXUF8.

Task: Learn as much as we can about the Kinect.

Task: Using the web cam and the reflective tape on the pole & pegs, automatically adjust the arm to the correct position to hang a tube. (Hint: Use the NI Vision Assistant…)

Task: Program the robot to autonomously hang a tube on any peg from a distance of, say, five feet from the pole.

Task: Program the robot to autonomously pick up (via vision recognition) a tube located on the ground in a known position and hang it. Involves recognizing the shape of the tube, maneuvering into position, rotating the arm down, etc.

1.2 Line Following

A particularly devious idea from Bruce:
Task: Have the kids reprogram the robot to follow a tape strip anywhere it goes, including circles. This will require a different strategy for steering than following a straight line of tape. The kids could work together, or compete with each other to see who can follow a path the fastest without leaving it. This might be a good thing for them to work on, then I come and comment on their code. The goal would be to give them more programming experience while we have plenty of time to discuss and rewrite it. You could add a requirement that the robot stop and the arm does something if the line ends at a T. To make it more difficult, you could allow the path to cross itself. They wouldn't know if it was a T or a cross until they get to the far side of the intersection.

1.3 Control Systems

The 2012 FRC beta tester documentation (http://forums.usfirst.org/showthread.php?t=18044) emphasizes that teams selected as beta test sites must (paraphrasing) "must agree to become control system experts and serve as area leaders to mentor other teams" and that they must host at least one Saturday controls training seminar or open house for all neighboring teams prior to the 2012 kickoff".

Task: Designate team members to attend one of these training seminars.

1.4 Mindstorm

From Chief Delphi and comments that Dean has made re: getting the FLL/FTC/FRC teams to work together, it sounds like mini-bots are here to stay, and future games will probably require them to be "smart", possibly in-play throughout the game (and entirely autonomous) – i.e. must include a Mindstorm NXT to be of any use at all. Additionally, the Mindstorm is a great platform for developing skills (can be programmed w/ LabView, use of sensors, autonomous algorithms, etc.) because they're cheap, so multiple people can be working on separate projects concurrently, even at home. So… we need to grow in this area. To that end…

Task: Use LabView to program the Lego Mindstorm NXT. Install and configure the "NI LabView Toolkit for Lego Mindstorm NXT".

Task: Teach entry-level programmers to use LabView to program the Mindstorm NXT. See http://bit.ly/ppmyCi for a training curriculum developed by team 3215.

Task: Construct a simple Mindstorm robot and program it to follow the perimeter of an arbitrary room (like a classroom, with chairs, tables, file cabinets, wires laying across the floor, etc.). Think a Roomba vacuum cleaner…

Task: Program a simple Mindstorm robot to follow arbitrary lines on the floor, such as those in the gym. Use any available Mindstorm sensors (distance, color, etc.). Consider what would be involved with programming a robot to follow a maze… Hint: There are probably numerous doctoral thesis' written on the subject…

Task: Figure out how to use the (cheap, readily available) Lego Mindstorm-compatible sensors with the cRIO. Research Mindstorm-compatible sensors from Lego, LogIT, HiTechnic, Vernier.

1.5 Software

LabView

On Chief Delphi, an "NI employee in the academic group" basically stated that we're getting an updated version of LabView this year. So…

Task: Install LabView 2011 (per a comment from a FRC 2012 KoP beta tester), port our 2011 robot software to it, and prove that it can be deployed to the robot and function correctly.

Autodesk

Goal: Improve our robot design practices. Do more engineering and less guessing.

Task: Become skilled enough with 3D modeling using Autodesk software (Which one(s)? I dunno. Figure it out!) to make it of benefit in the preliminary design of the robot or components thereof. Be able to quickly test alternative designs/configurations and answer questions such as: Are we within the height/width constraints? Will the arm interfere with the chassis? Will the mini-bot deployment arm reach the pole?

1.6 Languages & Libraries

Task: Research other languages available for programming the cRIO. Determine if there are any functional areas where one is significantly better than the others. Chief Delphi has numerous threads on this question.

Goal: Gain enough knowledge of what's available in public-domain robotics libraries to be able to recognize (during build season) that something that we need has already been written.
Task: Learn about libraries available in the public-domain from sources such as WPI (http://thinktank.wpi.edu and http://firstforge.wpi.edu), Carnegie-Mellon, SourceForge, etc. Are there any features that would be really cool to use?

2 Mechanics

2.1 Pneumatics

Task: Configure a pneumatic system on the bench. Include all safety precautions & restrictions specified for the 2011 game. Program the cRIO to actuate the solenoid when a button is pushed.
Issue: We don't have any actuators. Can we learn anything without them? If not, can we procure some for cheap/free, even if they're not suitable for use on a robot (too heavy, etc.)?

3 Controls & Sensors

3.1 General

Goal: Reduce the frequency (and associated time) of rebuilding/deploying the code from the development machine.

Task: Implement a system that reads variables that are changed occasionally/frequently from a file stored on the cRIO (e.g. a .INI file, .XML file, etc.). The file could be uploaded via FTP when changes were needed. Going into Disable mode would cause the file to be re-read at the next Enable. If we had controls on the DS (sliders, typing in numbers, etc.) to change the variables while the code is running (e.g. for adjusting limit points on the encoders), figure out a way to save all the variables to the file, which would persist across reboots. See http://www.chiefdelphi.com/forums/showthread.php?t=97531.

Task: Design and implement a system of "DIP switches" that are used to configure the cRIO and its software to run in various operational "modes" or "constraints". These could be binary switches, or variable (e.g. potentiometer). Examples: Enable/disable autonomous (we had an issue at Kettering Kickoff where we needed to completely disable autonomous mode, but didn't have time to change the software, download it, etc. As a result, we ran into an alliance partner and caused them to not be able to hang an ubertube.); Speed governor (100%, 75%, 50%, etc. would have been helpful/safer when doing demos at schools). Modes could also be used to do A-B testing of different algorithms/implementations or combinations thereof, completely disabling certain features that are known to be buggy while allowing us to test other features, etc.

Task: Incorporate the existing wheel rotation/position sensors into the code. Display the position and speed on the driver's station. Implement a speed governor based on drive wheel rotation speed, controlled by DIP switches, above.

Task: Figure out how to simulate a servo with a high-powered motor and position sensor(s) and/or limit switches. E.g. "Set the arm to XXX position as quickly as possible regardless of how much power it takes." (Can't recall the exact name that Michael B. used to describe this kind of system.) Would CAN bus help with this (see below)?

Task: Implement a system that absolutely, positively resets everything to exactly the same state that exists when you cold-boot the robot on the competition field – without requiring the 2-3 minutes that it takes to do a cold boot. E.g. a (well protected) push-button or momentary-throw switch on the robot that returns all actuators and end-effectors to their "home" position, resets sensors (rotation, position), resets all global variables, etc.

Task: Connect a servo to the robot and program the cRIO to operate it.

Task: Fix (or find a workaround for) the BUG which causes the arm to slam up or down when the joystick is released.

Bruce's Ideas

Learn how to use every sensor and tool that we have. CAN, gyro, accelerometer, vision, others??? What do we have drivers for in the LabVIEW code they give us?

Should know how to write a program that uses the sensor effectively, and know practical applications of the sensor.

Learn and use feedback-based controls. PID is good and simple to use. That would solve the arm positioning problems. I wanted to work on that, but we didn't have time.

> SDA: For arm preset position calibration, would it be possible to have "Save Preset #1" (2, 3, etc.) buttons in the Driver Station interface that would save the value to a file that would be persistent across reboots?

FYI, you can put a lot of controls/limits/switches into the code and control them from the software interface. Doesn't look too difficult, but never had time to figure it out.

3.2 CAN Bus

Task: Get CAN bus working! There are advantages to it, and it'll be to our advantage to know how to get it to work correctly/reliably. It's entirely possible that future KoP items will require CAN bus to be useful.

Task: Learn the conditions under which a Jaguar controlled via CAN bus using voltage-, speed-, position-, or current-based closed-loop feedback would be useful. (e.g. In DC motors, current is directly proportional to torque, so…).

Task: Configure a Jaguar on a CAN bus in closed-loop mode where the encoder is connected directly to A) the Jaguar, or B) the cRIO. Why would we use one over the other?

4 Utilities

Task: Design and construct a method to operate a motor controller or servo on a bench without needing the cRIO & driver station. There are times where it would be handy to be able to control a motor w/o all the overhead, such as when separately working on an actuator assembly. (And manually touching a wire to a battery isn't sufficient/safe!!!) This could be a completely analog system (breadboard from Radio Shack w/ a pot to vary speed or position), or could be made using an off-the-shelf R/C model radio & receiver (which outputs PCM). Create adapter cables to connect the R/C receiver to a Jaguar/VEX motor controller.

5 Speed/Quickness

Our robot was fast enough across the field. Where we were way too slow was in: A) picking up tubes, and B) hanging tubes – once the robot was within 5' of the tube (to pick it up) or the peg (to hang it) it took too long for the drivers complete the maneuver.
Task: I dunno… Suggestions???

6 Pit

Task: Design our pit, including "a place for everything and everything in its place". Needs to be detailed enough to build (i.e. dimensioned drawings). Take into account functional needs (breaking down to fit through doors and on bus). Extra credit: Create a 3D model of the pit in AutoCAD.

Task: Created a detailed checklist of everything that needs to come with us to a competition. Should be detailed enough that we can say "if it's not on the list, it doesn't go on the bus". Break the list down into categories, such as: stuff used when on the competition field (robot, batteries, driver station, cart, etc.); spare parts for repairs; spare parts and raw materials for possible on-site design changes; promotional items (signs, buttons, etc.); food; tools.

Task: From the above checklist, pare it down to what we need to take with us when we travel locally with the robot, such as to Relay for Life or demonstrations at schools.

6.1 Batteries

Task: Obtain in-depth knowledge of the chemistry & physics of rechargeable batteries. Learn the different charge/discharge characteristics of lead-acid vs. NiMHi vs. Lithium batteries.

Task: Determine the maximum safe charge rate that we can use. (The charger we have defaults to 2A, but take forever to fully charge a dead battery. Can we safely use the 4A or 6A settings? If so, does it shorten ultimate battery life? Will need to know this for each specific brand/model of battery – they may be different. What about a "smart" charger, that varies current and/or voltage based on battery state of charge?)

Task: Design and implement a system for tracking battery use. At all times, be able to answer the following questions: When was the battery last fully charged? How many charge/discharge cycles has it been through? In what competition rounds was each battery used? When was the battery last removed from the robot?

Task: Design and build (or spec and buy) a tool for accurately measuring the usable capacity left in a partially-discharged battery. Hint: Must test the battery under load.

Task: Calculate the total battery system supply (batteries + chargers) that would be required to compete for two full days (including practice and championship rounds). Take into account the tradeoff between number of batteries vs. number of chargers (which requires knowledge of maximum charge rate). Calculate the most cost-effective mix of batteries vs. chargers.

7 Shop

7.1 Tools

Goal: "Certification" of students to use power tools.
Task: Come up with a specific training program/curriculum/testing for each of the major power tools (Drill press, belt sander, grinder, band saw, etc.). Thought: The Boy Scouts produce books to follow for merit badge qualifications. These may be a perfect starting point, or entirely sufficient as-is.

7.2 Physical space

Task: Organize the tools. Create drawings and/or labels showing where each item is supposed to go.

Task: Ditto for raw materials, spare parts, office supplies, etc.

Task: Reorganize the shop so that construction activities are separate from assembly and electronics. Maybe we need wall partitions…

Task: Create a permanent place for the safety glasses at the shop entry door. Make signs so everyone knows you don't come in w/o putting them on. What about a labeled slot or hook for each team member's glasses?

8 Marketing & Team Communication

8.1 General

I need a way to communicate with the team (both students and mentors) that can be/is read by everyone. Based on the response rate to my "please simply reply to this email if you got it" requests and the frequency that I hear "I never got that", it's clear that people are either not getting them or not reading them. I wouldn't be surprised if less than 50% of the students get & read my email.

Task: Come up with an *efficient* method of communicating team info such that we can have confidence that everyone reads it in a timely manner. Maybe it's a matter of education. Maybe messages are being marked as spam. Maybe we need to use Facebook/Twitter. Maybe a phone call tree. Maybe a combination of all of the above is what it will take (though that isn't exactly *efficient*).

8.2 Printed Materials

Task: Create packages of printed materials for various target groups (potential donors/sponsors, existing donors/sponsors, current team member parents, potential team members and their parents).

Task: Create posters for recruiting. Include pictures of current team members. Assign people to hang them around the schools (after getting approval from coach).

8.3 Video & Photography

Goal: Record the team's activities throughout the year so we have an archive of material for use on the web site, award submissions, marketing materials, future recruiting, etc.

Task: Specify the format/resolution that will be used for video & pics. Figure out how to set each camera to this resolution. Make sure that everyone that uses the cameras knows how important it is to use the correct settings.

Task: Create a common repository where all team-related videos & pics are saved. (Saving them on the memory card is not an option!) Specify a backup method.

Task: Learn how to take quality video – steady, good composition

Task: Learn video editing software. Be able to create promotional videos containing text overlays, audio narratives, music, etc.

8.4 Web Site

Task: Determine whether or not to stay with the DotNetNuke-based architecture we have or switch to something new.

Task: If we stick with DNN, pick and purchase a new, cooler looking skin.

Task: Evaluate knowledge level of students that have expressed interest in doing the web site. Is there anyone qualified to improve on what we already have? Anyone that wants to become qualified?

Task: Determine what additional features we should have on the site, if any.

Task: Add a page to the site specifically targeted at potential new members.

8.5 Facebook

Task: Assign someone to own keeping our Facebook page up to date. Coordinate updates made to it with those on the web site.

8.6 YouTube

(http://www.youtube.com/user/FIRSTteam3667).

8.7 Twitter

We're there: @team3667