BOE-BOT

*(Board Of Education Robot)*

**Objective**

The main objective of this project to design a BOE-BOT with the features as Motion, Obstacle Avoidance, and Communication. The design is based on a Texas Instruments TI-MSP432P401R Launchpad with ARM Cortex-M3 processor. The main task for the BOE-BOT is to traverse a path having obstacles and collect data from pre-paced fixed Bluetooth sensor tags. The data from sensor tag consists of temperature, humidity, and pressure measurements at certain places. Figure 1 shows a prototype design with components and Figure 2 shows a general working BOE-BOT

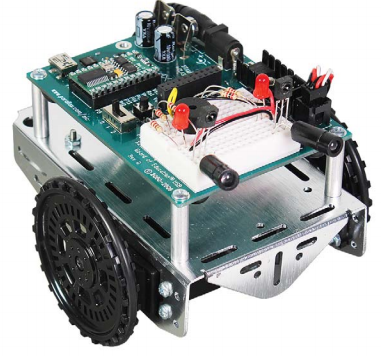


Figure 1: BOE-BOT Prototype Design

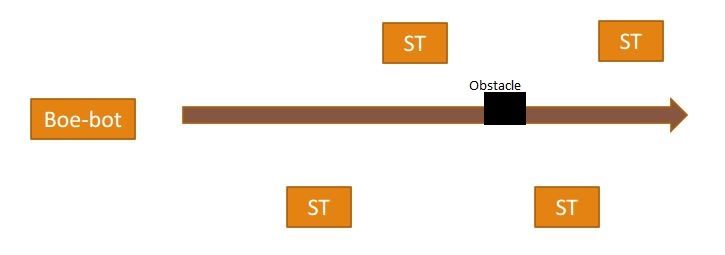


Figure 2: BOE-BOT general working

**Motion**

For the motion of BOE-BOT, two Parallax Continuous Servo Motors are used with Vcc as 5V. The Servo Motors operate on pulse width parameter. A pulse of width 1.5ms stops the rotation whereas a pulse width of 1.3ms drives motor in a clockwise direction and a pulse width of 1.7ms moves the motor in a counterclockwise direction. Each pulse width needs to be followed by a 20ms delay before the next pulse arrives in order to operate motor correctly. Figure 3 shows a Servo Motor used in this design.



Figure 3: Parallax Continuous Servo Motor

*Red Cable: Vservo = Vcc, Black Cable: GND, White Cable: data line*

**Obstacle Avoidance**

The feature of Obstacle avoidance will be implemented using an ultrasonic sensor. The module HC-SR04 will be used as an ultrasonic sensor. The sensor module operates in a trigger-echo way. The trigger is supplied by a pulse width of 10us which generates 8 40KHz pulses from ultrasonic transducer and transmits the pulses. The echo pin goes high as soon as an obstacle is detected and the pin stays high for an amount of time-related to the distance of the obstacle. The echo pin needs to be monitored for the echoed pulse and the time for which echo pin stays high needs to be measured. Now, the distance between obstacle and sensor can be using high pulse duration measured on echo pin and velocity of ultrasonic signal which is 340 m/s. Figure 4 shows an ultrasonic sensor HC-SR04. Equations (1) can be used for distance measurements.



Figure 4: An HC-SR04 ultrasonic sensor

*Shortest Range: 2cm = 116us*

*Longest range: 400cm = 23ms*

*Time > 23ms: no echo.*

*Min resolution: 0.3cm*

**Communication**

Texas Instruments has developed a Bluetooth CC2650 sensor tag to make implementation IoT easy. This sensor is useful to sense several features such as Temperature, Humidity, Pressure, light, and a 9-axis motion sensor. TI has also developed a Bluetooth CC2650 Launchpad kit which can be used to implement Bluetooth Low Energy (BLE) along with any other TI MCU Launchpad. Figure 5 and 6 show a CC2650 Sensor tag and a CC2650 Launchpad.

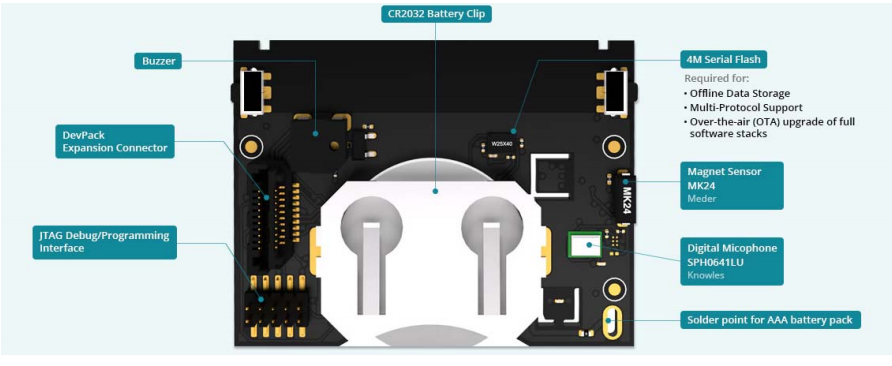


Figure 5: TI Bluetooth CC2650 Sensor Tag



Figure 6: TI Bluetooth CC2650 Launchpad

**Accelerometer and Digital compass**

The LSM303DLHC is a digital (I2C) accelerometer and digital compass(magnetometer) developed by Adafruit. The accelerometer allows you to measure acceleration or direction towards the center of the earth, and the magnetometer measure force, which is useful to detect magnetic north. Picture below shows a LSM303DLHC.



It can be used for applications like

* Compensated compass
* Map rotation
* Position detection
* Motion activated functions
* Free-fall detections
* Pedometer

**Summary**

The design approach we followed was to develop modules in CCS project and then add them to RTOS project based on threads and priorities. In CCS project, Servo Motors worked as expected based on duty cycle calculations according to datasheet. Stop, Clockwise rotation, and anticlockwise rotation were checked and confirmed as working correctly in normal CCS project. However, when same calculated configurations were used in TI-RTOS project, output was not as expected. Servos were moving in just one direction irrespective of configuration. This is the main challenge which we faced.

There were some challenges in interfacing the ultrasonic distance sensor module into this project but we overcame those hurdles and finally could generate stable number of cpu cycles for a given distance. Thus, the idea of a robot with eyes measuring the distance between it and the nearest object is reaffirmed with this humble but useful integration. We figured out how to interface interrupts with RTOS.

We got two modules working fine in CCS project but when we combined them with TI-RTOS tool then it din’t work as expected. The team is a seamless combination of talent and we gelled up pretty easily. The team co-ordination was good and everybody got chance to work on at least one module.

Given a chance to do the rework on this project, we would like to integrate the modules in a Non RTOS environment and we would also like to use Lidar module instead of Ultrasonic. Additionally, we were not able to interface compass module due to lack of supported driver libraries and documentation. Also, we decided the state diagram for avoiding an obstacle. The state diagram was also implemented, but we observed that the distance calculation of the sensor was not very accurate. The distance we showed in the video was according to the count of the timer A1, but actually we wanted to use the actual distance for deciding to choose the task.

Overall, this was a completely novel experience for us where we got to learn a lot of advanced concepts.