



EMBEDDED SYSTEMS

Assignment 5

CONSTRUCTION



- Carefully assemble your motor, acrylic disk, and breadboard to create your rotating platform.
 - The collar used for attaching the motor shaft to the acrylic platform is not built to withstand significant force. Use a box or other container to house your assembled components. Protect this at all costs!
- Attach the ultrasonic sensor to the front of your rotating platform by inserting it into the breadboard.
- Attach the compass sensor board somewhere on your rotating platform (also in the breadboard).
 - You will need to calibrate for true north so consider how you arrange it.
- Attach the rotating platform to the front of your large breadboard.
- Wire your MBED, OLED, motor and sensors to complete your radar hardware system.
 - The OLED should rest comfortably on your larger breadboard or off to the side of it.

RADAR SYSTEM



- Write an application that uses a FCFS scheduler to:
 - Rotate your moving platform 180° – sweeping the area in front of it.
 - This motion is continuous – back and forth endlessly.
 - The system displays a sweeping radar image on the OLED display (similar to the one from last assignment).
 - In the top right corner, display the current compass heading (calibrated to true North) of your moving platform. This will update as the platform sweeps.
 - When you detect an object, place a 2x2 pixel dot on the display at its discovered location.
 - First determine the maximum reliable sensing distance of your ultrasonic sensor.
 - Next determine what the radius of your onscreen arc is.
 - Use the ratio of these to determine the “depth” of your dot (ie. how far from bottom to top).
 - The dot remains on screen until it is no longer detected (by a subsequent pass).
 - Large objects should result in larger dots formed by multiple smaller 2x2 dots.
 - Tip: Use a buffer in memory to build the bitmap that you want to draw and then update the screen when necessary. And don’t forget your 10th grade geometry.
 - On your assignment wiki page explain how you built the system including the maximum sense distance, system events, and the scheduler (how many priorities, etc) (evaluate it’s effectiveness).
 - Commit the program to your team repository under the branch assignment4-fcfs.
 - Demonstrate with a short video.

- Rewrite the application from part 1 to use a priority scheduler.
 - On your assignment wiki page explain the differences between this version and the FCFS system (including the scheduler).
 - What was affected: efficiency, stability, maintainability, etc
 - Commit the program to your team repository under the branch assignment4-priority.
 - Demonstrate with a short video.