

SoutheastCon 2012 Hardware Competition

Final Presentation
IEEE Robotics Team
ECE 400



Harrison (Jay) Cross

Computer Engineering
IEEE Robot Team



Final Presentation:

Objectives

Senior Design
Robot Team

My Role

Communication
System Architecture

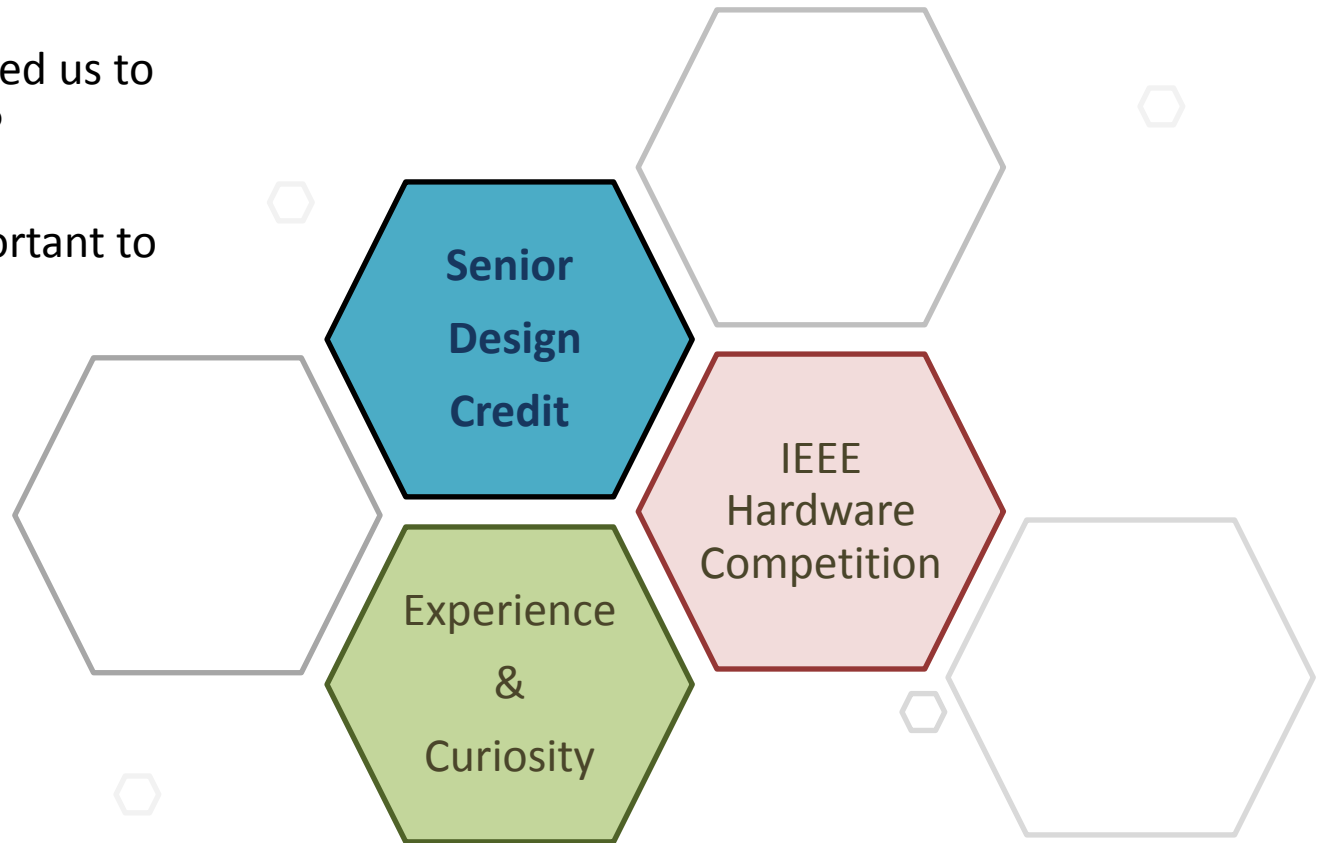
Notes

Things I Tried
Lessons Learned
Future Work

Questions

Objectives:

- What motivated us to build a robot?
- Why is it important to note this?



Senior Design:



According to ABET:

“Engineering design is the process of **devising a system, component, or process to meet desired needs**. It is a **decision-making process** (often iterative), in which the basic sciences, mathematics, and the engineering sciences are applied to convert resources optimally to meet these stated needs.”

“Students must be prepared for engineering practice through a curriculum culminating in a major design experience based on the **knowledge and skills acquired in earlier course work** and incorporating appropriate engineering standards and multiple **realistic constraints**.”

Senior Design:

Dr. Mongi A. Abidi:

Senior Design can be an opportunity for students to really learn and grow in an amazing way. ***

*** (*probably paraphrased poorly*)



Senior Design:

Questions I asked myself early on:

What knowledge / skills will I seek to apply?

What constraints am I dealing with?

How can I make the most of what I have available?

How can I use this opportunity to learn and grow?

Senior Design:

Knowledge I sought to apply / expand:

- microcontroller classes
- control systems class
- programming classes
- analog communications class
- reinforcement learning class
- CE curriculum in general

Constraints:

- budget woes
- time
- inexperience
- building move



Robot Team:



Team Objectives:

- holonomic wheels / movement
- sensor deployment with wireless communication
- robot large enough to carry a smaller robot
- line following
- Did I miss anything?

My Role:



I helped with the following:

- sensor boxes (mainly capacitor sensor, others in a minor way)
- mechanism for the main robot to read each sensor wirelessly
- motors, motor controls, line sensors, battery, and chassis (early on)

Things I tried:

- program interface for sub-components (API)
- simulator program to aid development of line sensor / main
- alternate robot hardware and software drivers

My Role:

In short?

The work I did with the Robot Team falls under these headings:



- Communications
- System Architecture
- Other

Communication:

Wireless – between main robot and deployed sensors:

- each sensor acts as a client to the main robot
- main robot polls each sensor to decide which way to go



Wired – between each of the robot components:

- helped with I²C between Arduinos on competition robot
- used I²C for sensor communication on alternate robot
- programming interface for main controller to sub-controllers
- SoftSerial communication for Arduino-Arduino on alternate robot

Wireless >> XBEE radios:

- low cost (~ \$30 each)
- low power
- ease of use
- met range requirements (course is < 10 feet)
- small form factor (2.438 cm x 2.761 cm)

The need to make further considerations and perform extensive testing was limited.

We were able to proceed very quickly with wireless communication portions of the robot tasks.



XBEE Power:

	ZigBee™ 802.15.4	Bluetooth™ 802.15.1	Wi-Fi™ 802.11b	GPRS/GSM 1XRTT/CDMA
Application Focus	Monitoring & Control	Cable Replacement	Web, Video, Email	WAN, Voice/Data
System Resource	4KB-32KB	250KB+	1MB+	16MB+
Battery Life (days)	100-1000+	1-7	.1-5	1-7
Nodes Per Network	255/65K+	7	30	1,000
Bandwidth (kbps)	20-250	720	11,000+	64-128
Range (meters)	1-75+	1-10+	1-100	1,000+
Key Attributes	Reliable, Low Power, Cost Effective	Cost, Convenience	Speed, Flexibility	Reach, Quality

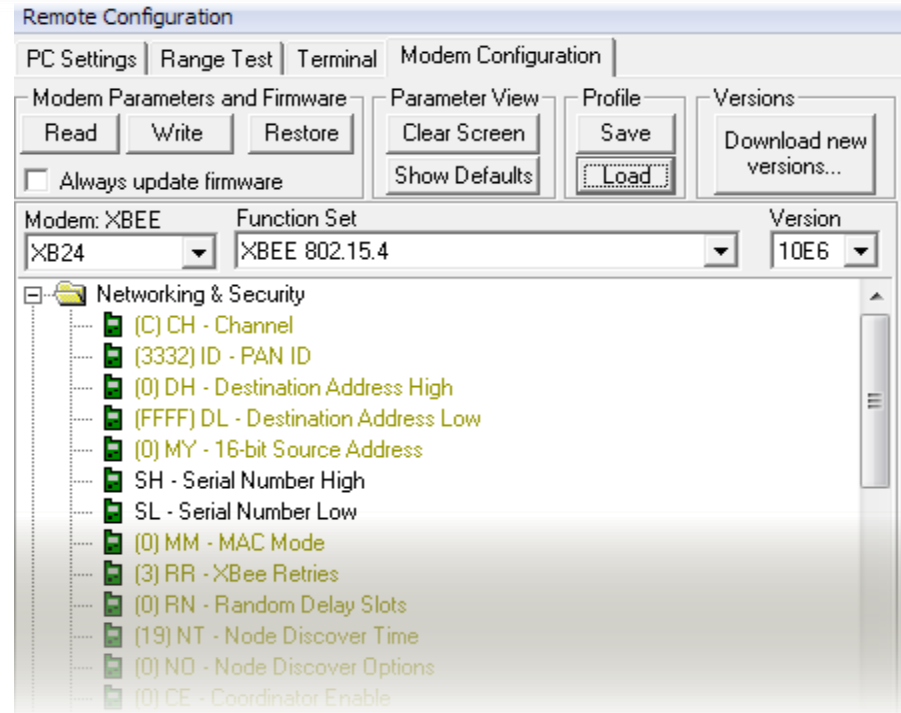
In my tests, I was able to run the XBEE/FIO circuits for about 5 minutes using a small LiPo battery. This was well within our needs.

XBEE Setup:

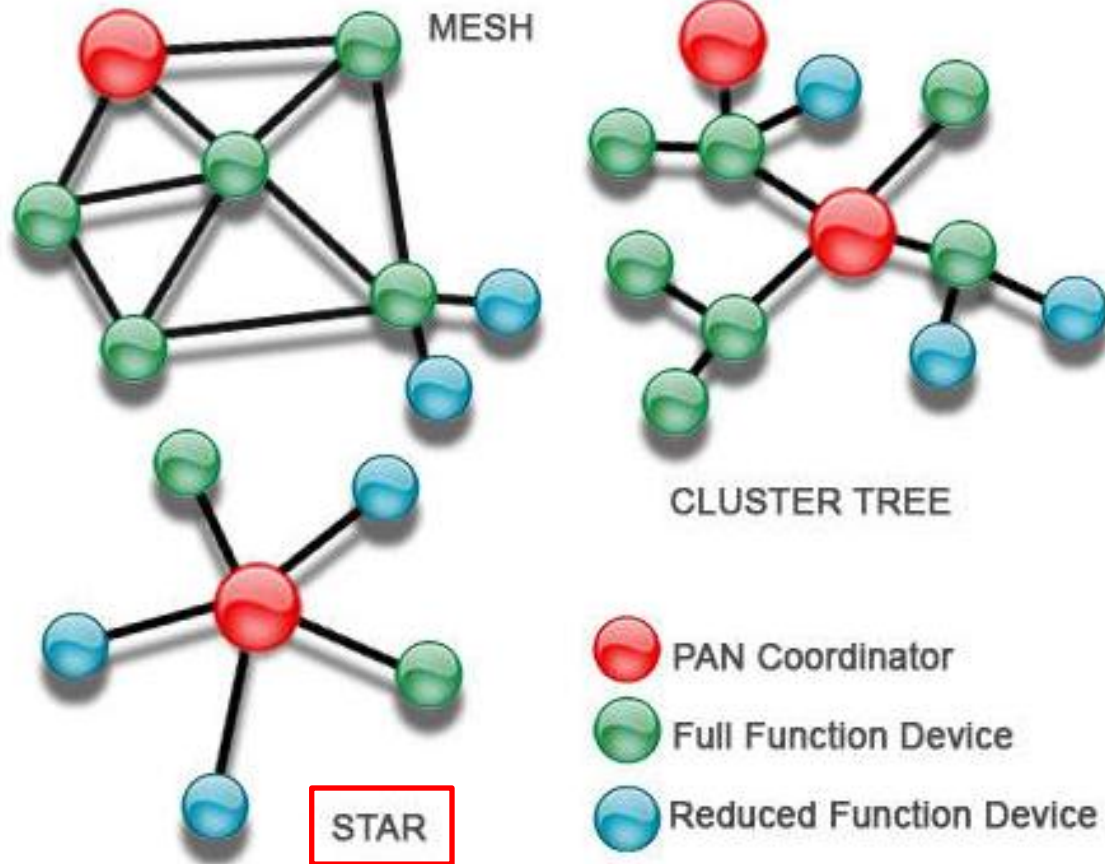
X-CTU configuration program was used to configure the master and slave radios for our sensors.

Channel can be selected and a PAN ID helps to ensure a privacy.

Encryption can be used to further isolate network from the environment. We found no need to use this feature in testing or at competition, however.



XBEE Topologies:



XBEE Range:

Table 1-01. Specifications of the XBee®/XBee-PRO® RF Modules

Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 μ A	< 10 μ A

Wireless – How transfer works:

```
struct TXRX_DATA_STRUCTURE{  
    int task_number;  
    int task_status;  
};
```



All data is exchanged serially over XBEE using Easy Transfer, an Arduino library. This encapsulates the data and provides a checksum to ensure reliable transfer.

Wireless – Implementation:



VS.



XBEE shield conflicts occurred at last minute on main robot so FIO was used connected to the main controller instead.

Alternate robot had no such issues. However, we were unable to program wirelessly in both cases. Reset port issue?

Wired Communication:

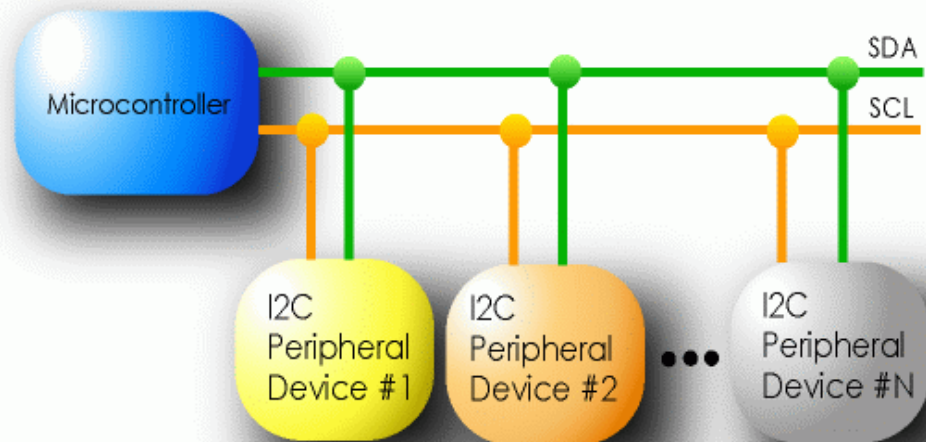
I²C

SoftSerial

Wired communication was used for communicating between the main Arduino and the secondary controllers.

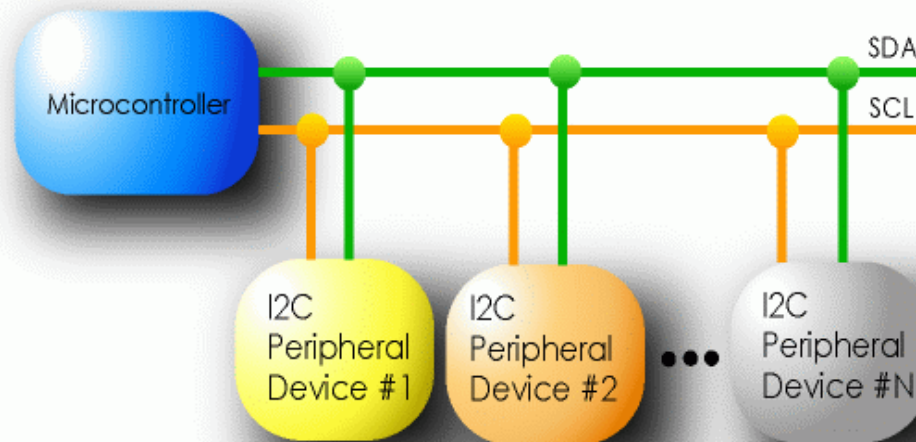
Both robots used I²C at first, I moved the alternate robot to SoftSerial after trouble with robot behavior that is inherent with using I²C for communication between Arduino's.

Wired >> I²C :



Requires 2 wires and a common ground / power source. Uses Arduino's Wire library. Devices can be chained together. Requires devices to be set up as master / slaves with **restrictions on which device can initiate data transfer** – not desirable in some situations. **Blocks on I/O.**

Wired >> I²C :

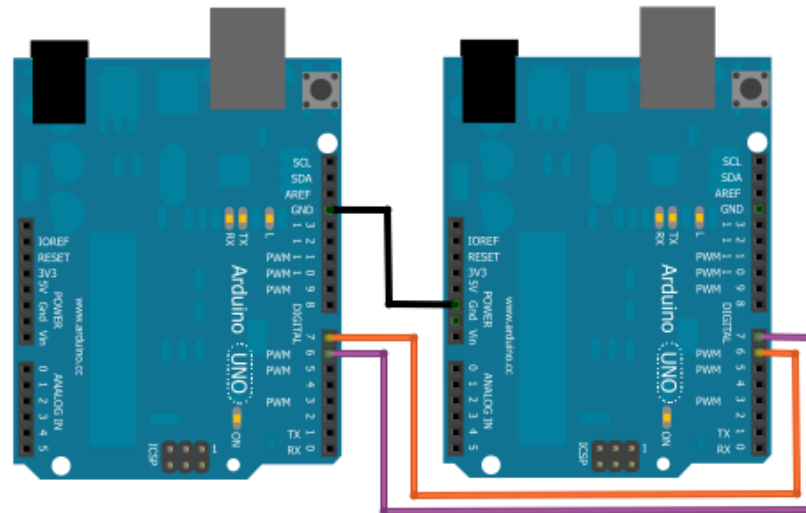


Not particularly suited for Arduino-to-Arduino communications in situations where both controllers must operate concurrently. Better used for sensor.

Fast I²C – 400 kbps vs. 100kbps.

Used this for alternate robot to talk with IMU which is 3 sensors in a chain.

Wired >> SoftSerial:

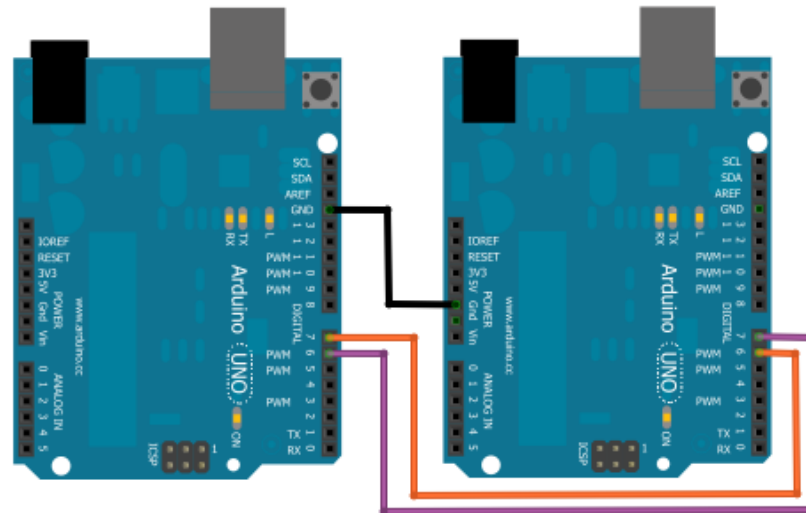


Requires two wires and a ground.

64-byte buffered, 2-way communication. Doesn't block, for all practical purposes.

Capable of running at all Arduino serial baud rates. In practice, I found it better to stick with 57.6K as was used with the FIO wireless communication.

Wired >> SoftSerial:



Used this setup paired with Easy Transfer library for alternate robot when communicating from main controller to secondary controllers A & B.

This required 4 ports on the main controller which was a bit of a sacrifice.

Wired >> Other considerations:

Interrupt communication – this was used to read encoders on alternate robot, but we also considered using interrupts to communicate between Arduinos.

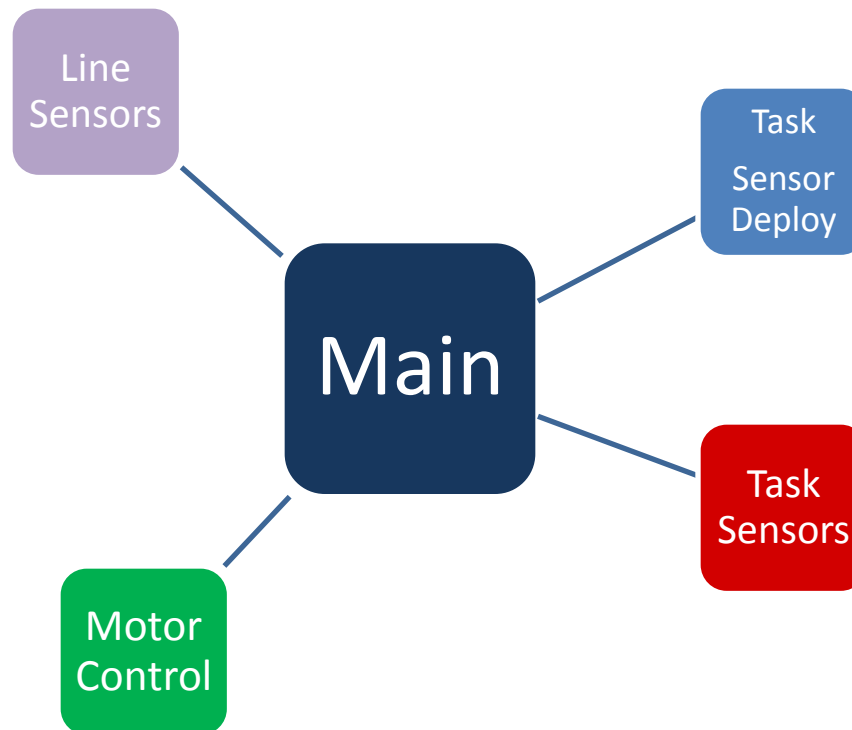
Serial Peripheral Interface (SPI) – uses a master/slave model method for communication.

Universal asynchronous receiver/transmitter (UART) – Arduino libraries are available that are hardware alternatives to the software serial that was used.

These **options were eliminated** due to time constraints, lack of experience on our part, and a somewhat limited amount of information/examples available on Arduino-to-Arduino communication.

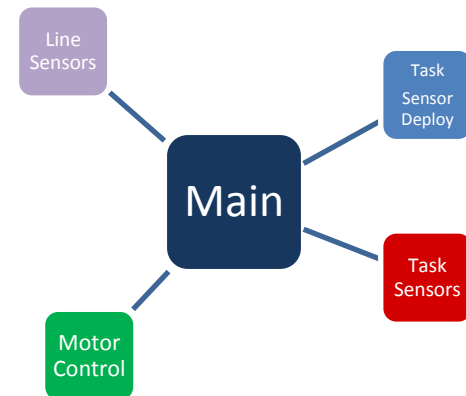
The main **reason other options were considered** were the limitations of I²C as detailed earlier.

System Architecture: Software



System Architecture: Software

PURPOSE: allow each of the components of the robot to be developed separately and independently yet still function as a whole.



ALTERNATIVE: one big file.

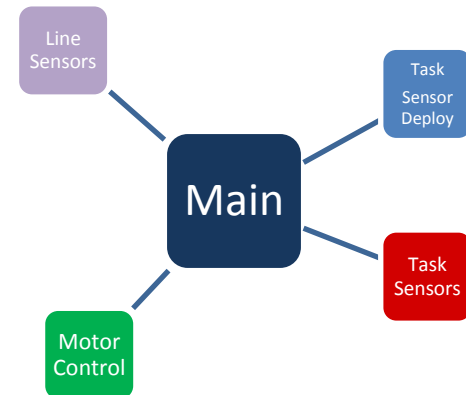
STATUS: the competition robot used a hybrid of the method I created and the alternative. The software structure I created was not used per se, but the same sort of idea was used.

The alternate robot was created using this approach with libraries for each separate robot component.

System Architecture: Software

STATUS (continued): The task sensor code I provided was in library form, but not used directly in code as we used a hardware alternative.

A good part of the **responsibility for not using** this approach on the competition robot was mine. I developed the API to a certain point and then **abandoned the software development aspect** of my System Architecture role in favor of a hardware oriented one.



IMPACT: I doubt software methodology really makes much difference in a hardware context of this scale. The effect of not being entirely well-ordered or modular is/was/can be **negligible** at a small scale.

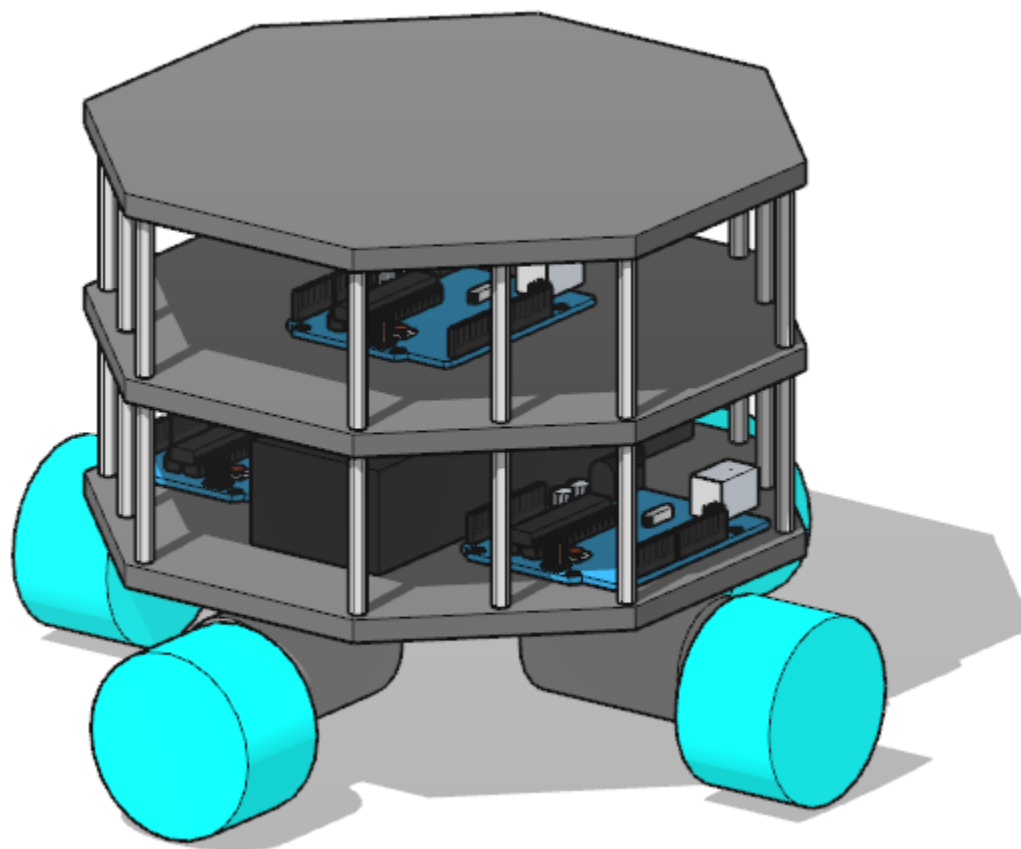
System Architecture: Software



```
class Robot
{
    public:
        char lineData[16];
        int motorData[4];
        char taskData[4];

        Robot(void);
        void start(int baudRate);
        int pollTask(void);
        int pollLine(void);
        int motor(int motor1, int motor2, int motor3, int motor4, int time);
};
```

System Architecture: Hardware

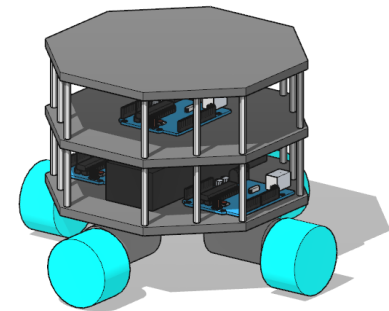


System Architecture: Hardware

Key Points:

Late start on development

Few real differences between Alternate vs. Competition robot



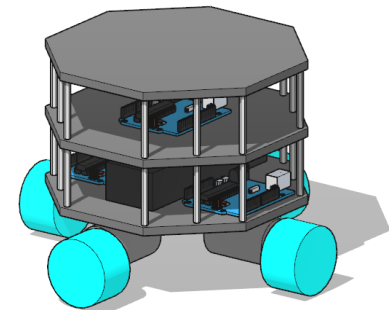
System Architecture: Hardware

Late Start:

I originally did NOT intend to develop hardware as a part of my System Architecture role other than to assist with competition hardware

But...

- I was mostly done with my other tasks
- I had other ideas I wanted to implement
- Competition hardware was “locked in” in ways that were somewhat unchangeable without scrapping the whole robot



System Architecture: Hardware

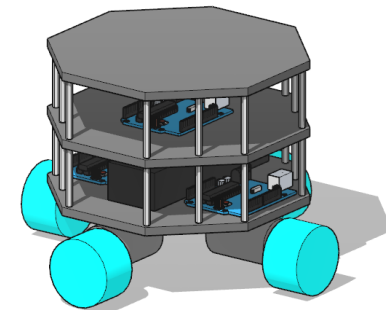
Major differences:

- wheels
- motors
- battery

That seems like everything right?

Well, not really...

still has holonomic movement, still can deploy sensors, still has room for a smaller robot, still uses similar motor control mechanism.



System Architecture: Hardware



vs.



- X or Y without holonomic operation
- can do holonomic (easier)
- mounting is difficult
- cheap, but not on hand

- X / rotate only without holonomic operation
- holonomic (difficult)
- mounting is easy
- more expensive, but on hand

System Architecture: Hardware



1st try:

Applied a hub with epoxy. Was very hard to center hub.



2nd try: two times the size of the first, but the hex bore enabled me to custom build a hub without machining. Extra height enabled me to try out camera.

3rd try: better rolling behavior due to dual rollers. Found a hub made for these after MUCH searching.



System Architecture: Hardware



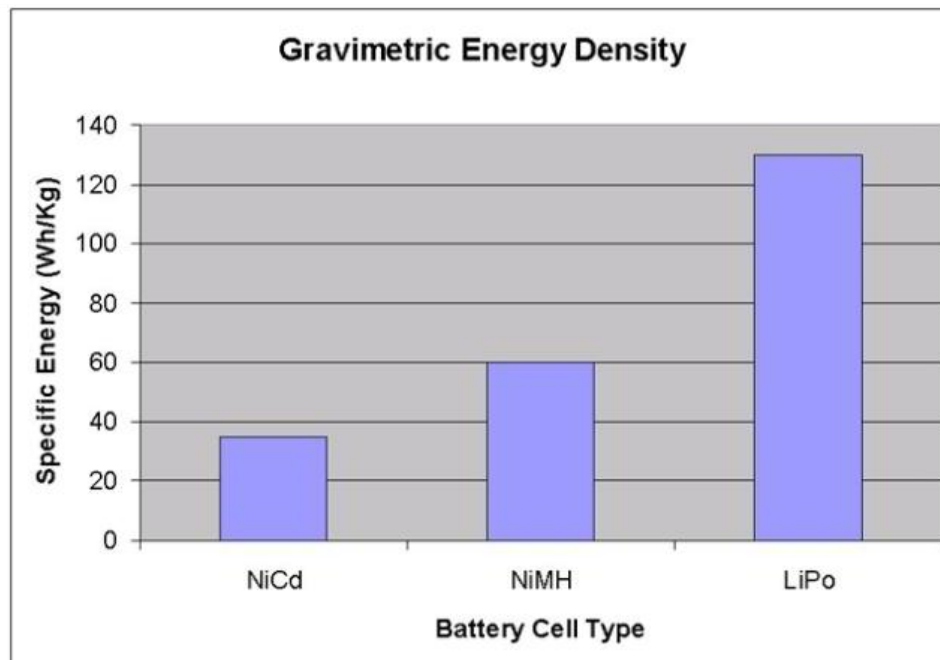
vs.



- Voltage: 12V
- 500 rpm (free run)
- 300 mA (free run)
- 6.1 Kg-cm
- 1.46" x 2.54" (minus shaft)
- \$39.95 (pololu) (not on hand)

- Voltage: 12V
- 120 rpm (free run)
- 74 mA (free run)
- 1.1 Kg-cm
- 1.46" x 1.75" (minus shaft)
- \$14.95 (jameco) (on hand)

System Architecture: Hardware



Comparisons between LiPo and other batteries are numerous. In general, LiPo's **perform better and are lighter but not as cheap**. The competition robot used what batteries were **on hand from previous years**. The alternate robot used an RC battery **purchased for about \$80** at Hobby Town in West Knoxville.



System Architecture: Hardware

Design considerations:

PRICE: All the parts used for the alternate robot were purchased with my own money. I am not rich by any means. I tried to go cheap when possible – no Faulhaber motors. 😞

TIME: I began work on the alternate robot in February. My goal was to have a working robot by competition in mid-March. This meant I went with Arduino's with motor controllers versus building a custom circuit, for example. The same design considerations were made with the competition robot with varying degree.

CHASSIS: Every attempt was made to stay true to the team's design ideas. However, with the chassis, I had to go with a different shape and motor mounting strategy because of the way the wheels worked. This made a big difference in appearance.

System Architecture: Hardware

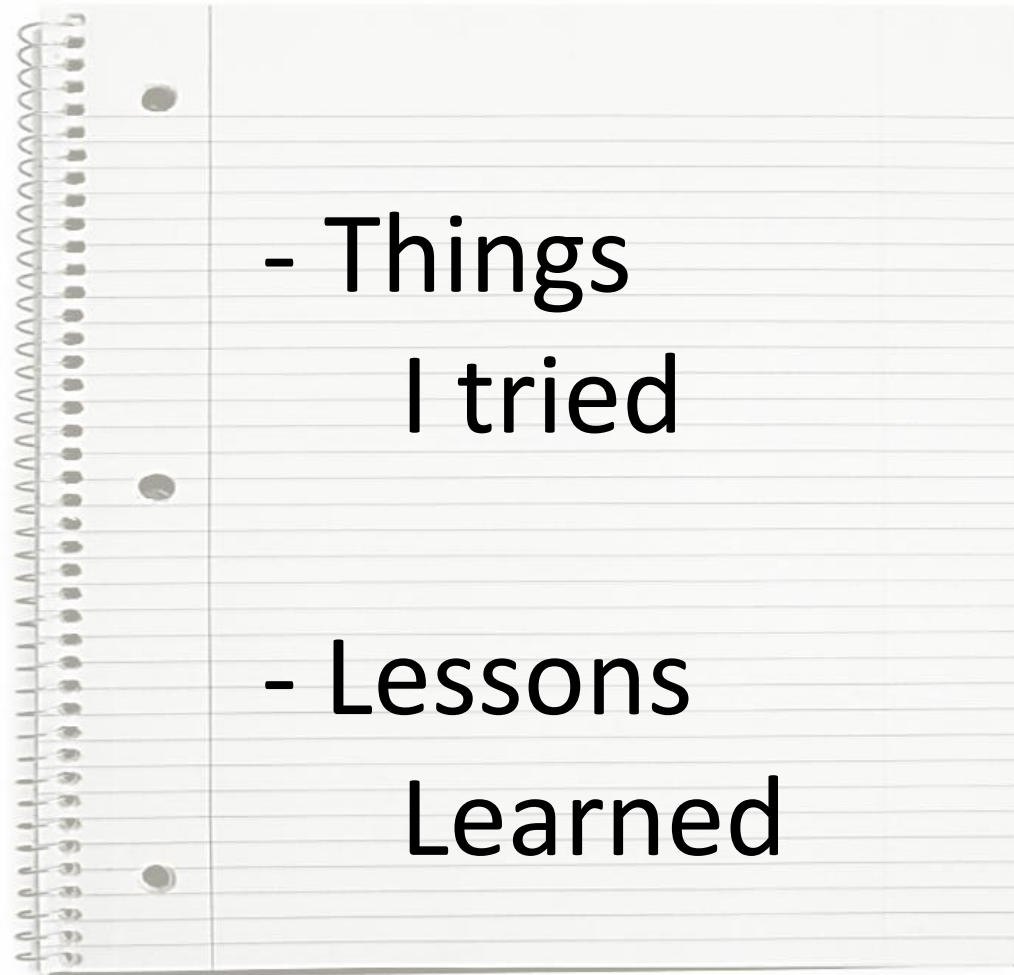
Shortcomings / Hurdles:

I had every intention for the alternate robot to **assume the position as main robot** with the advantage of **better performance**. However, that did not happen because I was **unable to get the hardware ready in time** for the team to contribute, i.e. write software, help me with hardware. Here is why.

WHEEL HUBS: Without centered wheels, the robot would not perform properly. I was unable to find a hub that fit these wheels until too late. I spent much of my effort on this problem and it cost quite a bit of time.

WIRED COMMUNICATION: Tutorials are available online for inter-Arduino communication, but they are not typically very thorough or very prevalent compared to the scores of information for those developing single-Arduino applications. Getting three Arduino's to talk properly was a costly task.

Notes:



Notes >> Things I tried:

SIMULATOR: software designed to accept serial from an Arduino and simulate robot movement. My goal was to provide a means for the main robot code to be written before we were able to complete the hardware.

ROVER: After I saw a demo from Ryan Young, I spent a great deal of time developing a robot using a store-bought toy RC car and a FIO. The major challenge with this robot was its 3.3V power rail. Most sensors require at least 5V.

ALTERNATE ROBOT: While a success from a hardware and associated drivers perspective, it was not so in other aspects. I am interested in using this hardware to explore ideas I had learned about in Control Systems and Reinforcement Learning classes, but I would have preferred to be able to run the course the way the competition robot did.

CAMERA FOR LINE SENSE: Again, successful in implementation, but not ever used for line following.

Notes >> Lessons Learned:

K.I.S.S. : were I to compete again, I would focus on adding simplicity as a feature and avoid difficult to implement tasks, i.e. I would have used a strictly differential drive.

Serial Communication: I knew very little about serial communication before I undertook this project. I have learned quite a bit since then.

Arduino: I began as an Arduino hobbyist. Now I can attack just about any Arduino task with a fairly high degree of confidence.

Interpersonal Communication: I have learned quite a bit about working with people since I began this project. Most of this has to do with presenting myself properly. I would say that the majority of this learning has been done on an emotional and sub-lingual level.

Notes >> Future Work:

Things I would like to see expanded / fixed in the future:

Camera For Line Following : while this may seem like overkill, a camera has the potential to be a bit more robust than many sensors. The power requirements are less. And it is an interesting challenge.

Smart Motors / Enhanced Motor Controller: I would like to see DC motor control that closely emulated servo control using encoders, PID controllers, and other hardware/software. Using an Arduino to read encoders pretty much puts it out of commission for any other work, it would be nice if some IC were used instead, for example. In fact, it would be great to not have to use an Arduino at all for motors.

Wireless programming using XBEE Shield – FIO's are nice, but they operate on 3.3V. It would be nice to work out the issue with wireless programming and get away from using wires for microcontroller<->PC except when needed.

Notes >> Future Work:

IMU and encoder localization: I would have loved to start the semester with the alternate robot and spent the whole time working out how accurate a fix I could get on position using the IMU and encoders.

PID controls: I wish I had more time to spend on tweaking the PID controllers I implemented. It seems that with PID controllers and encoder feedback alone it should be possible to get a reliable response from the motors in a low-slip or no-slip situation.

Data Fusion / Kalman filter: another interesting problem is that of fusing all the data from the IMU and encoders into a reliable model of the robot's movement. A semester spent on this would be rewarding and productive if approached properly.



Thanks for your time.

Questions?



Greg Drewry

Electrical Engineering
IEEE Robot Team Captain

2012 IEEE Hardware Team Competition

Team Formation:

Sept 2011

-What difficulties are there in forming a team?

How do we generate interest and get people involved?

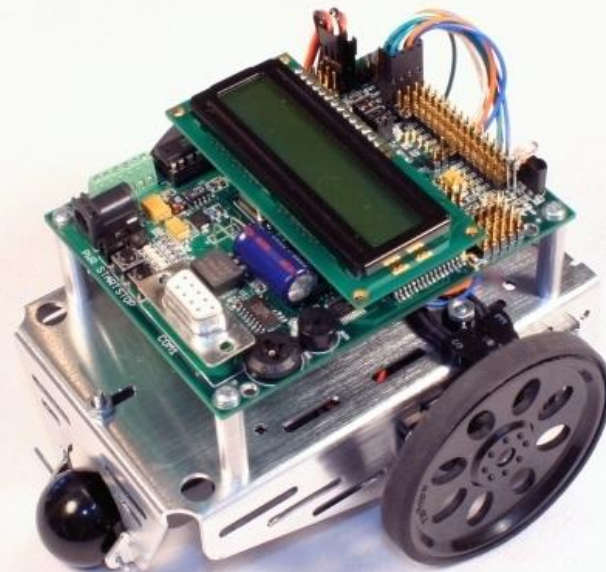
How do we keep people involved?

-Why did I join the team?

Interest in Robotics

Senior Design Credit

Opportunity to tackle real world problems



2012 IEEE Hardware Team Competition

Recruiting:

- How recruiting was handled:
 - * Personal emails and conversations
 - * Talks to classes
 - * Information sessions

Recruiting Results:

- Initial interest
 - * Eleven people initially interested in team
- Dedicated team members
 - * Nicole Pennington, Ammar Aalkhuwaiter, Ryan Young, Harrison (Jay) Cross, Chad Armstrong, Greg Drewry, Alan Hancock

Sept 2011



Key People involved in recruiting:

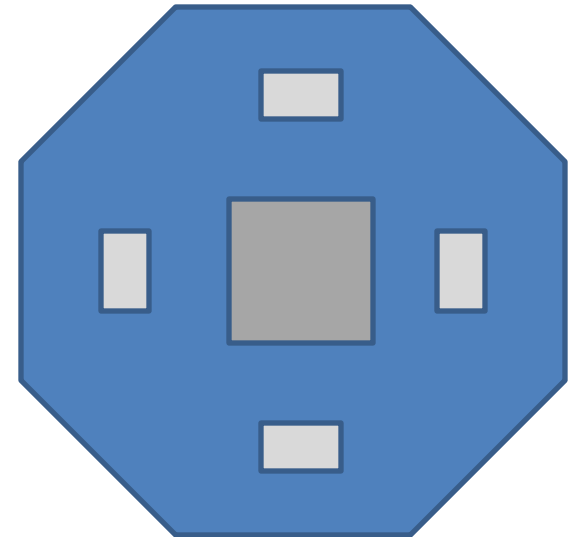
Will Martin
Rick Gonyea

2012 IEEE Hardware Team Competition

Design:

- Initial constraints of design
 - * Robot had to navigate an octahedral
 - * Robot had to test four stations to determine navigation for score
- Initial design ideas discussed
 - * Movement (tires, tracks, holonomic wheels)
 - * Sensor readings (remote or on robot)
 - * Navigation (wall or line following)
- Change of course layout to rectangular (Sept 30th)

Sept 2011



Decisions made:

- Team members that attended last meeting in September decided upon a team captain (Greg Drewry)
- Generalized ideas of robot design agreed upon (holonomic wheels and remote sensors)
- Agreed upon budget to propose to IEEE for funding

Final hardware competition rules can be found at:
<http://www.southeastcon2012.org/StudentProgram.html>

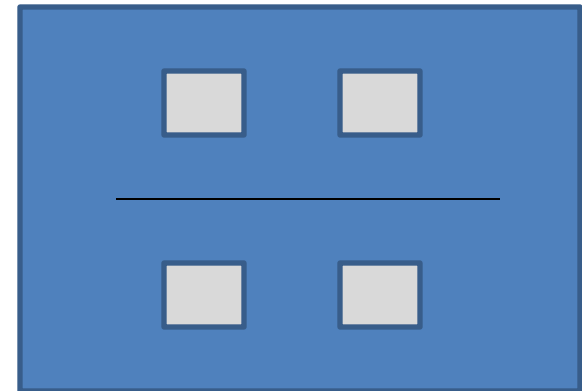


2012 IEEE Hardware Team Competition

Design:

Sept 2011

- Initial constraints of design
 - * Robot had to navigate an octahedral
 - * Robot had to test four stations to determine navigation for score
- Initial design ideas discussed
 - * Movement (tires, tracks, holonomic wheels)
 - * Sensor readings (remote or on robot)
 - * Navigation (wall or line following)
- Change of course layout to rectangular (Sept 30th)



Decisions made:

- Team members that attended last meeting in September decided upon a team captain (Greg Drewry)
- Generalized ideas of robot design agreed upon (holonomic wheels and remote sensors)
- Agreed upon budget to propose to IEEE for funding

Final hardware competition rules can be found at:
<http://www.southeastcon2012.org/StudentProgram.html>

2012 IEEE Hardware Team Competition

Budget:

Oct 2011

- Initial budget was agreed upon Oct 12th (for the amount of \$480)
- Initial budget proposal was finalized and presented to Shannon Eubank Oct 19th
- Potential additional funding sources discussed with Will Martin
- Account possibilities examined (University, private separate tax free, IEEE joint tax free)
This was particularly tedious due to outside forces preferring one way over another. A breakdown of deciding factors is below.

University Account

Pro

Donations are tax deductible (would go through UT accounting)
Purchase procedure makes fraudulent use of funds difficult

Con

Purchase procedure requires approval which can take time
Purchases may be difficult or impossible from some suppliers
Any money left in account at end of June is lost
Account would have to be created every year

Private Account

Pro

Freedom to spend money as we see fit
Account and balance would carry over each year

Con

No accountability for purchases made
Lengthy process to get tax exempt status
Donors could potentially question the legitimacy of the requests for funding

IEEE Joint Account

Pro

Simplicity of transfer of donated funds from IEEE student chapter
Accountability of having IEEE treasurer able to monitor transactions
Freedom to spend funds as we see fit
Account and balance would carry over each year
Donations would be made to IEEE student chapter (for Robot Team)

Con

Donation requests would require the cooperation of IEEE Treasurer

****Note:** The details of processes involved and regulations related to each account were investigated but are omitted here.

Budget Proposal that was presented to Shannon is available at
<http://www.imaging.utk.edu/classes/ece400/spring2012/gdrewry/reports.html>

Form

Storm

Norm

Perform

2012 IEEE Hardware Team Competition

Budget:

Oct 2011-March 2012

- Budget proposal was accepted October 20th
- Reminders sent every few days about setting up joint account and dispensation
- Joint account setup and deposit made November 8th (\$480)
- Discussed funding from outside sources with Shannon and Will November 9th

Will Drafted a form letter that required only the business name and amount of donation we were asking for from his requests sent the previous year. A list of businesses who've donated in the past and who we wanted to ask was compiled with amounts and handed to Shannon to send out. The list included any contact information we had from previous years and how the requests were sent in years prior.

- Inquired with Shannon about incoming funding donations January 11th

Shannon informed me she hadn't received a single response from anyone that she was asked to send donation requests to. She also stated that she had yet to send off the request to IEEE East Tennessee. She informed me on January 15th that she had just sent off the request to IEEE East TN.

- Received donation of \$1500 from IEEE East TN February 1st
- Shannon deposits \$500 from IEEE East TN donation February 3rd
- Shannon deposits remaining \$1000 from IEEE East TN Donation February 21st
- Transportation and lodging budget proposal approved by Dr Tomsovic February 21st

Dr Tomsovic approved a joint budget for the hardware and software teams March 5th

Form

Storm

Norm

Perform

2012 IEEE Hardware Team Competition

Budget Numbers:

4/18/2012

- Cash donations received totaled \$1980
- Nonfinancial donations from the IRIS lab covered our cost of transportation
- The department reimbursed the team for lodging and registration fees (~\$2000)

Budget Management:

- Budget constraints limited the design options we could entertain and committed us financially to following through without much room for deviation from original plan
- Limited to two income sources by a bottleneck in supply process out of our control
- Once registration fees are reimbursed by the department and final accounting is done there will be a surplus of approximately \$150 in the account

Budgeting Ideas for the future:

- Involve a couple of Business/Marketing majors to assist in fundraising
- Develop and maintain an updated contact list for businesses that are annual donors
- Try to establish a savings designed to cover miscellaneous costs such as cookouts and newspaper advertisements thanking sponsors

Form

Storm

Norm

Perform



2012 IEEE Hardware Team Competition

Roles Defined:

Oct 2011

- The idea of modular design allowed for independent work on roles
- Preliminary roles were identified as chassis, sensors, communication, navigation, motors, power and racer

Problems encountered with defined roles:

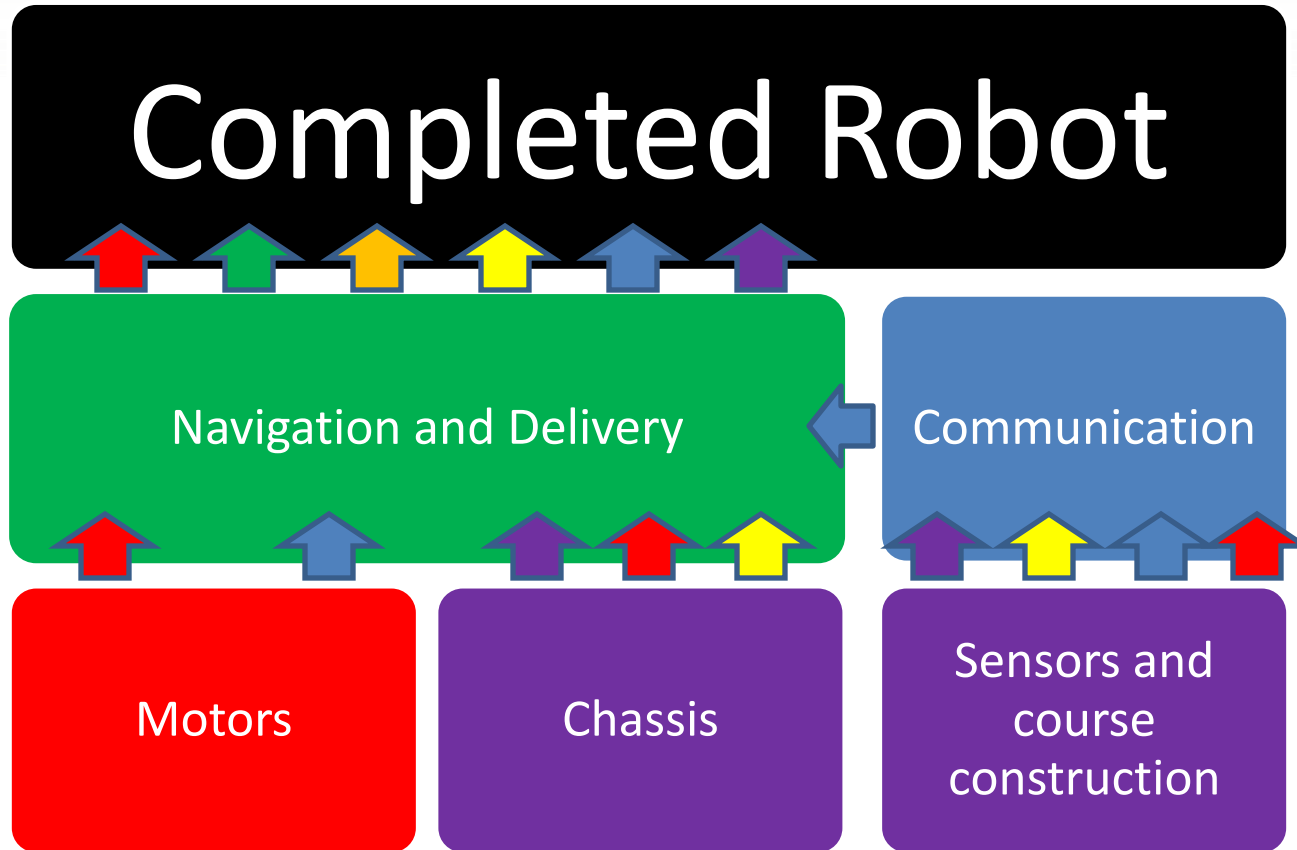
- Lack of funding resulted in a lot of talking, but little work getting done initially
- Loss of interest caused some preliminary roles to be abandoned by temporary team members (possibly partially caused by lack of any materials to work with)
- Some roles were reliant upon others being completed prior

How problems were overcome:

- Roles left open from abandonment were absorbed by remaining team members
- Long hours became common once funding and material issues were resolved

2012 IEEE Hardware Team Competition

Completed Robot



-  Greg
-  Ammar
-  Jay
-  Nicole
-  Ryan
-  Chad

2012 IEEE Hardware Team Competition

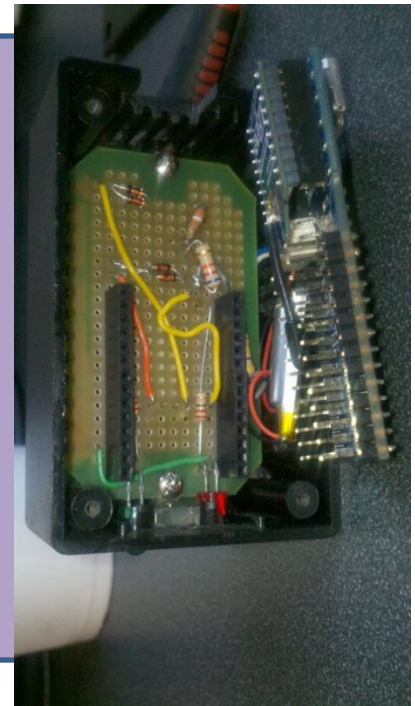
Sensor Design:

Jan 2012

- Design, test and realize a waveform recognition sensor (5V rms Square vs Saw tooth)
 - Sampling rate was limited below the Nyquist rate
 - Wanted to keep design as simple as possible
 - Design had to be able to be implemented on Arduino Fio

-Solution

Use the differences in the properties of the waveforms to distinguish the peak value of a 5V rms square wave is $\sim 5V$, whereas the peak of a saw tooth wave is $\sim 8.66V$. Regardless of sampling rate, if sample set is large enough probability is there will be at least 1 sample above 5V. By sampling and recording the maximum value seen when sampling a square wave a threshold was able to be determined. A buffer factor was added to the threshold to insure any samples above the threshold were not due to random noise. If over any sample period at least one sample was above the threshold, a flag was set indicating the waveform was a saw tooth. The flag would be updated at the end of each sample set.



2012 IEEE Hardware Team Competition

Sensor Design:

Jan 2012

-Design, test and realize temperature sensor

Temperature sensor needed to swing quickly over a range of at least 40° F

Design had to be able to be implemented on Arduino Fio

Design had to form good connection with temperature plate

-Solution

Tested a digital sensor to find side most sensitive to temperature changes. Mounted temperature sensor so that the stiffness of the leads leaving the sensor acted as a spring to keep the sensor pressed against the plate once the remote sensor box was delivered properly. Thermal grease was applied to the contact side of the sensor to further increase thermal transfer.



2012 IEEE Hardware Team Competition

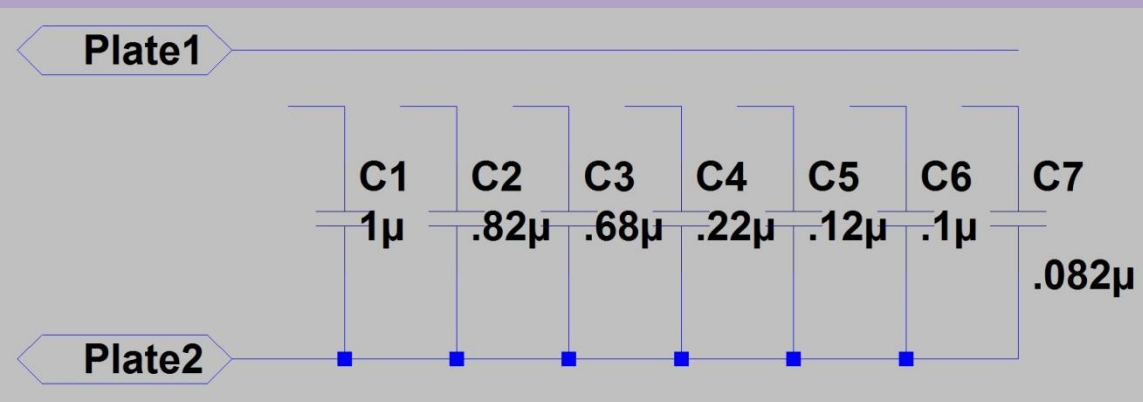
Sensor Stations:

Jan 2012

-Design, construct and verify sensor stations for capacitance, temperature and waveform tests. **Ryan had previously done such for the voltage station

-Capacitance station

Station was realized by wiring switched capacitors in parallel so you could achieve both the $<100\text{nF}$ and $>1\mu\text{F}$ thresholds as defined by the rules. Capacitance values measured a less than 10% below marked in all case combinations in testing.



Form

Storm

Norm

Perform

2012 IEEE Hardware Team Competition

Sensor Stations:

Jan 2012

-Temperature station

Station was realized by using a Peltier TEC. The station needed to swing $\pm 10^{\circ}$ F from ambient temperature. The datasheet for showed the TEC could handle up to 7A of current, but no more than 15V. With 2 ohms internal resistance, I used a 10 watt 10 ohm resistor to limit the current from a 12V power supply to achieve temperature swing times of a couple of seconds. A heat sink and thermal grease was added to one side to help prevent heat buildup causing degraded or failure of performance.



2012 IEEE Hardware Team Competition

Sensor Stations:

Jan 2012

-Waveform Station

The station was constructed by running two wires from a waveform generator under the course into the station. The generator was set for a 5V rms High Z impedance output. The switching of waveforms was accomplished by pushing the proper waveform selection on the generator. The wires were there connected to the plates located on the front of the boxes from the hardware securing the plates. The plates, identical on all stations except the temperature station, were secured by bolts through the plates extending into the inside of the box. The wire terminals could then be attached underneath the nuts inside.



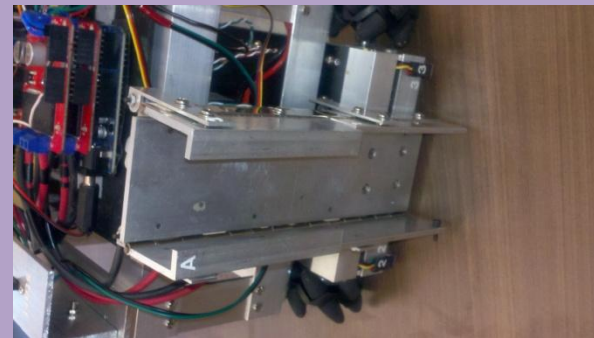
2012 IEEE Hardware Team Competition

Sensor Delivery System:

Jan 2012

-The sensor delivery system had to be able to transport and deliver reliably the remote sensors. This was to be a modular unit.

Design was constrained by limited space available for delivery, delivery time and reliability. There were a number of designs considered including arms, pincers and clamps. The design that best made use of space available took advantage of the fact that there wasn't any space issues extending above the robot. This was done by having one sensor loaded into the delivery unit while a second sensor sat directly above the first in a storage unit. Once the first was delivered, the second sensor would travel until ready for deployment. It would then be reliably dropped into the clutches of the delivery unit for deployment. The idea of only stacking one sensor was to eliminate the need to consider how the weight of additional sensors would change the reload process for the delivery mechanism. This way the process was identical for both units. Ryan assisted by constructing one of the delivery units. The delivery unit is built from aluminum angle and plate stock.



2012 IEEE Hardware Team Competition

Chassis:

Jan 2012

-The chassis had to be stiff enough to remain square under the weight of the robot. The chassis also would act as the mount for the motors. Because of this it also had to be sturdy enough to resist the twisting due to the torque of the motors.

The first two versions of the chassis were built in haste out of plastic wood without worries of getting things square or stiff just to get something that Ammar could work with. There was a set of motor mounts purchased between chassis versions 2 and 3, but they weren't stiff enough that anyone involved in the movement of the robot felt comfortable using them. Chassis version 3 is constructed out entirely out of aluminum angle and plate stock. Version 3.5 saw the addition of a basic enclosure designed to house a smaller racer robot inside. The motor controllers, power supply and battery were to be moved to a shelf above the enclosure. **Note: The racer never came to fruition, but the enclosure was completed aside from mounting a fold down launch ramp.



2012 IEEE Hardware Team Competition

Power Supply:

Jan 2012

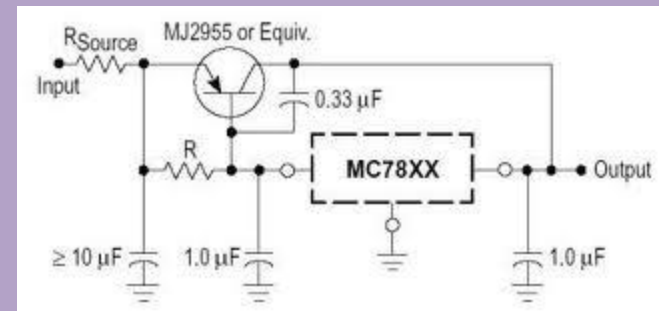
- Needed to regulate a battery from 14.7V down to a 9V rail and 5V rail
- The 9V rail was to supply power to the onboard Arduino microcontrollers
- The 5V rail would power the custom analog sensor array

Demands:

The 9V rail had very limited current demands, but needed to be stable and spike free. The 5V rail needed to be able to meet the current demands of the sensor array it supported. The analog sensor array was originally supported using a LM7805, but the current demand of nearly 700mA was causing the regulator to go into thermal shutdown.

Solution:

I continued using the LM317 tuned to 9V to support the 9V rail. To solve the current demands of the sensor array I built a switching power supply in the configuration of the diagram. I used a 2N5684 PNP power transistor to supply the current for the rail. My components varied a little as I used a LM7805 to regulate the CE junction and voltage on the rail. Capacitance and resistor values were chosen to make it work, but not optimized as the power supply was built on short notice to get everything working again so work on the navigation code could continue.



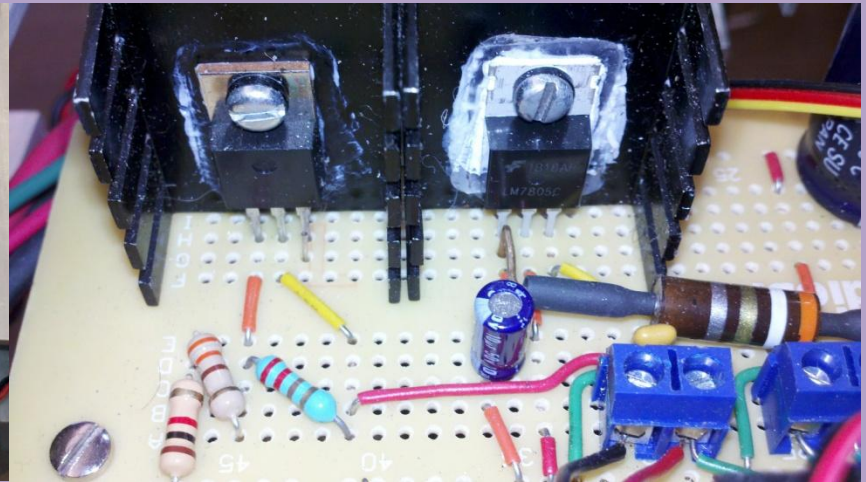
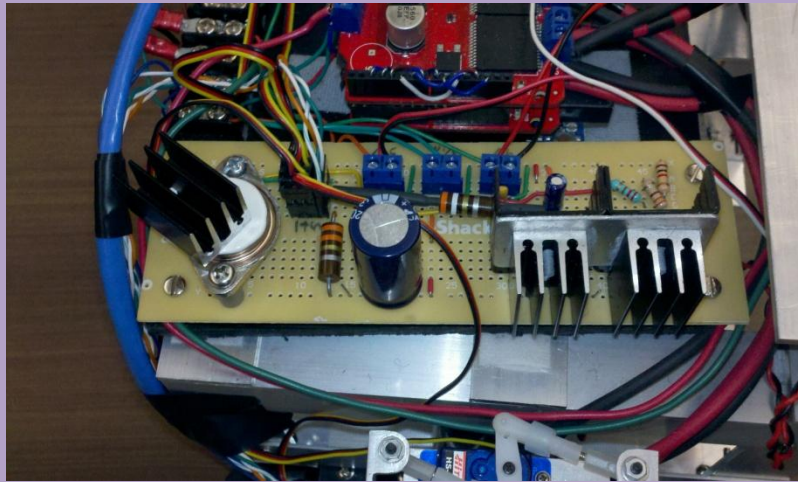
2012 IEEE Hardware Team Competition

Power Supply:

Jan 2012

Additional power supply information:

Notice the all the regulators and transistor have a heat sink. This was done in an effort to minimize risk of component damage and prevent thermal shutdown from choking output currents. The heat sink on the PNP transistor was initially installed just using thermal grease as an adhesive. It kept getting bumped off, so the heat sink was later secured in place with a thermal epoxy. During the curing phase someone bumped it and thus it came to permanently rest skewed off to one side of the transistor. The rebuilt 5V rail has supported a load around 1A without any voltage drop on the rail. With proper cooling it should be able to support at least 1.5A without voltage roll off. Ammar had previously biased the LM317 before I redesigned and rebuilt the built the power supply. Those biasing resistors were retained in the rebuild process.



Form

Storm

Norm

Perform

2012 IEEE Hardware Team Competition

Team Management:

Spring 2012

-Tee Shirt design

Secured the assistance of Project Phantom in making the brainstormed idea of RoboSmokey a visual creation. The guys are incredible artists and former UT graduates. They provided a very good design at a discounted cost when compared with the other commercial places around town.

Securing printing wasn't as easy as I had to be in regular contact with six different local and non-local businesses who could do the screen printing for us. I recorded the quoted costs and how the costs were setup. I also developed a contact (Mike Keener) at the collegiate licensing committee (CLC) here at UT. That was also recorded so that any future designs to be screen printed can be preapproved before they ever go to the printer.



2012 IEEE Hardware Team Competition

Team Management:

Jan 2012

It wasn't until the semester got started that we really started working as a team. We started the semester still feeling out our place in the storming phase. Then we set some clear short term goals and the team quickly fell into their roles and began performing. There were key moments when the team would revert back to the storming phase, but each time it normalized and began to perform again more quickly than the last.

Mistakes in management I made:

I should have been more outspoken and decisive. On a couple of occasions we sat stagnant for a period of time because no one knew where or how to act next. Any action, right or wrong, would have been better than sitting stagnant. I also could have been more open to alternative solutions to the problem instead of figuring out how to make what we have work.

Things I did well:

Budget the limited funds. Coming in under budget should give next years team a little head start.
Setting up the account so funding and fund transfers will be simpler in the future.
Creating contacts within the department and with outside businesses. The search no longer starts at zero.
Being available. I tried to be available anytime to help whether it was an order or building access.
(marginal: diffusing tense situations between team members)

Form

Storm

Norm

Perform

Additional Work

-In process

Promoting/recruiting for next years team

- Information session 4/18 at 4PM in Min Kao 337

- Information session put on screens in building, posted on EF website, asked Julia Elkins to send departmental email.

Interest in potentially competing in additional competitions next year

- Automated lawnmower competition

- Currently researching if it's possible to get the university to insure a robot for competition

- Plan on talking to Dr. Arel and Dr. Parker about being faculty advisors for other competitions

Working to setup a UT Robotics Club (student organization)

- Have current constitution and bylaws of IEEE student chapter. Plan to use it as a reference in conjunction with other student organization constitutions to write the constitution for a robotics club. Website will follow once organization becomes officially recognized.

Working with Mark Collins to diversify the team composition

- Mark Collins is the faculty advisor for the American Marketing Association.

- Want to work this summer to attempt to develop an updated contact list for fundraising and donation requests. Hope to begin to secure funding for next years team this summer.

Questions?

Ammar Alkhuwaiter

Computer Engineering

IEEE Robot Team

Final Presentation



Team Objectives:

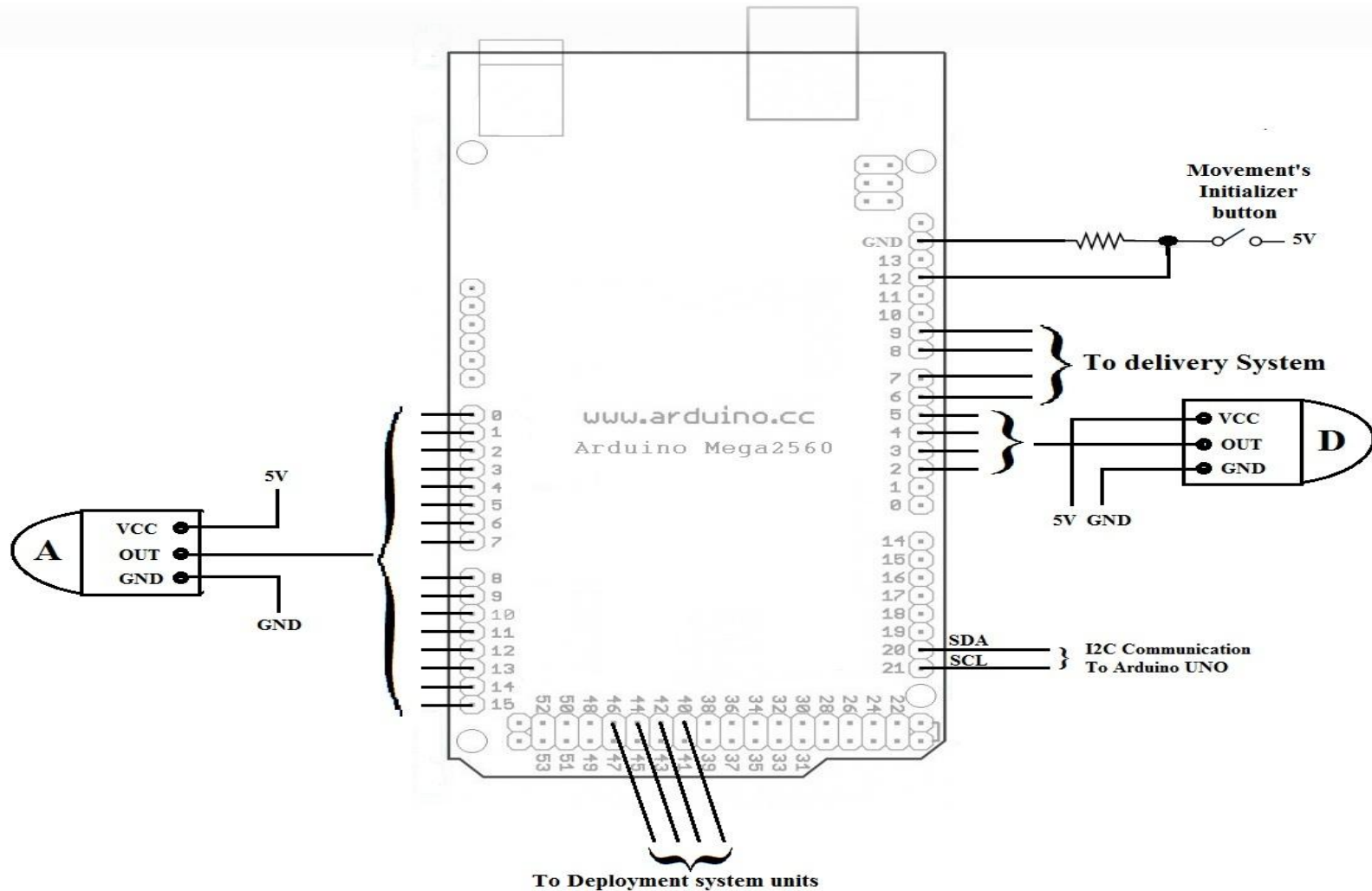
It was required to design and implement a Robot that can compete in the SouthEastCon2012 Hardware competition. The Robot was tasked with traversing a maze-like course, analyzing electrical signals and other Information along the way to point it in the right direction. Points were assessed for correct readings and interpreting the information consistently as the robot circulating the track.



My Role:

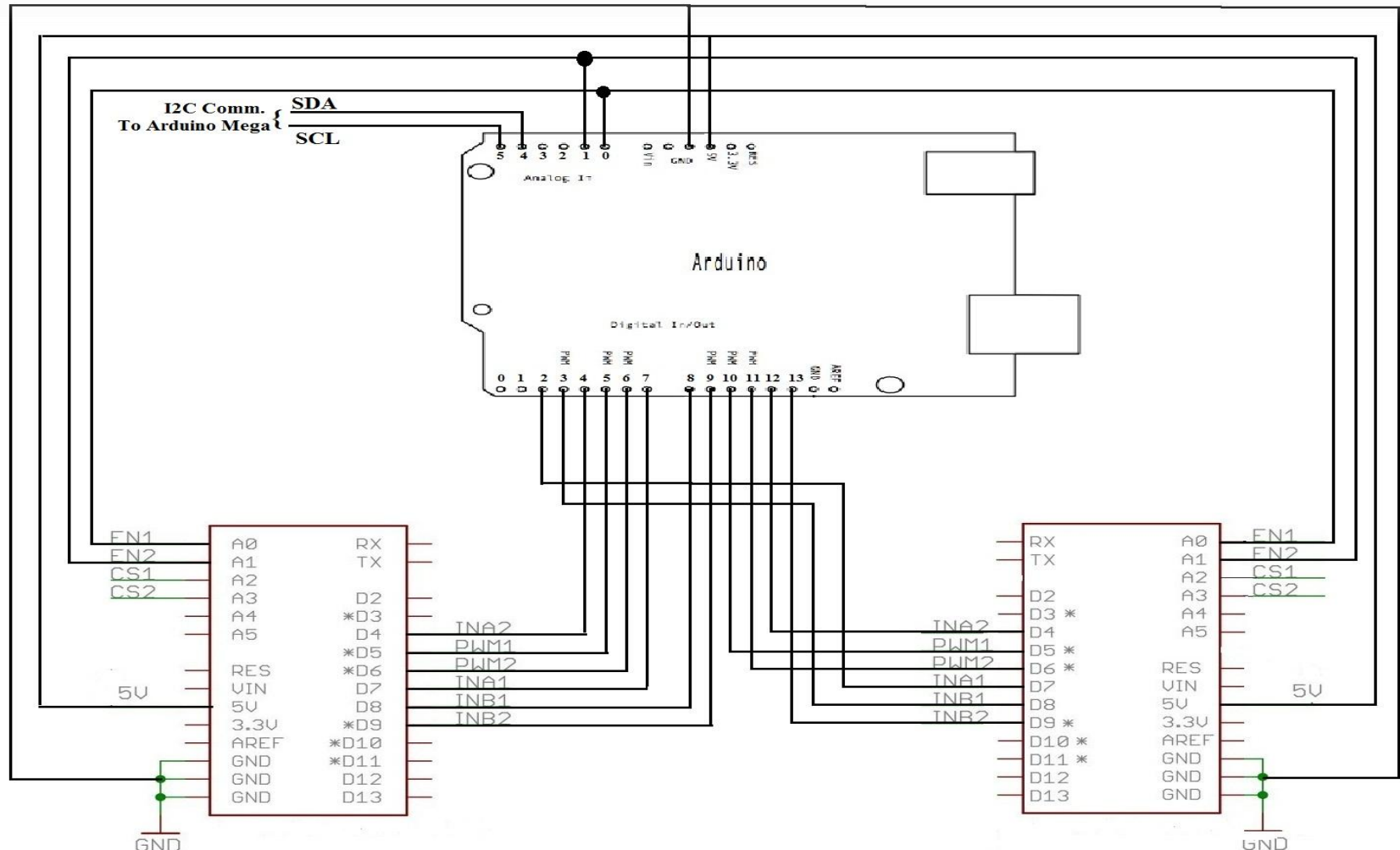
- 1) Robot's Movement Control
 - using a 4 wheels drive using mecanum wheel.
 - controlled by Arduino board.
 - Using Pulse Width Modulation to control speed.
- 2) satisfying the power requirements for the robot.
- 3) Miscellaneous tasks

Main brain Wiring diagram



Robot's movement Control:

Movement unit wiring schematic diagram





Monster Moto shield motor controller

- Dual of VNH2SP30 full H-bridge motor drivers.
- Maximum/continuous current rating: 30 A/14A.
- Voltage max: 16V.
- Circuit protection including:
 - Thermal Shutdown.
 - Undervoltage and Overvoltage shutdown.
- Easier to connect to Arduino.
- higher efficiency in delivering current.



Monster Moto shield motor controller

- EN: to “enable” the movement for that motor
 - CS: “current sense” to measure the current going to that motor.
 - INA: give order to move in clockwise direction
 - INB: give order to move in counter clockwise direction
 - PWM: to control the movement speed.
PWM value range 0-255 with 255 being the fastest.
 - 5V: The shield’s power supply directly from Arduino.
 - GND: ground
- Numbers indicate which motor to be controlled.

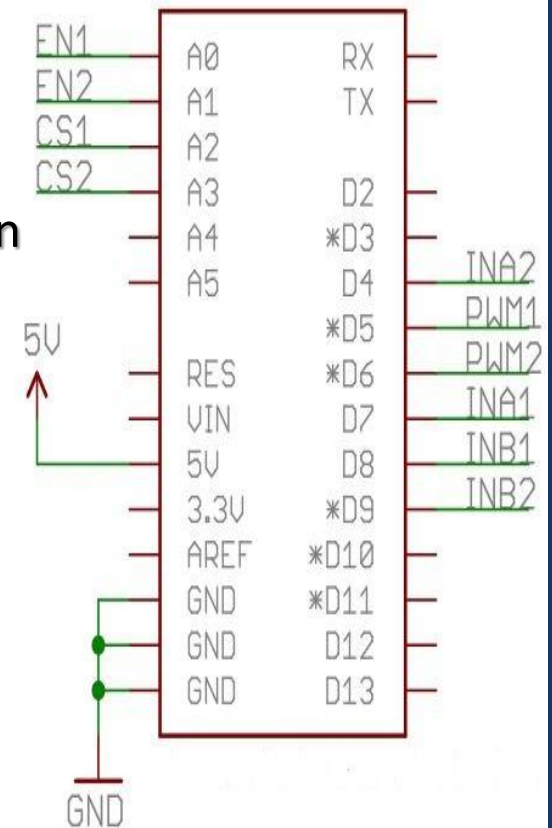




Table 1
Monster Moto Shield protection

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit
V_{USD}	Undervoltage shut-down				5.5	V
	Undervoltage reset			4.7		
V_{OV}	Overvoltage shut-down		16	19	22	
I_{LIM}	High side current limitation		30	50	70	A
V_{CLP}	Total clamp voltage (V_{CC} to GND)	$I_{OUT} = 15A$	43	48	54	V
T_{TSD}	Thermal shut-down temperature	$V_{IN} = 3.25V$	150	175	200	°C
T_{TR}	Thermal reset temperature		135			
T_{HYST}	Thermal hysteresis		7	15		

Table 2

Truth table in normal operating conditions

IN_A	IN_B	$DIAG_A/EN_A$	$DIAG_B/EN_B$	OUT_A	OUT_B	CS	Operating mode
1	1	1	1	H	H	High Imp.	Brake to V_{CC}
	0				L	$I_{SENSE} = I_{OUT}/K$	Clockwise (CW)
0	1			L	H		Counterclockwise (CCW)
	0				L	High Imp.	Brake to GND



Table 3
Truth table in fault conditions

IN _A	IN _B	DIAG _A /EN _A	DIAG _B /EN _B	OUT _A	OUT _B
1	1	0	1	OPEN	H
	0				L
0	1				H
	0				L
X	X		0		OPEN
	1		1		H
	0				L

Table 4
Switching and delay time

Symbol	Parameter	Max	Unit
f	PWM frequency	20	kHz
$t_{d(on)}$	Turn-on delay time	250	μs
$t_{d(off)}$	Turn-off delay time	250	μs
t_r	Rise time	1.6	μs
t_f	Fall time	2.4	μs
t_{DEL}	Delay time during change of operation mode	1800	μs
$t_{off(min)}$	PWM minimum off time	6	μs

DC Motors:

Part #: HN-GH12-2217Y

- Rated Voltage: **12vdc**
- Voltage Operating Range: **6-12vdc**
- Rated Load at 12vdc: **1.14Kg-cm**
- No Load Speed at 12vdc: **120 RPM**
- Speed at Rated : **95 RPM+/- 10%**
- No Load Current at 12vdc: **< 108mA**



Movement Code's Functions

- move_forward/reverse(speed)*
- move_stop
- turn_right(speed)
- turn_left(speed)
- forward_right (harshness)**
- forward_left (harshness)
- back_right (harshness)
- back_left (harshness)
- pivot_front_center
- pivot_front_right
- pivot_front_left
- pivot_back_center
- pivot_back_right
- pivot_back_left

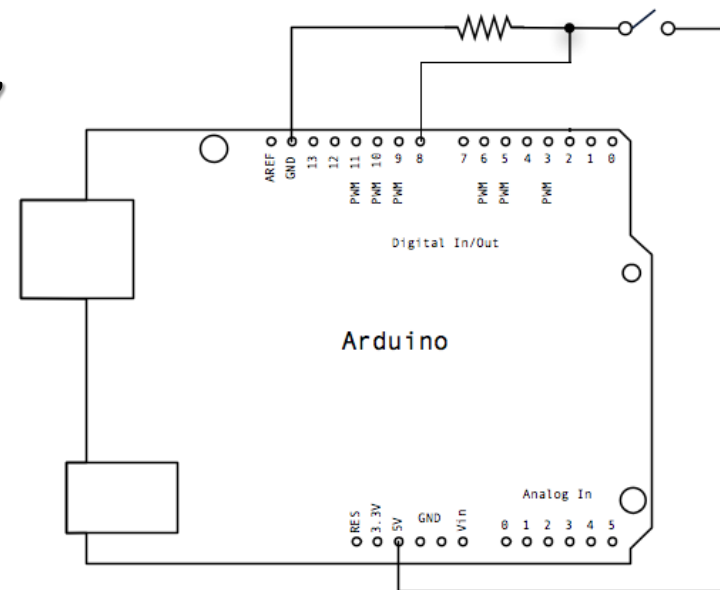
18 special Pivot turn functions for each specific turn in the course for perfect movement

*Speed: Integer variable from 1 – 4 with 4 being the fastest

** Harshness : Integer variable from 1 – 4 with 4 being the harshest

Movement Initializer Button

- Connected to any **DIGITAL** pin on the arduino.
*we connected to D12.
- using a 10k Ω pull down resistor to the ground.
- When switch is “ON”, the digital pin set “HIGH”
- When switch is “OFF”, the digital pin set “LOW”
- DigitalRead the pin and initialize the movement when D12 is “HIGH”

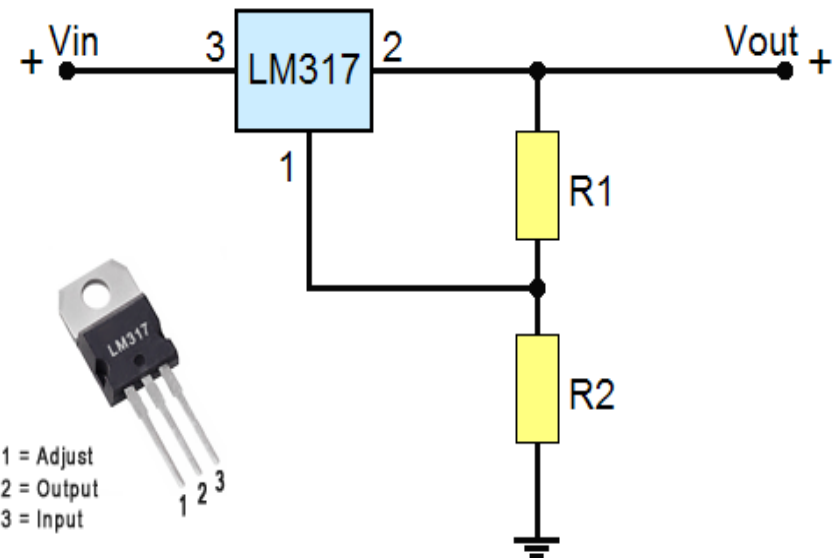


Voltage regulator

Using LM317:

- Adjustable and simple circuit
- Output voltage range 1.2 – 37 Volts
- Current limitation 1.5A.
- $V_{OUT} = 1.25 * (1 + R_2/R_1)$
- R_1 should be in the range of 100-1000 Ohms (typically a 240Ω)
- With the 9V regulator:

$$R_1 = 240\Omega \quad R_2 = 1.5k\Omega$$



Power Consumption

Unit	Number of units	Current (Amp)	Voltage (Volt)	Power (Watt)
Analog sensor	16 sensors	0.05	5	4
Digital sensor	4 Sensors	0.05	5	1
DC Motor	4 Motors	~ 1.5	~ 12	72
Deployment units	4 Units	0.25	5	5
Arduino Boards	2 Boards	0.05	9	0.9
Arduino mega Pins	31 Pins	0.04	5	6.2
Arduino Uno pins	16 Pins	0.04	5	3.2
Total Power Consumption				92.3 Watts

The battery that we used is 14 cells **NiMH** battery

Total Battery power = **14.4Volts * 4000mAH = 57.6 Watts.hours**

Approximate Battery life in the robot = $57.6/92.3 = \underline{\underline{38 \text{ minutes}}}$



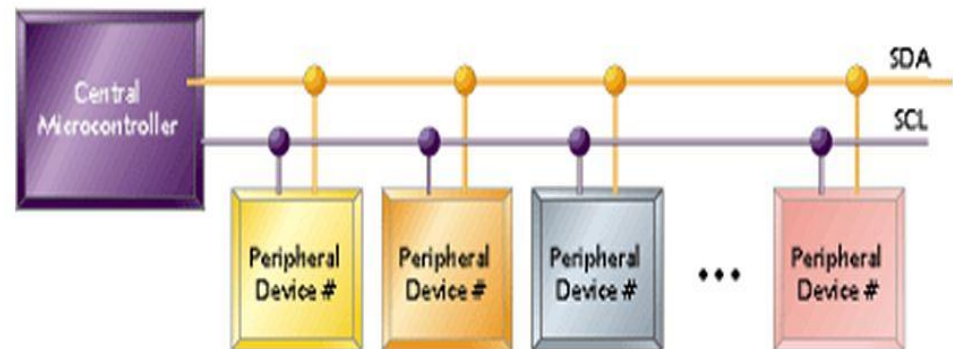
Miscellaneous Tasks:

I also Helped in different tasks in the robot's design and implement, include:

- Search and implement the I2C communication into the robot.
- Search and implement the Movement Intializer button.
- Search and code the Temperature sensor reading through Arduino board.
- Fine tune movement related tasks in the navigation.
- Help in building the sensors' array board and maintain the sensors' functionality.

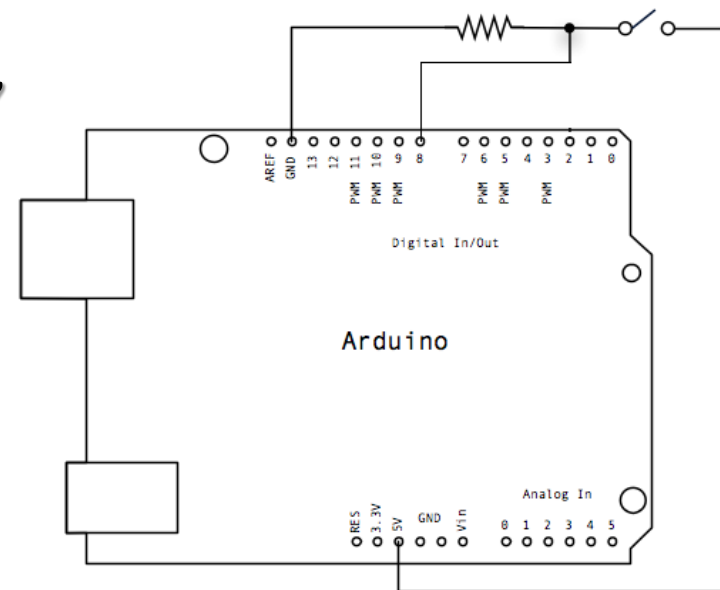
I2C Communication:

- Assign one Arduino as a Master, every other Arduino will be assigned as a slave with its own unique address
- **SCL** a serial clock pin that the Arduino pulses at a regular interval.
A5 pin for Uno, D21 pin for Mega
- **SDA** a serial data pin over which data is sent between the two devices
A4 pin for Uno, D20 pin for Mega
- As the clock pulse rises and falls, a bit of information is transferred from the master Arduino over the SDA line.
- A **common ground** should be used for all the connected Arduinos.



Movement Initializer Button

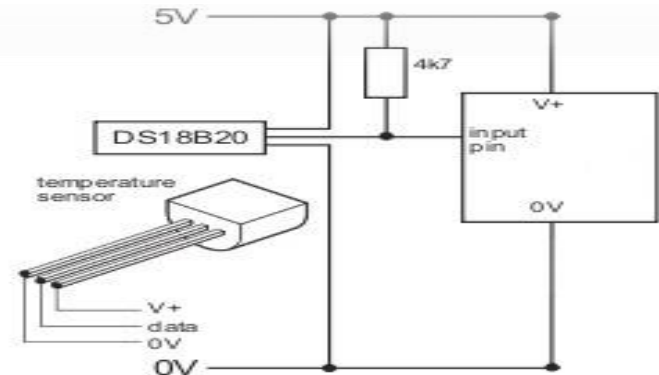
- Connected to any **DIGITAL** pin on the arduino.
*we connected to D12.
- using a 10kΩ pull down resistor to the ground.
- When switch is “ON”, the digital pin set “HIGH”
- When switch is “OFF”, the digital pin set “LOW”
- DigitalRead the pin and initialize the movement when D12 is “HIGH”





Temperature sensor reading

- Uses **DS18B20** Temperature sensor.
- Easy to connect and work efficiently.
- Unique 1-wire interface requires only one port pin for communication
- Reads in Celsius scale.
- operating temperature range of -55°C to $+125^{\circ}\text{C}$
- Accurate to $\pm 0.5^{\circ}\text{C}$ over the range of -10°C to $+85^{\circ}\text{C}$



Questions?



Or

Any comments!!



Nicole Pennington

Computer Engineering
IEEE Robot Team



Final Presentation

Objective

Successful Course Navigation

Design Decisions

Navigation Method

Line Sensors

Sensor Orientation

Finite-state Machine Navigation

Line Following

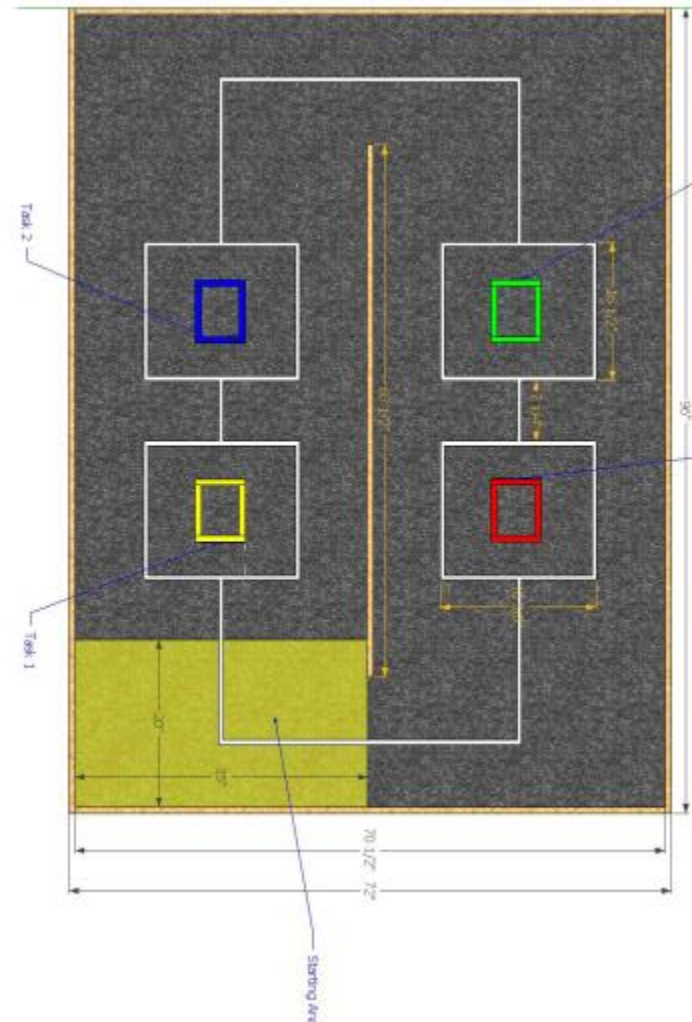
Turning System

Accomplishments

Future Work / Questions

Objective: Course Navigation

- Continuous navigation for 4 minutes, moving as quickly as possible
 - State machine approach
 - Line following
 - Integration of motor controls and deployment system



Design Decisions: Navigation Method

- **Image processing**
 - Complex, expensive
- **Magnetometer and encoders**
 - Navigation based on angle and distance from starting location
- **Line following sensors**
 - Readily available examples from previous years
 - Economical, reliable



Design Decisions: Hardware

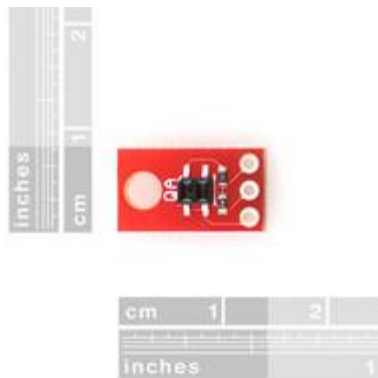
Line Sensors

Analog Sensor

- Voltage output varies the sensor output
- Output varies from 0 to 1023
- Code is simply an analog read call

Digital Sensor

- Capacitor discharge time varies the sensor output
- Requires more code to determine the reading



QRE1113 Line Sensor

- Dimensions: 0.3" x 0.5" x 0.1" (without header pins installed)
- Operating voltage: 5.0 V
- Supply current: 25 mA
- Optimal sensing distance: 0.125" (3 mm)
- Maximum recommended sensing distance: 0.375" (9.5 mm)

Design Decisions: Hardware Microcontroller

Arduino Mega 2560

- programmed using a Wiring-based language (syntax and libraries), similar to C++
- 16 analog sensor pins

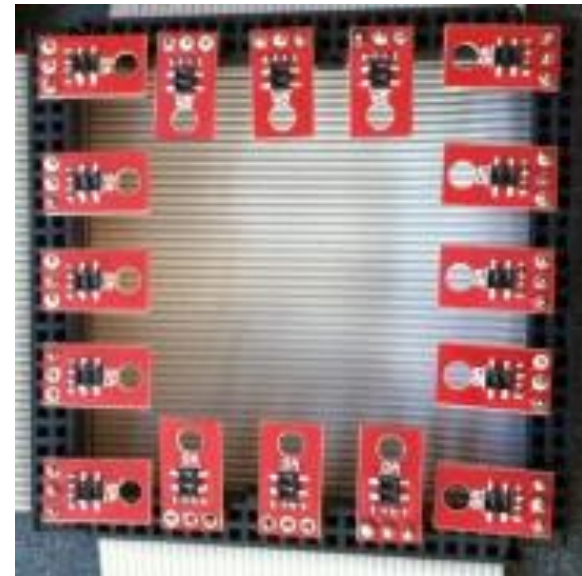
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz



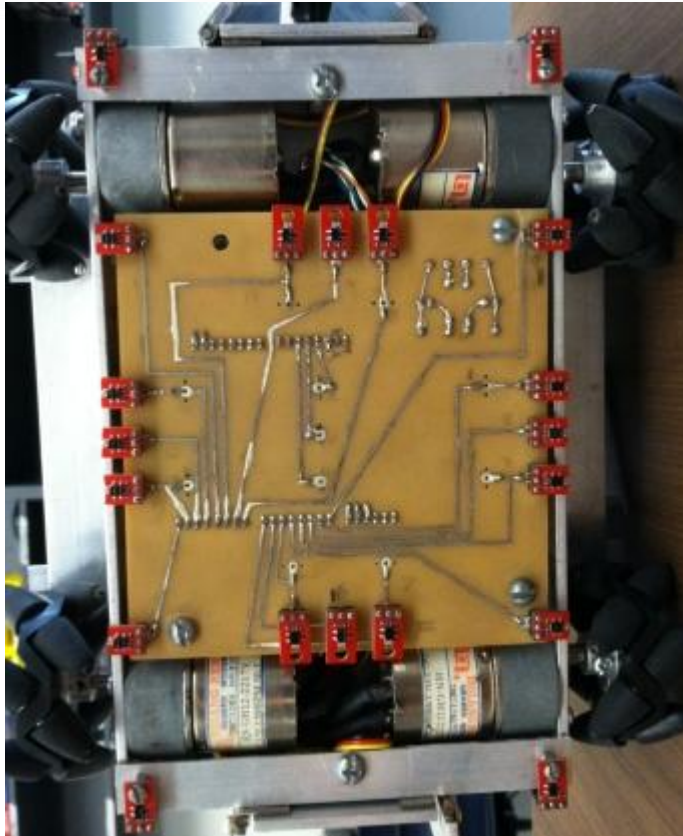
Design Decisions: Sensor Orientation

Original Design

- 16 analog sensor square
- Symmetric readings for forward/backward and lateral movements
- Designed with $\frac{1}{2}$ inch spacing to allow 2 sensors to read line ($\frac{3}{4}$ inch) at once to aid in line following adjustments
- Saw this sensor orientation on other robots at competition



Design Decisions: Sensor Orientation



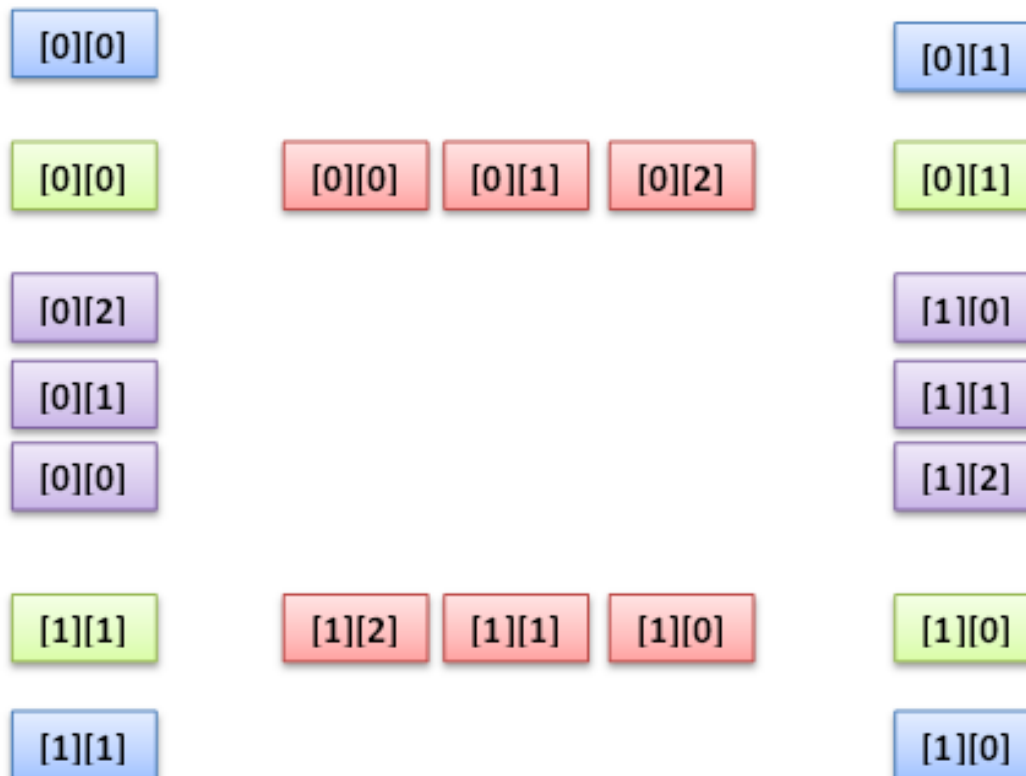
Final Orientation

- Extended original square without adding more analog sensors (16 port limit)
- Added 4 digital sensors to robot corners to reduce overshoot due to stop action delay
- Custom etched circuit board created by Ryan Young

Design Decisions: Sensor Orientation

Symmetry advantage in code: 4 sensor arrays

- Farcorners (blue), Corners (green), Sidesensors (purple), and Sensors (red)
- Front and left sensors are dimension [0], and back and right sensors are [1]

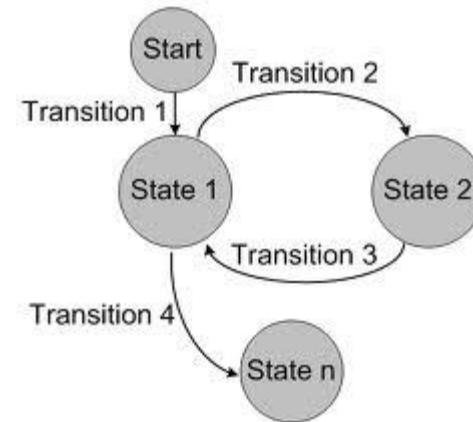


Design Decisions: FSM Navigation

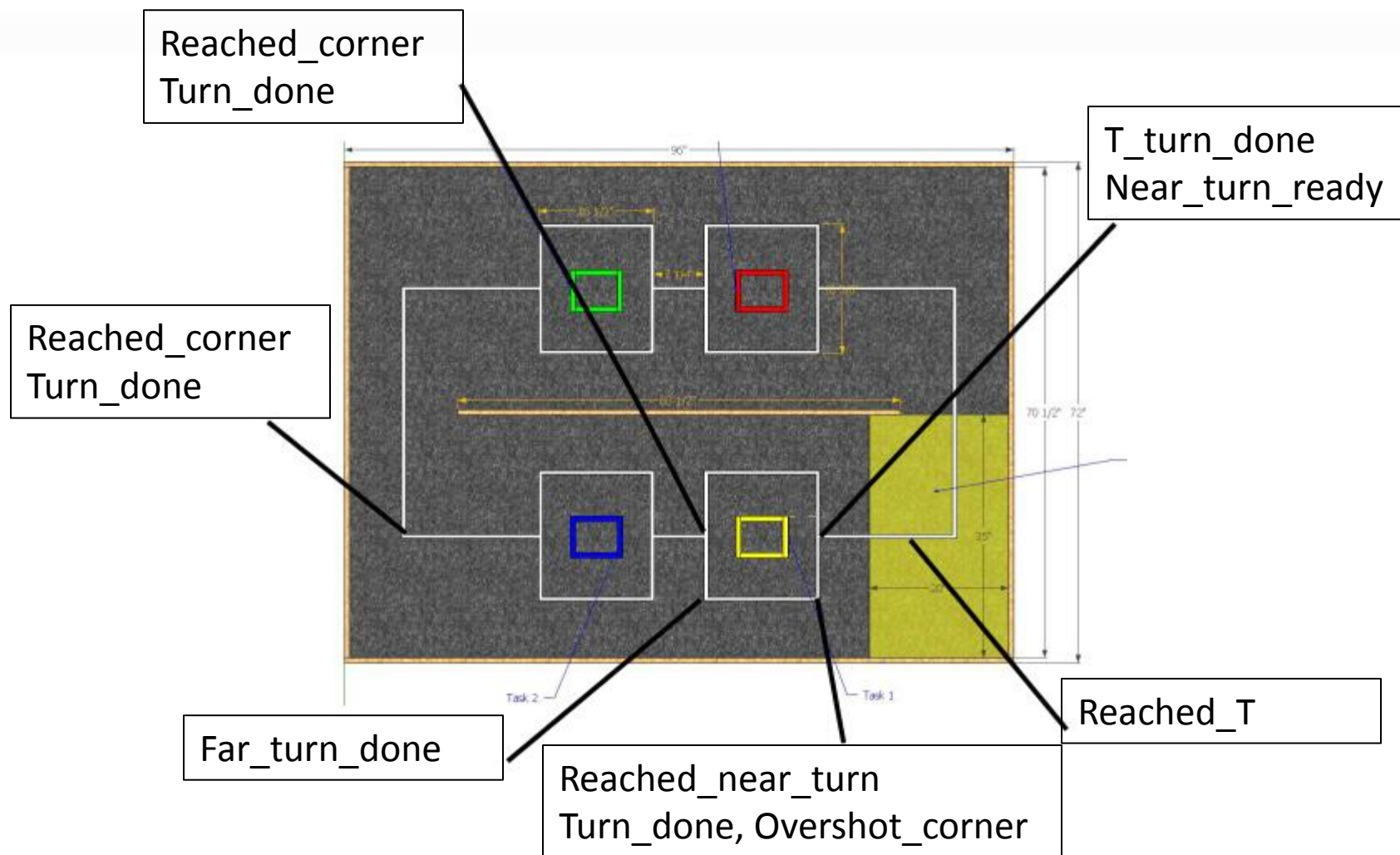
Note: Programming decisions were a joint effort with Chad Armstrong

Finite-state Machine (FSM)

- Navigation is based on trigger events which cause state transitions
 - Trigger events: sensor readings
- State-based navigation takes advantage of the symmetry of the course
- Line following and deployment were built into the appropriate states



Design Decisions: FSM Navigation



Design Decisions: Line Following

Pseudocode

front = total of 3 binary-valued front center sensors

back = total of 3 binary-valued back center sensors

if (front)

total += (left front)/front - (right front)/front

if (back)

total += [(back right)/ back - (back left)/back] / 2

if (total > 0)

total = (total >= 1) + (total >= .75) + (total >= .5) + (total >= .25);

veer_left[front](total);

else if (total < 0)

total = (total <= -1) + (total <= -.75) + (total <= -.5) + (total <= -.25);

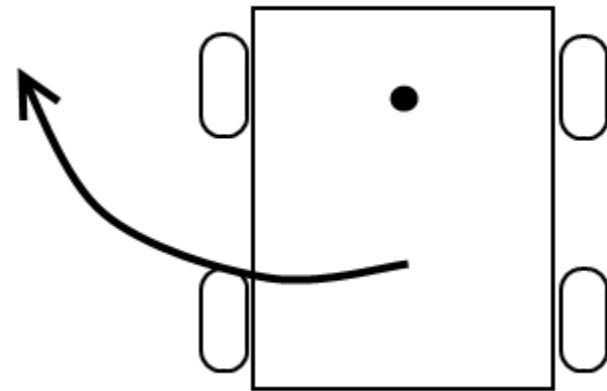
veer_right[front](total);

else

move[front](Speed);

Design Decisions: Turning System

- Original design hoped to use lateral shifting only – no turns
- With Chad and Ammar, worked on turning system that would still take advantage of wheels
- Turns were executed more precisely by pivoting
 - Pivot point was adjusted for each turn/direction experimentally



Accomplishments

- Wrote test code for sensors which displays output via serial port
- Assisted in API design
- Experimentally determined:
 - Optimal sensor threshold based on distance from floor (5mm)
 - Motor controller calibration based on sensor and motor response times (with Ammar and Chad)
- Delivery system integration (API provided by Ryan Young)

Future Work

Ran out of time for..

- Sensor calibration in code or data-driven sensor threshold detection
 - Some competitors used sensor lighting
- Navigation shortcuts – leaving the line intentionally to cut corners
 - Many competitors used this technique



Thank You