

# Dennis Moore

<https://github.com/dennis-moore>

Email : denmoore93@gmail.com

Mobile : 281-608-9655

## EDUCATION

- 
- **Texas A & M University** College Station, TX  
*Bachelor of Science in Electrical and Computer Engineering* Aug. 2011 – Dec. 2015

## EXPERIENCE

- 
- **SOE** Houston, TX  
*Embedded Software Engineer* May 2018 - Present
    - Optimized large C code base to minimize code size, improve performance and remove warnings. Reduced code size by 25%, build time by a factor of 6 and removed 40,000+ warnings.
    - Fixed C makefile to cross-compile code with the ARM GCC toolchain inside a Docker container.
    - Developed Python scripts to integrate with Unix tools and generate metrics on the code base. Added these tools to the Docker + Gitlab CICD environment to auto-generate metrics every build.
    - Improved performance of navigation system in embedded RTOS environment. Configured sensors over I2C.
    - Updated project to log data over the serial port. Analyzed data with Jupyter Notebooks to design low-pass filters.
  - **Cognizant Technology Solutions** Dallas, TX  
*Software Engineer* Jan 2016 - April 2018
    - Worked in an Agile environment to design, develop, and test production software. Developed and maintained various Java, Python and C# applications that were used across many teams.
    - Developed multiple Java applications that utilized REST APIs to improve upon our CICD environment.
    - Maintained Windows virtual machines running Jenkins used for regression testing and CICD. Developed Java and Python code to automate parts of deploying new VMs. Provided VM support during production deployments.
    - Created a couple C# web apps hosted internally on a Windows VM. Both apps used .NET, Javascript and MySQL.
    - Worked on new ways to enhance the Selenium and TestNG testing framework through software innovations. Wrote a Java application to speed up regression execution, and improve testing coverage and efficiency.
  - **SOE** Houston, TX  
*Embedded Software Engineer Intern* Summer 2015
    - Defined a low level interface between two processors to allow for SPI communication between the devices. Designed a level-shifter to match their voltages. Implemented error checking to throw away bad packets.
    - Researched and experimented with CAN bus communication between a Raspberry Pi and TI C2000.
    - Wrote Python code to plot real-time magnetometer data coming over serial port.
  - **AMBER Robotics Lab** College Station, TX  
*Student Researcher* May 2013 - July 2014
    - Proposed a hardware system to test a nonlinear cruise controller. The idea was to port the controller to an Arduino, and use the Arduino to control a gas powered RC car. This allowed the simulation to be tested in the real world.
    - Interfaced optical encoders with the Arduino over I2C to compute wheel velocity. Applied a digital filter to this.
    - Helped with the electrical design(microcontroller and sensors). Wrote a portion of the control system to feedback wheel velocity based on the encoder readings and drive the motors with PWM signals.

## PROJECTS

- 
- **NCOB:** *Spring Java web application for connecting and remotely controlling robots through an online interface*
    - Frontend created with HTML, CSS, jQuery and Thymeleaf. Backend created with Spring MVC and MongoDB. Deployed with Docker and Gradle.
    - C client code uses ZeroMQ to communicate with JeroMQ message broker on the web server. Google protocol buffers are used for message serialization.
  - **AggiE-Challenge:** *Autonomous UAV that explored and mapped an unknown environment*
    - Setup Robot Operating System(ROS) on the Raspberry Pi. Developed TCP socket code in Python to communicate with a host machine. Later added ROS to the host machine to ease wireless communication and SLAM.
    - Interfaced ArduPilot flight control system with the high level controller on the Pi over Serial. Developed controller on the Pi to route ArduPilot sensor data, switch between flight modes, and drive the quadcopter with user input.
    - Wrote device driver in C to interface LiDAR sensor with the Pi. Later ported to ROS to ease system integration.
    - Developed Python code to visualize real time IMU data as a 3D cube to represent the attitude of the quadcopter.
  - **Senior Design:** *Microcontroller system for cars designed to monitor the environment and driver vitals*
    - Wrote serial communication drivers in Python and C to communicate between processors. Implemented memory mapped IPC to communicate between processes.
    - Sampled EKG sensor with ADC to determine heart rate information. Interfaced LiDAR sensor over I2C to receive distance information. Utilized pin interrupts and Hall effect sensors to calculate wheel velocity.
    - Developed a control system based on kinematic equations and sensor inputs to send alerts over the serial link to the bluetooth interface.