Lab 6: IR Distance Sensor and Data Acquisition

- **Objectives:**
  - Understand the configuration and operation of on board ADC14.
  - Understand the operation concepts of IR Distance Sensors.
  - Perform sensor calibration.
  - Gather samples and perform data acquisition.

- **Summary:**

In preparation for collision avoidance operation, and fine movement of the robot, distance sensors need to be calibrated so as to have a clean mapping between measured signal strength and distance to objects. Part A of this lab estbalishes such mapping for every sensor to efficiently control how close the robot should approach an object before stopping or changing course of movement. Part B of the lab will implement exploration with collision avoidance through careful choice of sensor smapling times and robot movement speed.

Part A: IR Distance Sensor Calibration

1. Create a CCS project to facilitate the calibration of the three Sharp distance sensors mounted on the robot.
2. Calibration is done as follows:
   a. For every sensor of the three (left, center, and right), the distance from wall to sensor must be known using a measuring tape, and the corresponding received signal strength must be recorded for at least 30 times.

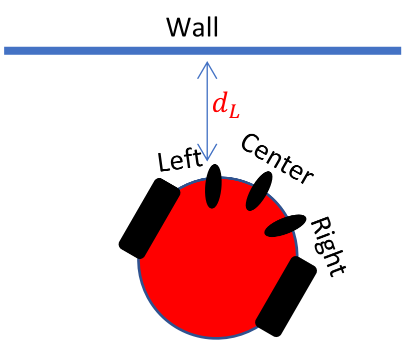


Figure 1 Calibrating Left Sensor

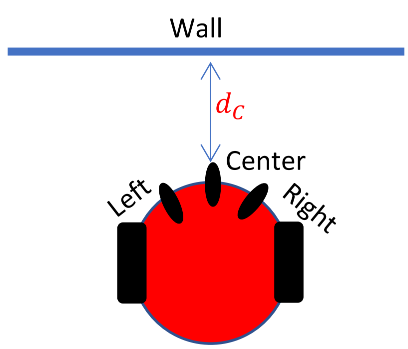


Figure 2 Calibrating Center Sensor

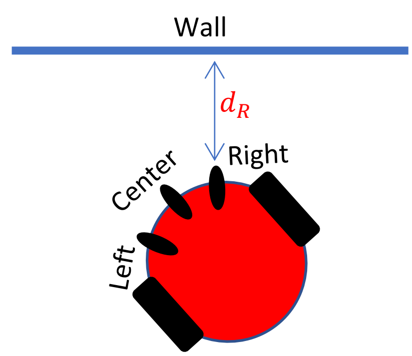
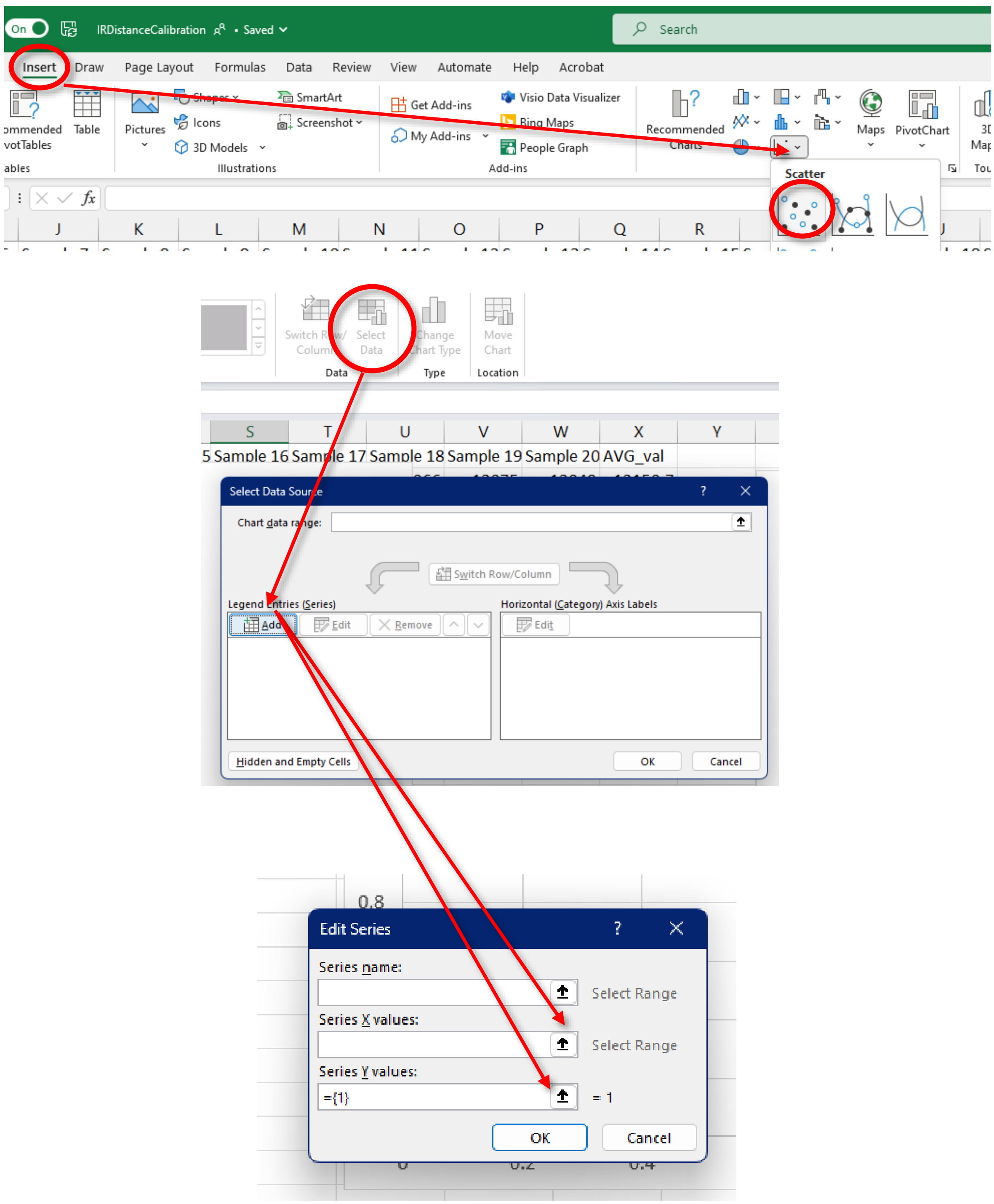


Figure 3 Calibrating Right Sensor

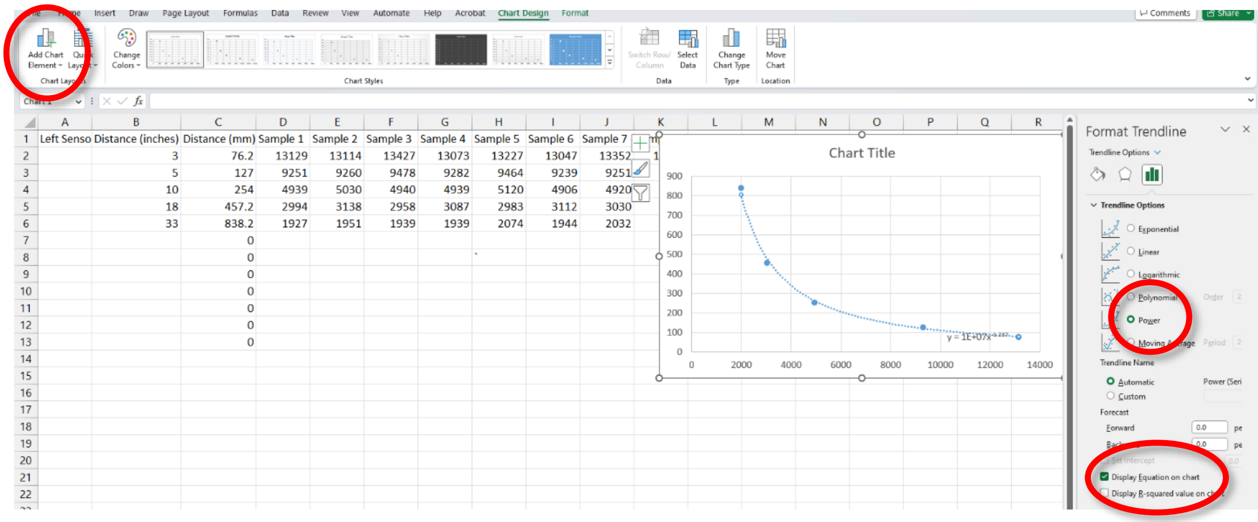
   b. Create an Excel sheet for every sensor to record the measurements and calculate their average. Here is a template for the left sensor:

Distance d_L(inches)	**Distance d_L(mm)**	**Sample 1**	**Sample 2**	**…**	**Sample 30**	**Avg value**
3	76.2	13129	13114	…	13073	13150.7
5	127	9251	9260	…	9282	9297.45
10	254	4939	5030	…	4939	4939.6
18	457.2	2994	3138	…	3087	3034.2
33	838.2	1927	1951	…	1939	1991.55

c. On Excel, draw a scatter plot (Insert -> Charts -> Scatter) showing the average ADC output for the current sensor on the x-axis and the corresponding distance on the y-axis (Click the inserted blank chart -> Select Data -> Add, then choose the range of cells for the x-axis and the range of cells for the y-axis). On the x-axis is the ADC average output value and on the y-axis is the distance in mm



d. Apply curve fitting to find the best function that accurately maps the ADC output into actual distance from object. To that end, select the chart, choose Add Chart Element -> Trendline -> More Trendline Options -> Power, and check Display Equation on Chart



3. Use the provided function `void ADC0_InitSWTriggerCh17_14_16(void)` (in file **ADC14.c**) to initialize ADC14 for sampling through pins P9.0 (left sensor), P6.1 (center sensor), and P9.1 (right sensor).
4. In an infinite loop, the ADC should be sampling the distance sensors and return readings for calibration. The provided function
   `void ADC_In17_14_16(uint32_t *ch17, uint32_t *ch14, uint32_t *ch16)`
   lets the ADC sample the three sensors and return results through the three input pointers ch17 (left sensor), ch14 (center sensor), and ch16 (right sensor).
5. In each iteration, a delay of about 10 ms should be added to allow for accurate sampling results.
6. In order to collect measurements, run the Debug mode and add a Breakpoint at the end of the iteration. This way, you can manually run one iteration at a time and record the corresponding sensor measurements through the Expressions window.
7. You should show the curve and function to your instructor.

<u>Part B: Area Exploration with Collision Avoidance</u>

1. Create a CCS Project to equip the robot with a collision avoidance system enabled by the IR Distance Sensors.
2. [Extra credit] Add a collision detection system to the robot.
3. The Robot should be normally moving forward with front lights ON at a speed of about 33% duty cycle, and the distance sensors sampled every 100 ms and the distance to nearby objects is measured to assess a possible collision.
4. If any sensor detects an object within 100 mm, the robot should assume it is on a collision course with that object and accordingly react as follows:
   a. Left and right sensors both are within 100 mm from object(s): turn right for about 1 second (i.e., turn around).

   b. Right sensor only is within 100 mm from object: turn left for a whole sample time (i.e., swerve left)
   c. Left sensor only is within 100 mm from object: turn right for a whole sample time (i.e., swerve right)
   d. Center sensor is within 100 mm from object: switch OFF front lights, switch ON back lights, back up for 0.5 sec then turn right for a whole sample time.
5. If all sensors are within at least 110 mm or more from objects or walls, robot moves forward with front lights ON for a whole sample time.
6. Here is the list of files needed for that project.

File Name	Description
Main_lab6b.c (write it)	• Has the `main` function that initializes the robot modules (e.g., Clock 48 MHz, lights, motors, ADC14, etc.) and operates the robot with collision avoidance enable and collision detection optional.
Motor.c **Motor.h** (reuse from Lab 3)	• Motor movement functions that rely on PWM
PWM.c **PWM.h** (reuse from Lab 3)	• PWM signal generation functions and duty cycle control
RobotLights.c **RobotLights.h** (reuse from Lab 2)	• Initialization of robot LEDs • Front and Rear robot lights control functions
BumpInt.c **BunpInt.h** **CortexM.c** **CortexM.h** (reuse from Lab 4)	• Enable collision detection through the bump sensors (optional for extra credit) • Support for MSP432 interrupt system
ADC14.c **ADC14.h** (provided)	• Initialize the ADC14 to sample the IR distance sensors. • Support for sampling IR sensors signals
Clock.c **Clock.h** (provided)	• Initializes the 48 MHz clock source • Provides delay functions

- **Check off and Deliverables:**
   1. For Part A, measurement curves and curve fitting equations should be shown to your instructor. For Part B, the robot should explore an area (e.g., a maze) without colliding with objects. Instead, it should make the necessary turns in response to the detected object position.

   2. Submit a lab report on Canvas detailing the architecture of the main file functions for each part of the lab. For Part A, add the measurement table and the associated curves and curve fitting functions.

   3. Include a video capture of your robot performing collision avoidance.

4. Submit a compressed folder for your lab projects on Canvas.