|  |
| --- |
| University of Texas at Arlington |
| EE6314 Final Report |
| Laser-Beacon Indoor Navigation and Control System |

|  |
| --- |
| Fall 2018 |

Table of Contents

[Overview 2](#_Toc531616839)

[Concept of Operations 2](#_Toc531616840)

[Laser Beacon 2](#_Toc531616841)

[Robot 7](#_Toc531616842)

[Software 10](#_Toc531616843)

# Overview

The Fall 2018 term project for EE 6314 was to design/develop, as a class, a robotics system built around a laser-beacon-based internal navigation concept. The idea behind this project was to design and demonstrate a novel navigation system that is not reliant on inertial signals, odometry, or global positioning systems. The robot is built to navigate around a space (in 2-D Cartesian coordinates) with fixed base stations providing signals from which position and location details can be derived. The navigation space is determined by the location and configuration of these base stations. Each base station, or beacon, operates a rotating laser. As each robot in the space collects signals from the rotating laser beacons, it is possible for the robot to compute its absolute position in the space via triangulation of the visible beacon signals.

# Concept of Operations

This system is intended to allow a robot to navigate around a fixed area with a high level of position accuracy. The top-level components that allow this system to operate are: a user-interface control station, three laser-beacon base stations, and a robot equipped with an omni-directional laser sensor and pulse-processing algorithms (see Figure 1). These components all interact/operate wirelessly via Digi International XBee S1 Radio Frequency (RF) modules.

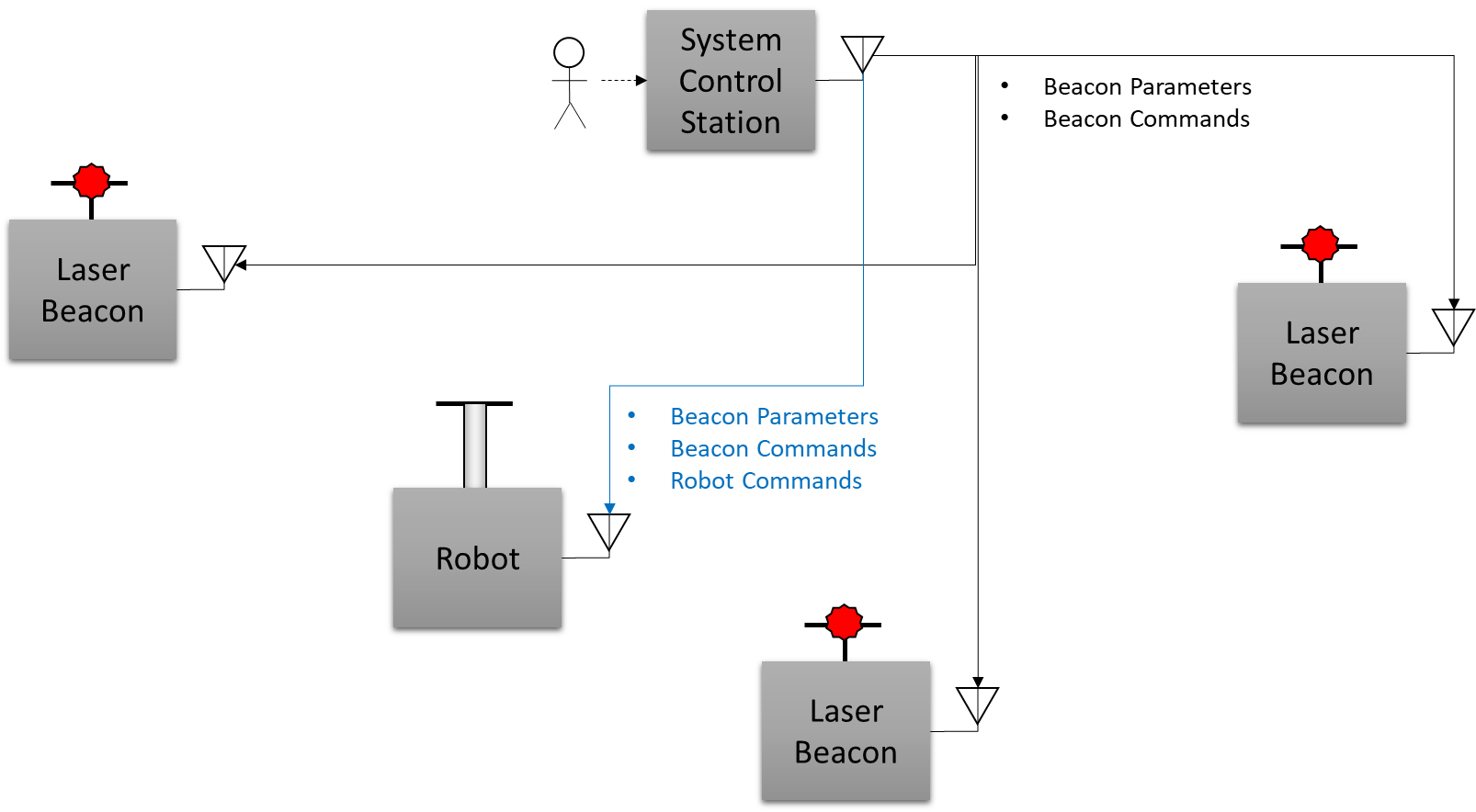
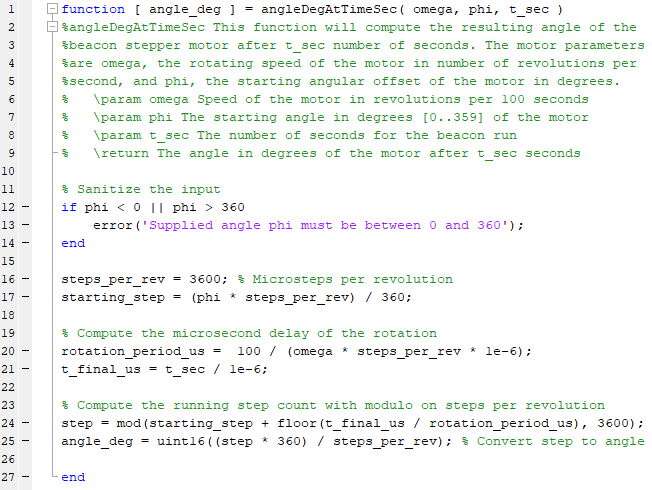


Figure 1 System Architecture

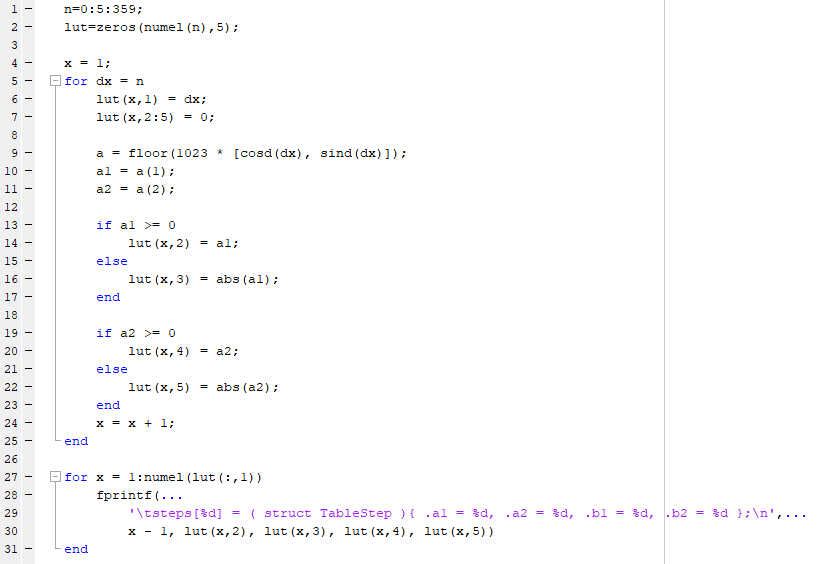
# Laser Beacon

The Laser Beacon component of the system is designed around the use of stepper motors to drive the rotation of the laser. The system utilizes a TI SN754410 dual-channel bipolar motor driver. This driver chip is capable of driving two separate DC motors or, as is used in the beacon design, driving a single stepper motor.

The laser beacon subsystems utilize micro stepping algorithms to provide a smoother, more-continuous sweep of the laser (when compared with full-step operation). The motors being utilized operate at 1.8 degree full step increments; if a laser is mounted on the motors and is rotated in 1.8 degree steps, the distance between fixed laser pulses will be too great for continuous detection by the robot’s sensors. To mitigate the risk of the robots missing laser pulses, the motor outputs are pulse-width modulated to micro step the motor’s internal induction/teeth mechanisms. The micro stepping is done in 0.1 degree increments; this means that it takes the motor 3600 discrete steps to complete a full revolution. The following function details the computation for determining the position of the sensor (in degrees) after a set amount of time:



The microstepping is achieved in software by using Pulse Width Modulation (PWM) to provide ratios as the input to the motor driver chip. These PWM signals allow for teeth inside of the motor to be held in-between positions based on simultaneous forces being applied to the induction coils. In software, the PWM values needed for each step of the motor are stored in an indexable lookup table. Each index of the table tells the system what the inputs to the motor driver need to be. The look up table approach is used in order to avoid costly trigonometric functions being needed at runtime. The values in the lookup table are autogenerated by the following MATLAB script:



The beacon hardware is comprised of the aforementioned motor driver, a photo-interrupter sensor, and a TI M4F microcontroller. The microcontroller uses PWM outputs on pins PB5, PB6, PE4, and PE5 to control the motor driver, and it uses pin PE3 as an analog input for reading the value from the homing sensor (the photo-interrupter) The homing sensor is a photo-interrupter package consisting of an infrared LED and a phototransistor; when a disruption is passed in between the LED and the transistor a low reading is supplied to the microcontroller.. The system power is provided by 8 AA batteries in series (providing an initial maximum voltage of about 12V; a voltage regulator is utilized to regulate 5V from the battery input to act as a 5V power line for the microcontroller. The 5V output from the voltage regulator is supplied to the M4F microcontroller via the VBUS pin. The non-regulated power from the battery is supplied as VCC2 motor power to the motor driver chip. A shared common ground is used between all components. See Figures 3 and 4 for Laser Beacon schematics.

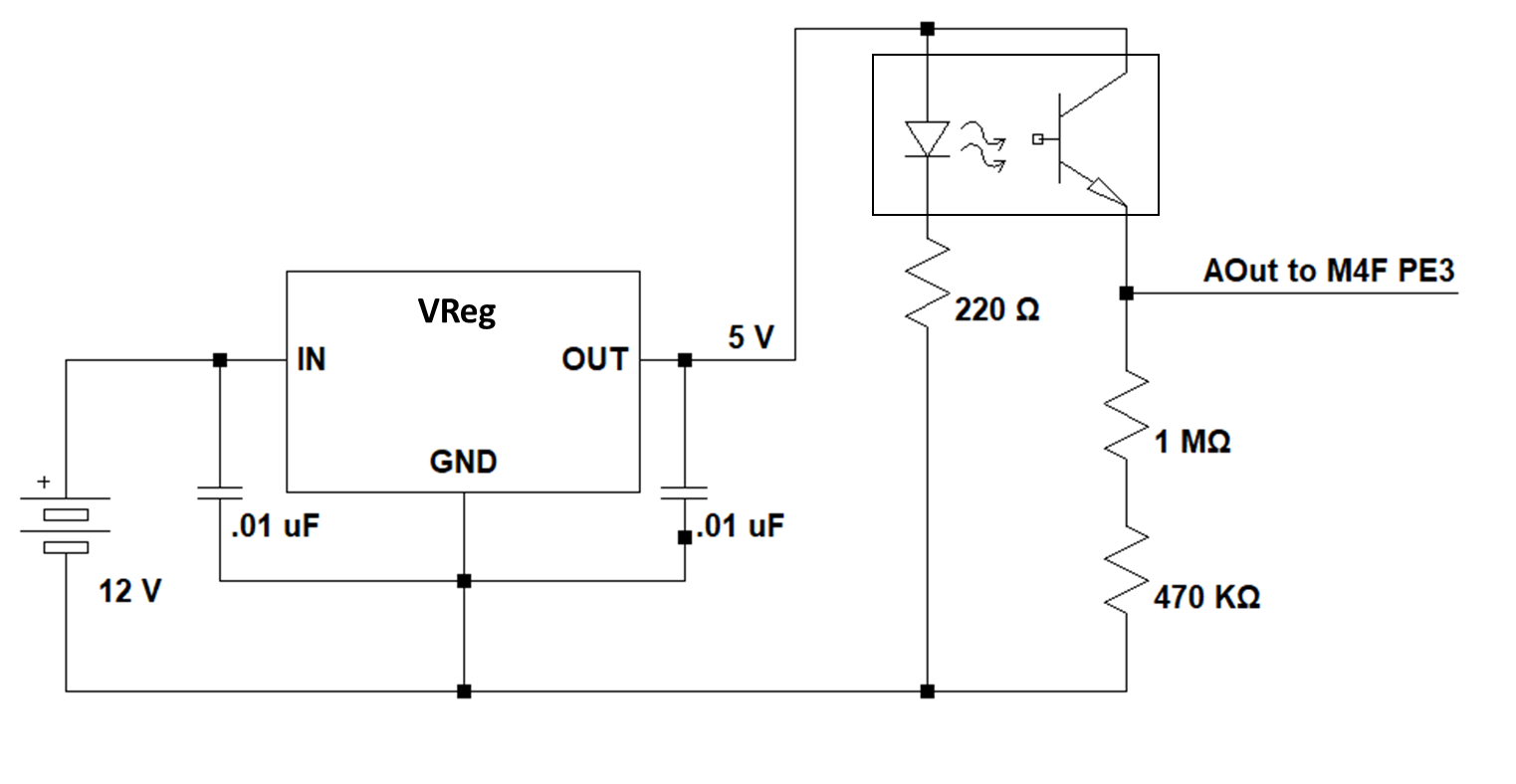


Figure 2 Homing Sensor Hardware

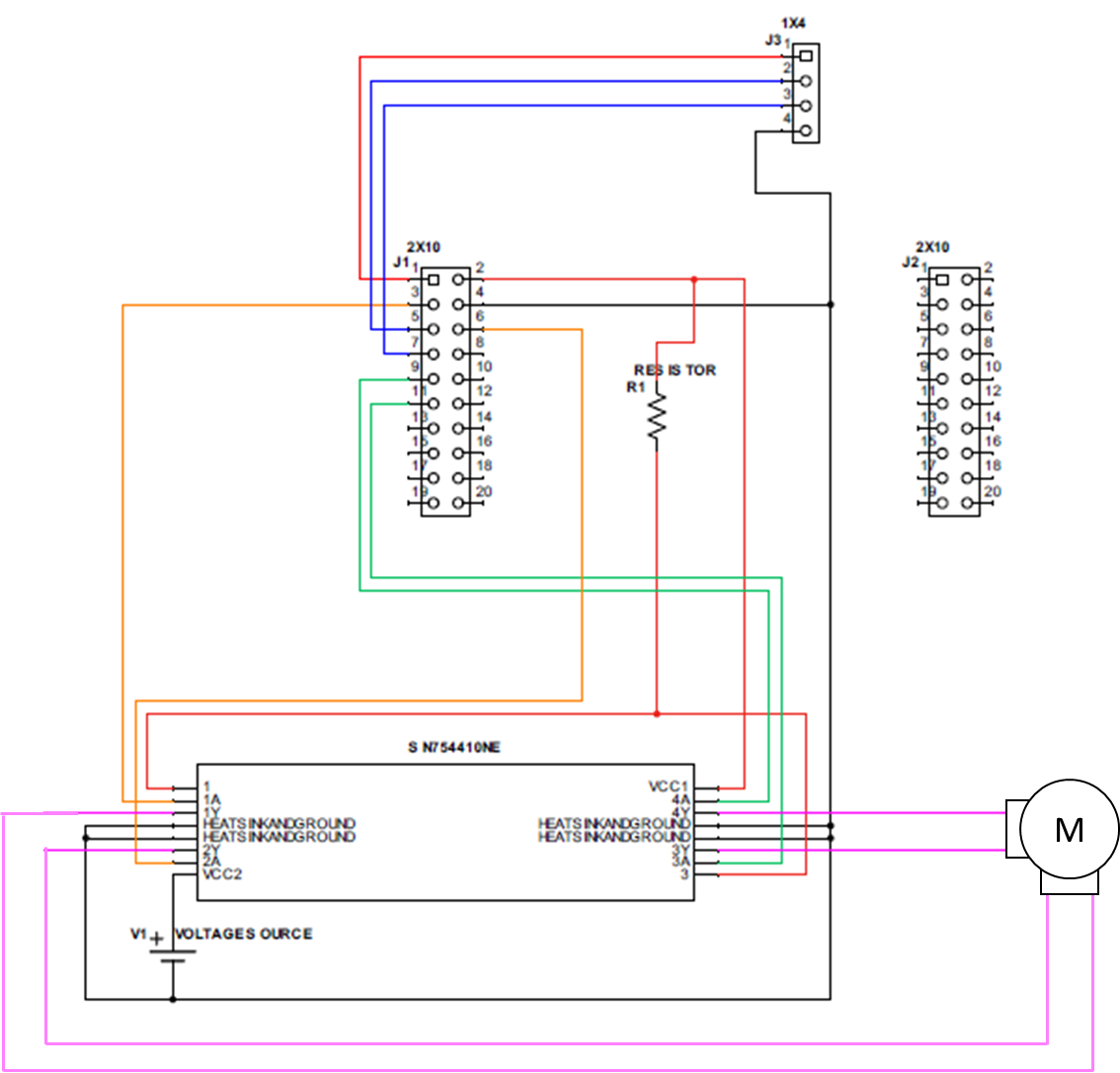


Figure 3 Beacon Motor Hardware

The beacon system uses an XBee S1 module for wireless communication to the other components of the system. The XBee S1 module is connected to the M4F microcontroller via the UART 1 module on pins PB0 and PB1. This communication link to the beacon system is used to configure each beacon (each beacon in the environment has a hard-coded, pre-configured unique identifier). The beacons are configured with a speed and a starting angle. The speed is supplied to the beacon in units of revolutions per hundred seconds. This unit is used as an integer scaling mechanism so as to avoid utilizing floating point registers in the hardware; for example, a speed of 100 is equivalent to 1 rev/sec. The communications link is also used on the beacons to tell each unit to start, stop, and reset (i.e. go home). The following table lists the commands the beacon utilizes via the XBee wireless module:

|  |  |  |
| --- | --- | --- |
| Command | Parameters | Description |
| set omega <angle> | <angle> = rotations / 100s | Sets the rotation speed of all listening beacons |
| set omega <angle> <id> | <angle> = rotations / 100s  <id> = 0-9 device ID | Sets the rotation speed for all devices with IDs matching <id>  Robot will also listen in on this and store beacon speeds |
| reset |  | All listening beacons will go to the home position |
| start |  | All listening beacons will start spinning at their respective omega |
| Stop |  | All listening beacons will stop spinning |
| Location |  | Will home all beacons, then it will wait 1s for all beacons to complete homing. Then beacon with id=3 will transmit a start command, so all beacons start at the same time |

# Robot

The robot built for this system was comprised of pieces from a MakeBlock robot kit. The circuitry and hardware build for the robot is almost identical to the design utilized for the laser beacon system. The main difference is that instead of using a dual channel stepper motor (as used in the beacon system) the robot is built with two DC motors for the system wheel drives. The benefit with using the TI SN754410 motor driver is that it can drive the motors in both architectures (either one stepper motor or two DC motors). Like all other subsystems in this project, the hardware is integrated with an embedded TI M4F microcontroller. See Figure 4 for the design schematic of the robot motor drive.

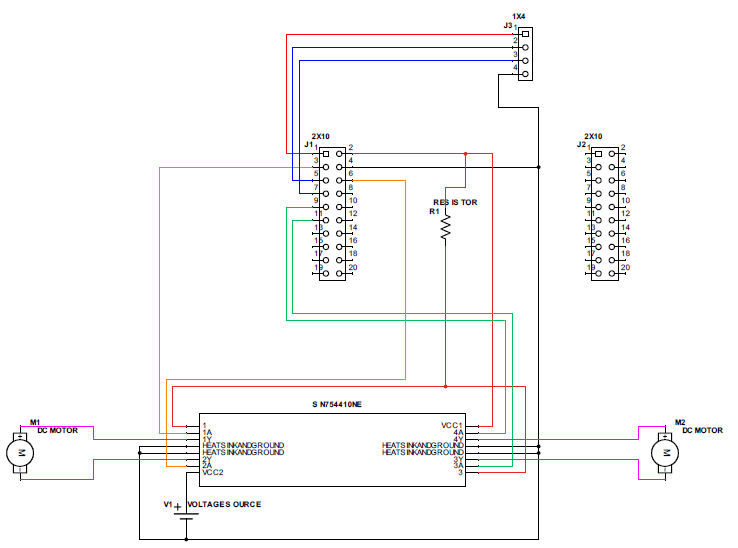


Figure 4 Robot Motor Drive Hardware

The software used by the robot to drive its motor uses PWM to control the speed and direction of the motors. The design uses the same four microcontroller pins used by the beacon system for PWM motor drive. One pair of the PWM pins drives the left motor (PB5 and PB6), and the other pair drives the right motor (PE4 and PE5). Separate functions were generated to control the motor direction of the two DC motors. By having this control, the robot can be rotated in place in either direction, as well as move forward and backwards.

Due to limitations of the location calculation time, it was desired for the robot to run at 50% duty cycle to slow down its movement. This caused an issue when starting from a stop, or other high-load instructions. To mitigate this, the robot detects when high-load direction changes are instructed and runs at 100% duty cycle for 100ms, before reducing to cruise speed.

The robot is built using a laser sensor that is designed to sense pulses from the rotating beacons as the laser passes across the robot’s sensor. The laser sensor is designed using a photo-resistor connected to pin PC7 on the microcontroller via a voltage divider. PC7 is configured as an analog input with an attached comparator used to provide the embedded software with event interrupts whenever a pulse is received (i.e. the voltage reading drops below 1.5V). The laser sensor itself is an opaque, manufactured tube with the photo-resistor on the bottom. Above the resistor is about 4 inches of opaque beads used for reflecting received light down the tube onto the sensor.

Three beacons are identified based on whether the time value obtained from their pulse detections lie within the respective beacon’s frequency configuration plus or minus the error band value. In order to identify the beacon robot do not move till it identify the beacon sets based on their frequencies. Having known the initial position of the beacons and based on the beacons available, respective angles are calculated to find the location of the robot. Three beacons are placed in a triangular manner with an overall degree of 360; a minimum of two beacons are required to detect the location of the robot.

When the robot identifies its original location, it moves forward for 2000ms and calculates it’s heading angle based on the change in position. It then calculates the target angle based on the current angle and position. Then the robot will rotate w=for a set amount of time to change its angle to match the target angle. After the rotation is complete, the robot will start moving forward towards the target. The heading angle and position is checked in fixed intervals to make sure the robot is on track, if it is not a rotation will take place to compensate for the error. The robot will stop when it reaches the target.

Just as with the laser beacon subsystem, the robot uses an XBee S1 module for wireless communication. The communications hardware and software design is complete reuse between the beacon and robot subsystems. The S1 is connected to the robot hardware via UART module 1 on pins PB0 and PB1. The communication link is used to tell the robot the configuration parameters of the beacons in the environment as well as to provide commands to the robot to start, stop, reset, and query location. The following table lists the commands the beacon utilizes via the XBee wireless module:

|  |  |  |
| --- | --- | --- |
| Command | Parameters | Description |
| Robot Forward |  | Moves the robot forward |
| Robot Left |  | Rotates the robot counter clockwise |
| Robot Right |  | Rotates the robot clockwise |
| Robot Stop |  | Stops the robot |
| Robot Wiggle |  | Rotates the robot clockwise for 1s then rotates it counter clockwise for a short period of time.  The aim was to align the caster wheel (does not work, too finicky) |
| Start |  | This tells the robot that beacons have begun spinning and to start storing laser interrupts.  Currently, after 20 time points, the robot will transmit its current location to everyone. |
| Set omega --- |  | Robot listens in to all beacon speed changes and stores the values |

# Software

All of the software source code for this system is written in the C99 (ISO/IEC 9899:1999) programming language, with minimal pieces of code written in the ARMv7-M Thumb instruction set, all for the Cortex-M4 processor. All the code was developed and built using the TI Code Composer Studio Eclipse Workspace, and can be uploaded to the MCU with the TI Flash Programmer. One of the goals of the team on this project was code reuse; the team found opportunities for writing common code for wireless communication processing and motor drivers. This common code (in union with unique, application specific code for each subsystem) was used to pull together the complete binary packages for each part of the system.