Voice Controlled Robot

Presented By: Brandon Galletta, Kory McTague, and Stephanie Valley

Outline

- Importance of Project
- Block Diagram
- Speech Recognition
  - Hardware and software
  - How it works
- Robot
  - Hardware and software
  - How it works
- Wireless Communication
  - Hardware and software
  - How it works
- Deliverables
- Future Work and Improvement

Importance of Project

- Speech recognition is the next step in the future of devices
    - Alexa, Google, Siri
    - Home Automation
    - Smart Cars
- Huge leaps in robotics are starting to be seen
    - Boston Dynamics
    - Sophia
    - iRobot Vacuum (Roomba)
- Helps visually, hearing, and physically impaired individuals

Overall Block Diagram

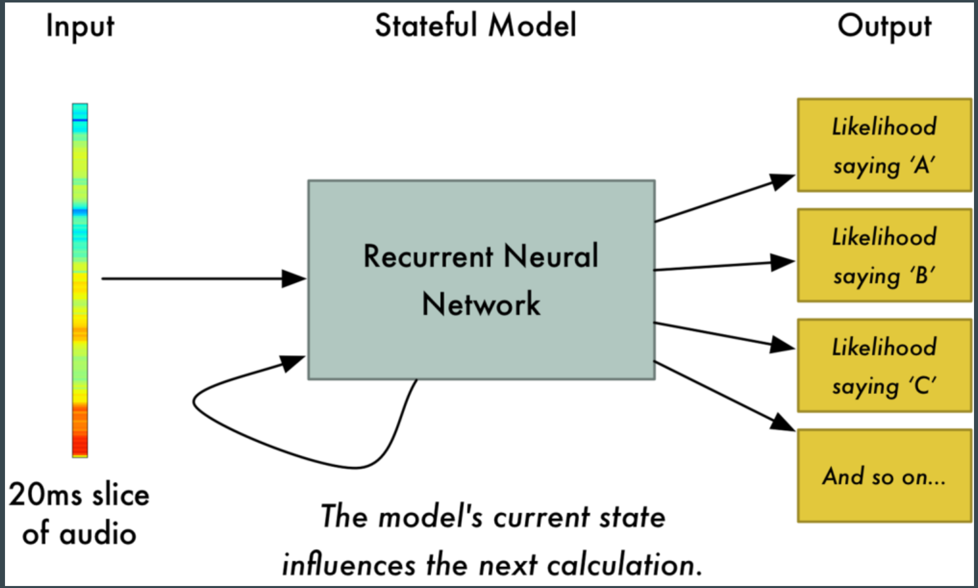
Speech Recognition

Hardware Used

- Raspberry Pi 3 B+
  - “Low cost, credit-card sized computer”
- Jounivo USB Microphone
  - 16-bit depth
  - Sampling rate of 48kHz
  - Omnidirectional

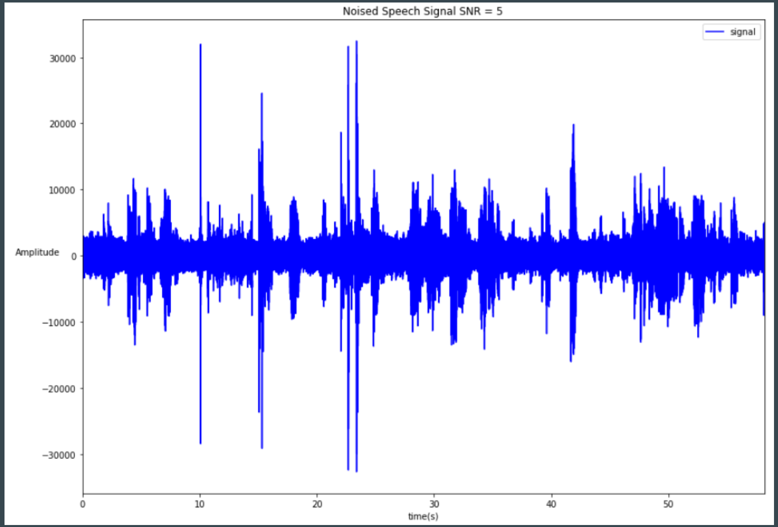
How Speech Recognition Is Typically Done

- Neural Networks
  - Series of algorithms to recognize relationships within data
  - Mimics the way the brain operates
- Hidden Markov Layers
- Requires lots of training
  - Deep Machine Learning Process



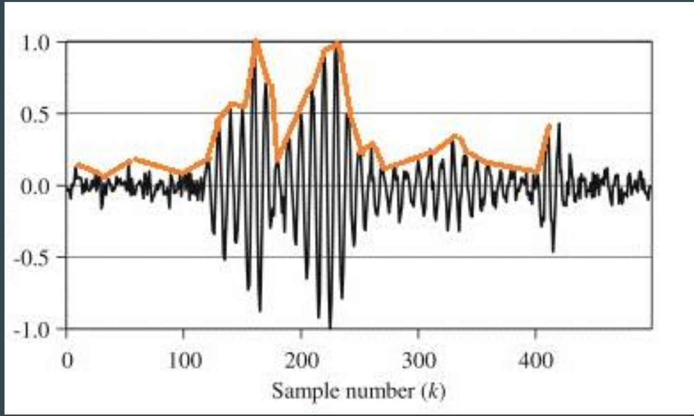
How We Accomplished Speech Recognition

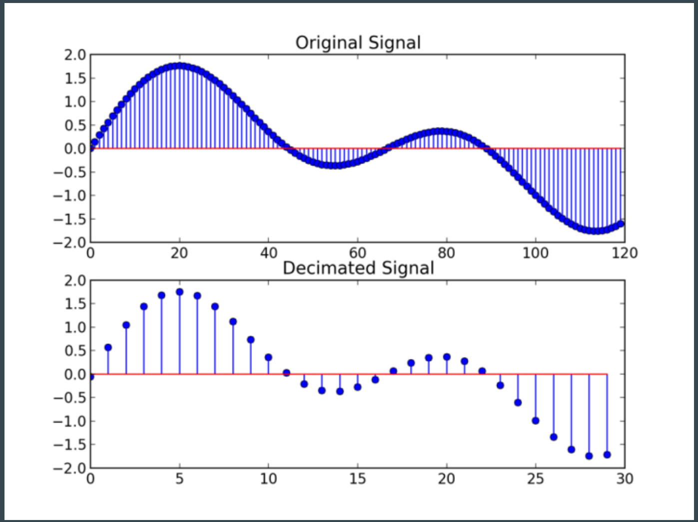
- Two main parts in our algorithm:
  - Speech Detection
  - Speech Analysis
- Detection
  - Finding voice activity within an audio signal
- Analysis
  - Figure out a way to “define” a word



Speech Detection

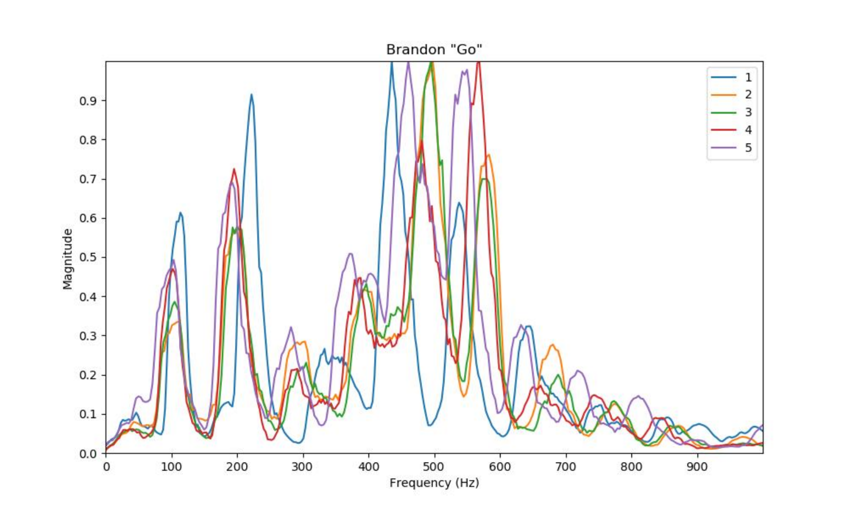
1. Normalize audio amplitude
2. Low Pass Filter
   a. 8th order Chebyshev Type 1, IIR
3. Downsample audio (16kHz)
4. Remove non-speech within the signal
   a. Noise floor & moving average

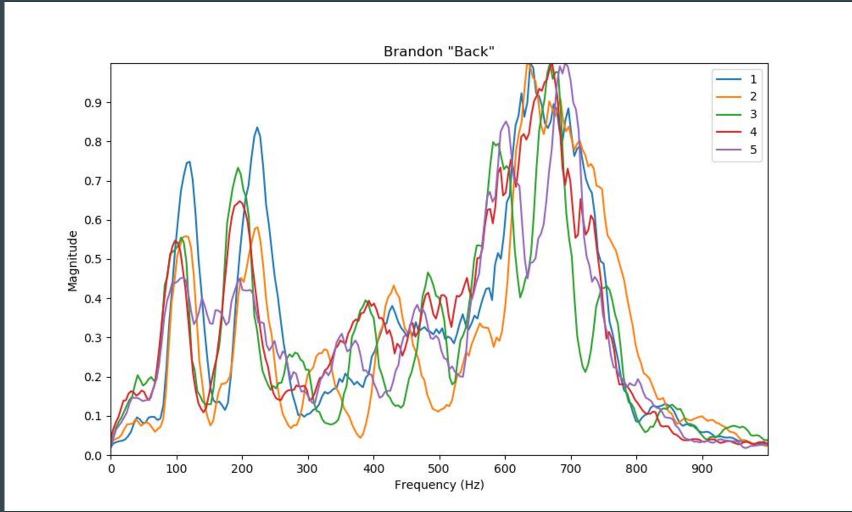




Speech Analysis Trials

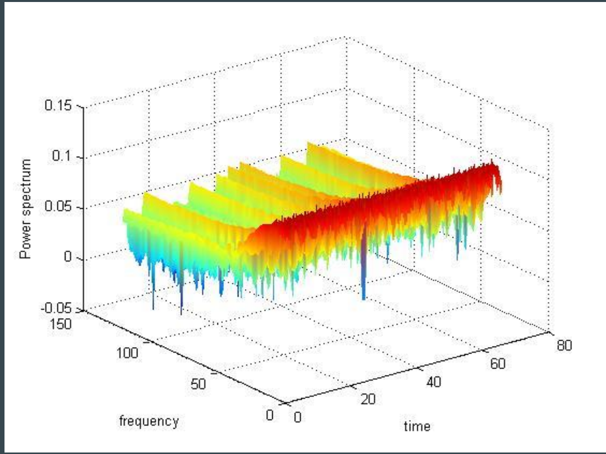
- Originally just used the Fourier Transform (FT)
  - Decomposes time function into its frequency components

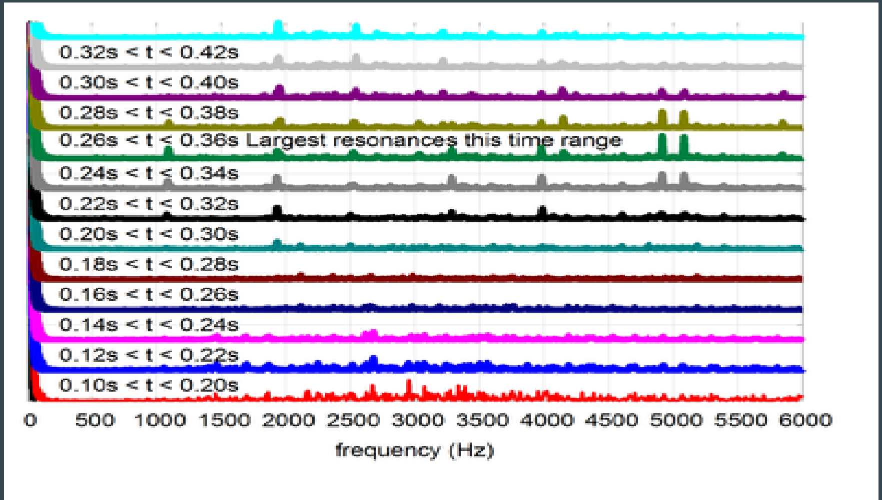




Short-time Fourier Transform

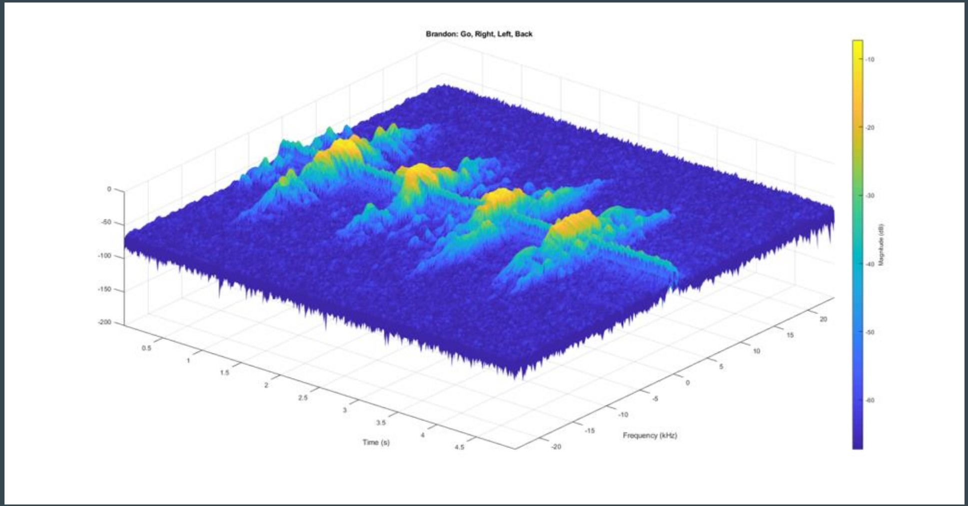
1. Frequency vs. Time vs. Magnitude
2. Used to determine the sinusoidal frequency and phase of sections of a signal as it changes with time
3. Used to divide a longer time signal into smaller sections, of equal length
4. Computes the Fourier transform of each section separately
5. Creates the spectrogram plot





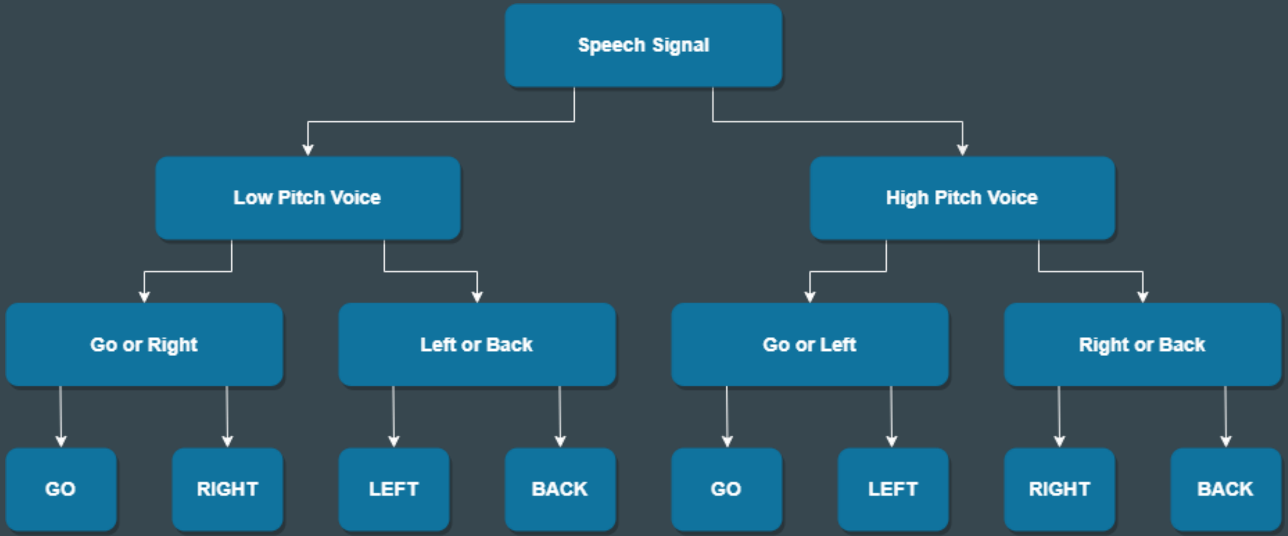
Speech Analysis

1. Short-time Fourier Transform
2. Spectrogram
3. Compare and contrast

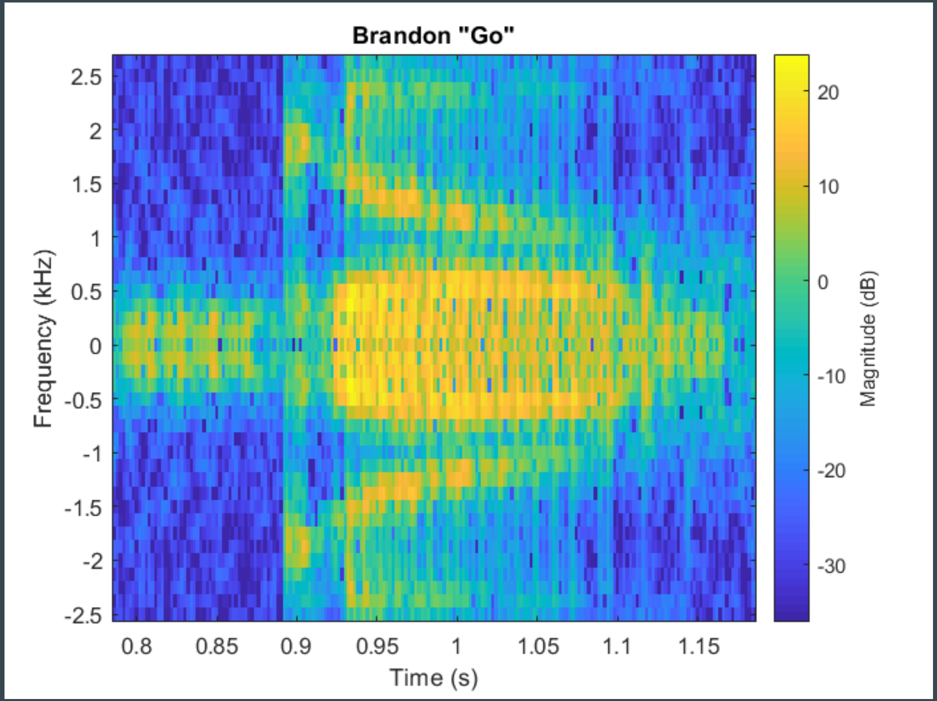


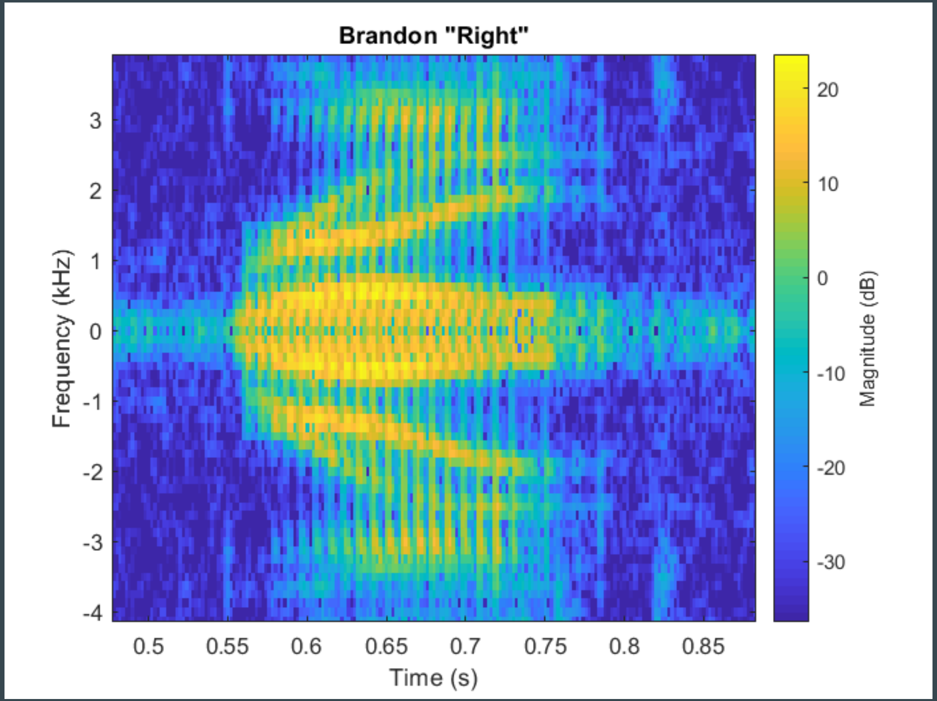
Analyzing Processed Speech

- Split algorithm into two branches based on pitch of the speaker
  - Fundamental Frequency of Men vs. Women
- Take four commands and make two groups of two commands
- Chooses one of the two remaining options

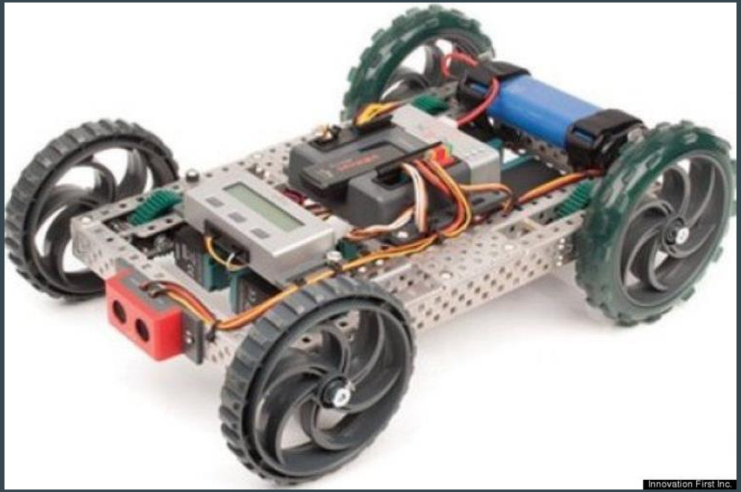


Spectrograms



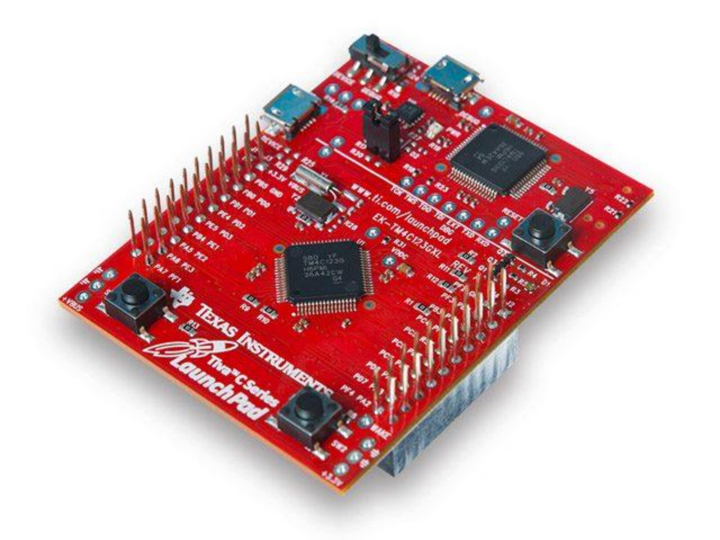


Robot



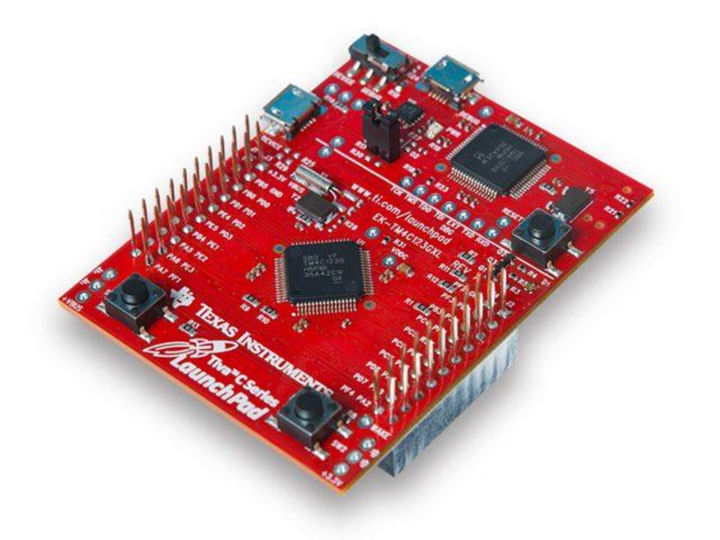
Hardware Used

- Tiva C (TM4C123G) Launchpad
  - As low as 370 µA/MHz
  - 16 PWM outputs
  - 8 UART
  - Up to 80MHz
- Vex 2 Wire Motor 393
  - All motor specifications are at 7.2 volts
- Vex Motor Controller 29
  - 1ms, full reverse
  - 2ms, full forward
  - 1.5ms, neutral



Hardware Used Continued

- Vex NiMH Battery
  - 7.2 V
  - 3000mAH
- YwRobot Breadboard Power Supply
  - Input Voltage: 6.5 V to 12 V (DC)
  - Output Voltage: 5V
  - Maximum Output Current: 700 mA

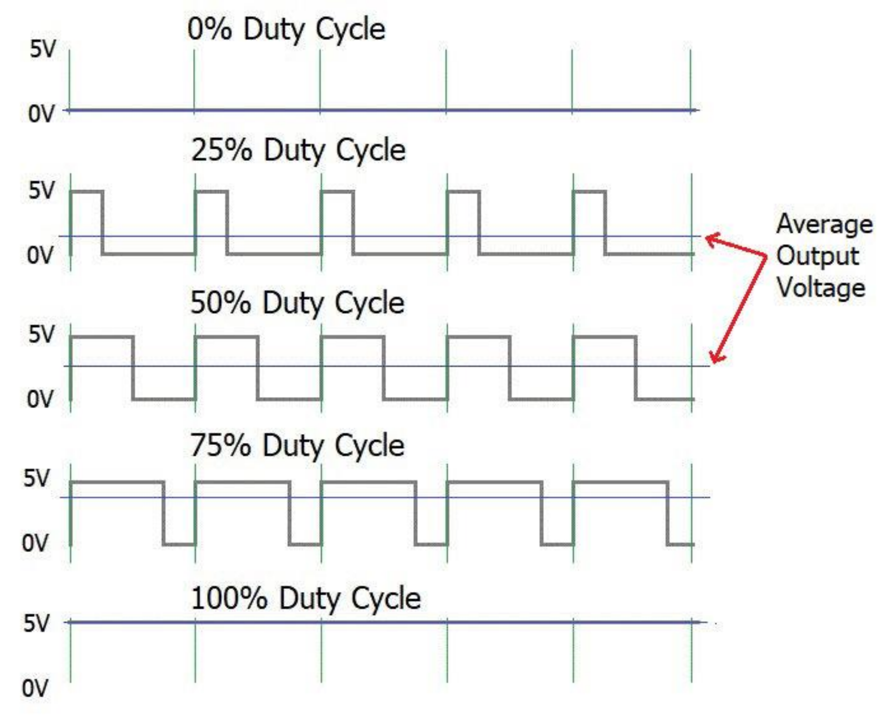


Software & Peripherals Used

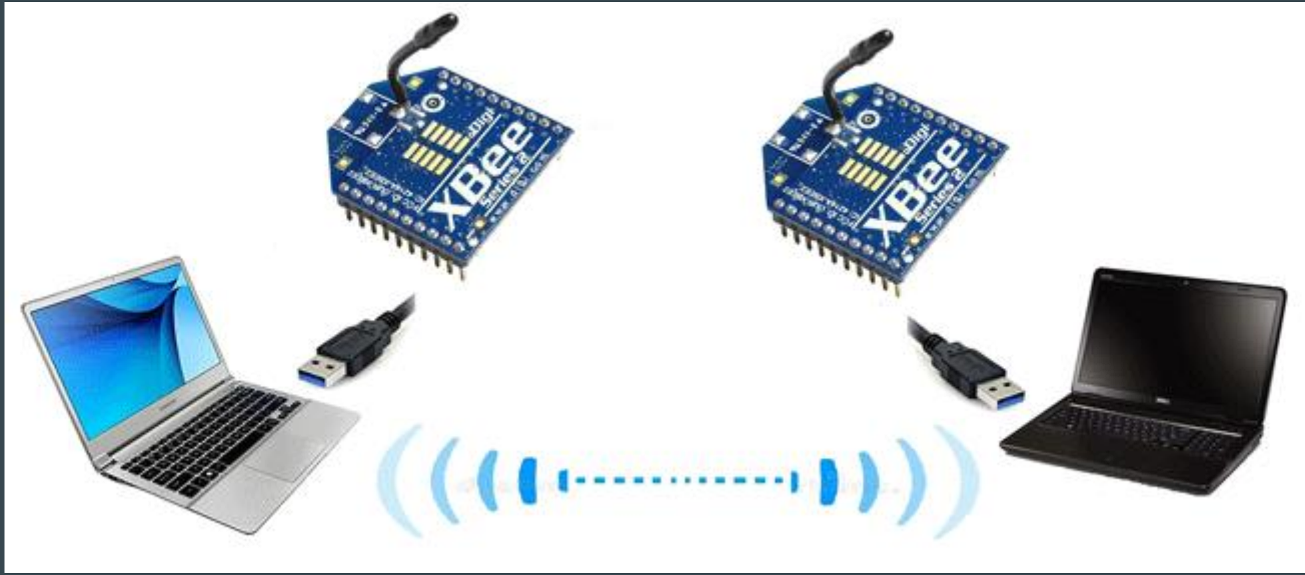
- Texas Instruments Code Composer Studio
    - Two Motion Control Modules
        - 8 high-res PWM outputs
    - Serial Connectivity
        - Universal Asynchronous Receiver/Transmitter (UART).
- Separate movement functions
    - goStraight(), goBack(), turnLeft(), turnRight(), and stop()
    - Get the needed movement by changing a variable that gives us the proper pulse width

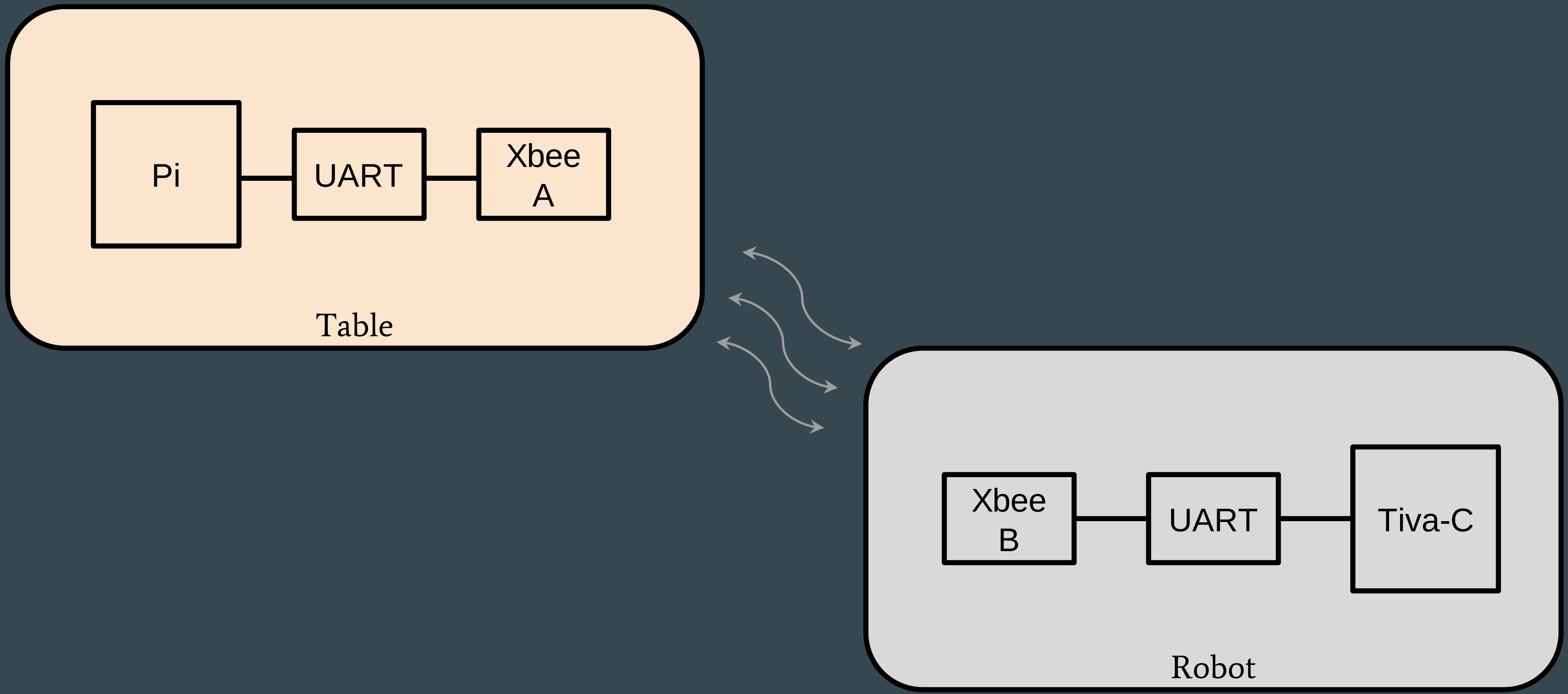
Pulse Width Modulation (PWM)

- Method of encoding analog signal levels
- Duty cycle of square wave is modulated to encode analog signal
- Since the programmed frequency is 55HZ and the period is 18.2mS, dividing that by 1000 gives us a pulse resolution of 1.82µS. Multiplying that by 83 gives us a pulse-width of 1.51mS.



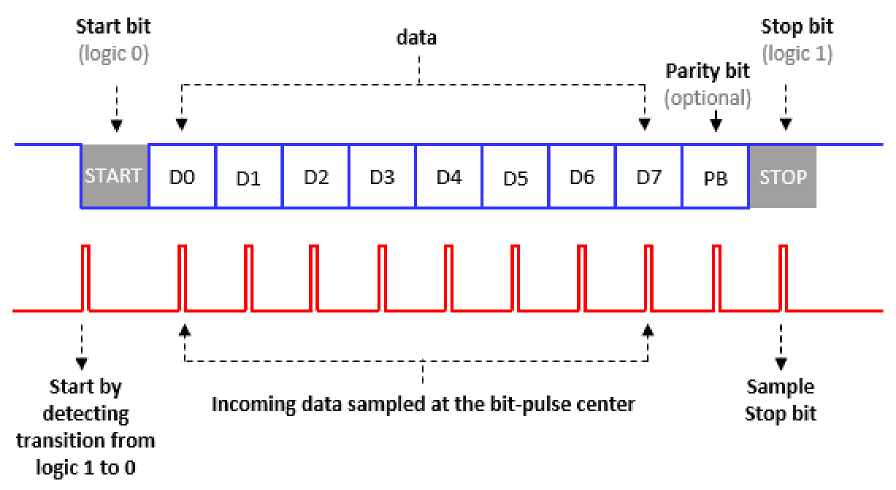
Wireless Communication



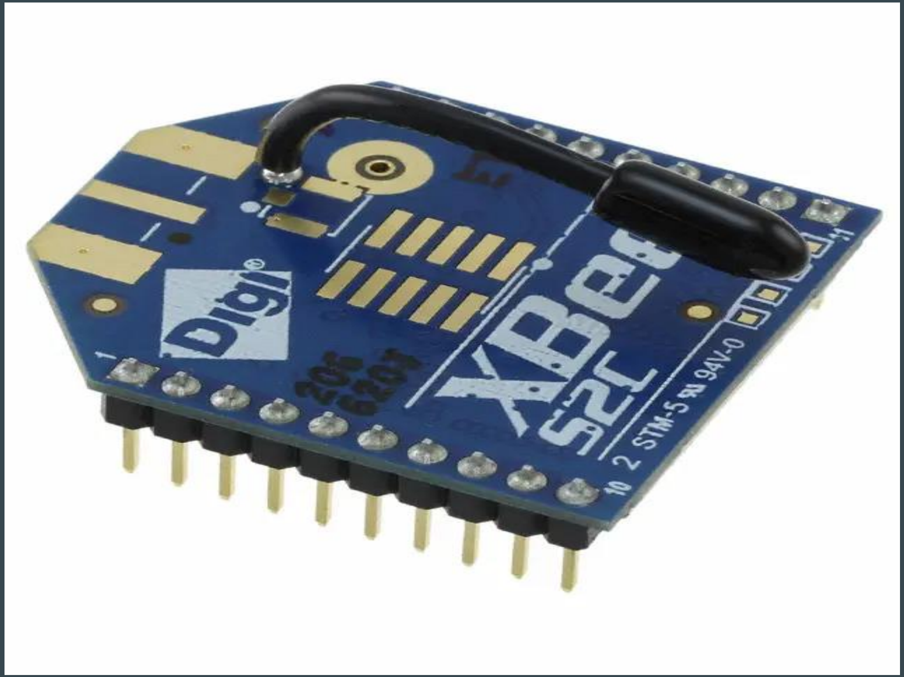

Pi
UART
Xbee
A
Table
Xbee
B
UART
Tiva-C
Robot

Universal Asynchronous Receiver/Transmitter (UART)

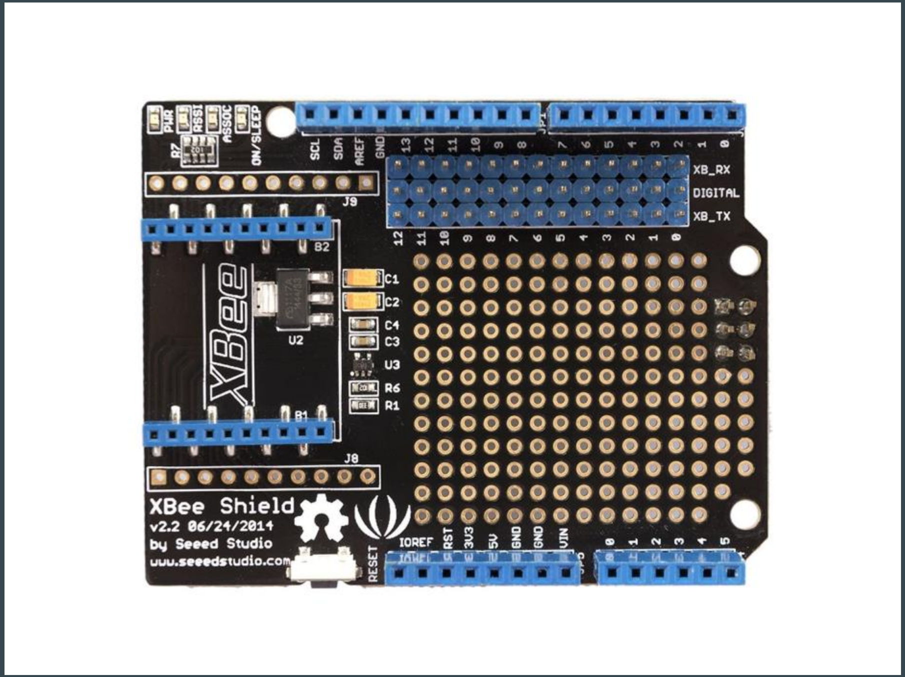
- Main purpose is to transmit and receive serial data
- Data flows from the Tx pin of the transmitting UART to the Rx pin of the receiving UART
- Transmit data asynchronously
  - Uses start and stop bits



Hardware Used



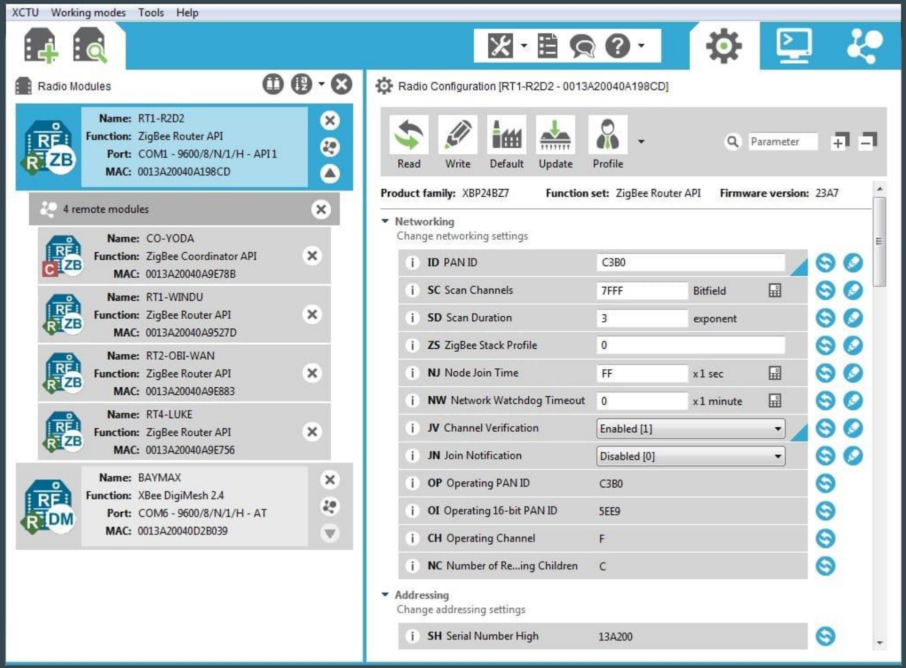
Two Xbee S2Cs



Xbee Shield v2.2

Configuration and Testing Software Used

- XCTU
  - Free next generation platform for Xbee's
  - Provided by Digi (manufacturer of Xbee)
  - Compatible with Windows, Mac OS, and Linux
  - Graphical network view for simple wireless network configuration
  - API frame builder is a simple tool to allow the building of API frames
- Used for testing purposes when learning to use the XBee modules
- Allowed configuration of our XBees for easier back-and-forth communication



How Wireless Capability Was Achieved

- Raspberry Pi:
    - Python library: *serial*

1. Initialize a serial port
2. Create a message ‘msg’ to send
3. Write ‘msg’ to serial port

```
import serial

ser1 = serial.Serial()
ser1.baudrate = 9600
ser1.port = 'COM5'
ser1.open()

if voice == 'go':
    msg = bytes('g', 'utf-8')

ser1.write(msg)
ser1.close()
```

- Tiva-C:
    - Initialize the UART
    - Take in a single char from receiver
    - Perform action of received character
        - ‘g’ = go
        - ‘l’ = left
        - ‘r’ = right
        - ‘b’ = back
    - Echo the received char back through the transmitter
        - Lets the voice recognition system know the command has been completed and ready for a new one

Characterization of System (Robot)

- Battery 3000mAH
  - Xbee S2C - 45mA
  - Tiva C - 30mA
  - Vex Motors
    - Max current - 3A, we will not use full forward/full reverse speed
    - Idle current - 12mA
- System will be using motors about 10% of the time
  - Battery capacity - 3000mAH, discharge rate - 3075mAH
  - Run time - less than 1hr (0.97hrs)
- System will be idle for about 90% of the time
  - Battery capacity - 3000mAH, discharge rate - 99mAH
  - Run time - about 30hrs
- Total system run time
  - About 27 hours

Characterization of System (Speech Recognition)

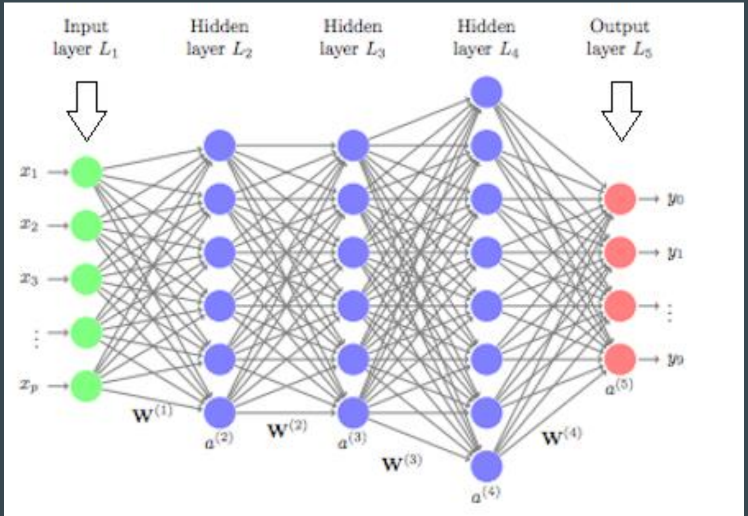
- Overall, very accurate within the group members
  - Fairly accurate for other students as well
- Main things that affect accuracy:
  - Accents
  - Distance from the mic / volume of the speaker
  - “Clarity” of the speech

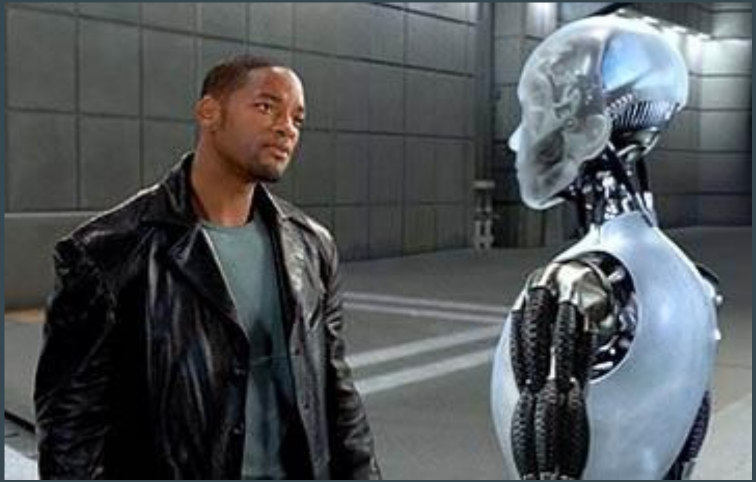
Deliverables

- Voice-controlled robot
- Stationary system on table/desk
  - Utilization of wall power
  - Operated by Raspberry Pi
- Offline Speech Recognition
- Robot performs basic movement commands;
  - Go, Back, Right, Left
- Wireless capability
- Characterization of System
  - Power consumption comparable to baseline values, SNR estimation from classrooms/labs
- Always Listening

Future Work and Improvements

- Improve the robot
  - Add more commands
  - Add sensors for automation
  - Improve the physical design
- Allow internet access
  - Larger database for speech recognition
- Use Deep Learning
  - Neural Networks
  - Improves accuracy
- Add more microphones
  - Stronger noise reduction





References

- [Xbee S2C]
- [YwRobot Power Supply]
- [Vex Battery]
- [PWM and UART]
- [UART2]
- [XCTU]
- [Short-time Fourier Transform]
- [Time-domain audio processing]

Questions?