ECEN 345/ECEN 3450-001

Mobile Robotics I

Prof. A. Gilmore (Instructor)

Sajjad Alhassan (TA)

Laboratory Assignment 3

Lab 3 - Obstacle Avoidance with IR Sensors

Written by

David Perez

Department of Electrical and

Computer Engineering (ECE)

University of Nebraska

Lincoln/Omaha

Due: 02/25/2024

Introduction

The focus of this lab was on the object avoidance using infrared (IR) sensors. Using these sensors the CEENBoT robot was programmed to avoid objects based on detections from the IR sensors. These sensors were tested with 4 different surfaces (soft, dark, light, at an angle) to test the accuracy of the sensor's detection. The resources that were used to complete this lab were the lab 3 handout, the CEENBoT robot, and some arbitrary objects for testing. This took about 5 hours to complete and was completed solely by David Perez.

Background

The CEENBoT robot contains two microcontroller units (MCU's), the ATmega 324p and the ATtiny48. Additionally the CEENBoT has two IR sensors located at the front left and right of the robot. To illustrate how these sensors communicate to the main MCU the ATmega324p and illustration is shown in **Figure 1**.

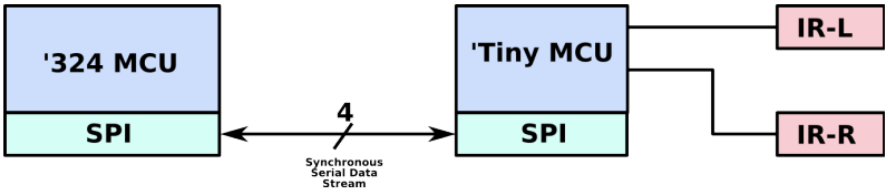

Figure 1: CEENBoT IR Sensor Architecture

The communication protocol that enables the communication between these MCU's is the synchronous serial peripheral interface (SPI). To communicate over SPI there are 4 required communication lines. The MOSI line stands for master-out serial-in, MISO which is master-in serial-out, an SS line which stands for slave select, and a CLK or clock line for synchronization.

When programming the CEENBoT there is a firmware function called *ATTINY_get_state()* that returns a boolean indicating if the IR sensor has detected an obstacle or not. This function takes in additional arguments that allow the user to get the state of the left, right, or both sensors. The sensor that is equipped on the CEENBoT is a *GP2Y0D810Z0F* sensor. This sensor has a maximum reading distance of 4 inches

which is crucial to know when designing object avoidance on this robot. Furthermore, when testing the programs on the robot there are two diodes on these sensor modules that illuminate green if the sensor has detected an object.

To achieve the task of object avoidance the main functionality is to be broken into two functions. These two functions are the *IR_AVOID()* and the *CRUISE()* functions. The *IR_AVOID()* function should be used to determine the robot's behavior when something is detected. The *CRUISE()* function on the other hand should drive the robot around when there are no objects detected. It is important to note that the *IR_AVOID()* function should take precedence over the *CRUISE()* function as to ensure the robot can respond to objects in an appropriate time frame.

Procedure

The task that this lab looked to explore was object avoidance using IR sensors. When it comes to implementing object avoidance on the CEENBoT robot the first step taken was to get the vehicle to stop when an object is detected. To test the accuracy and consistency of these sensors there were four different surfaces that were used. The first surface was a light surface, followed by testing a dark surface, then a soft surface, and lastly a surface at an angle. Using the results from these experiments (discussed in the *Results* section) the robot was then programmed on how to respond to these stimuli.

Given that there are two sensors there are 4 different scenarios that the robot can find itself in. The first scenario considered was when there were no objects detected by the sensors. In this scenario the robot was programmed to simply drive forward. When an object was detected by both sensors that robot would first backup 4 inches then turn in an arbitrary direction (the direction chosen was left). The turn that was implemented was a reverse backwards turn. This turn was achieved by spinning both motors in reverse where the motor on the side of the desired turn direction would rotate about 60 degrees while the other motor would reverse about 30 degrees. This behavior was necessary since the robot would not have sufficient room to turn in plane when an object had been detected. When only the left sensor was triggered the robot was instructed to complete this reverse and turning behavior to turn right. Therefore, when

the right sensor detected an object the robot would move backwards about 3 inches and turn left.

Source Code Discussion

The implementation architecture chosen for the object avoidance lab was a lower motor controls layer with an application layer performing those functionalities. In *app.c* the function *IR_AVOID()* which is the main functionality of the attached c-program zip file. Within this function there are three if conditions where it first checks if there is an object detected directly in front of the robot.. This is done by checking both IR sensors to detect an object by calling *ATTINY_get_IR_state(ATTINY_IR_BOTH).* The additional are a check if the left sensor detects something then checks the right sensor. If there is an object detected the robot stops, reverses 3 inches, and reverse turns via the 3 corresponding function calls: *STEPPER_stop(STEPPER_BOTH, STEPPER_BRK_ON), move_backwards_in_inches_stwt(3), reverse_backup_60_degrees(false)*.

```
void reverse_backup_60_degrees(bool turn_left)
{
        if (turn_left)
        {
                // Turn left (~60 degrees)
                STEPPER_move_stwt(STEPPER_BOTH,
                STEPPER_REV, 175, 200, 400, STEPPER_BRK_OFF,  // left
                STEPPER_REV, 30, 200, 400, STEPPER_BRK_OFF); // right

                } else {
                // Turn right (~60 degrees)
                STEPPER_move_stwt(STEPPER_BOTH,
                STEPPER_REV, 30, 200, 400, STEPPER_BRK_OFF,  // left
                STEPPER_REV, 175, 200, 400, STEPPER_BRK_OFF); // right
        }
}
```

Figure 2: Reverse Turn Robot Code Snippet

As seen in **Figure 2** the reverse turn is used to rotate the robot right or left. To achieve a left turn the robot will rotate the left motor in reverse by *175* steps and the right motor by *60* steps.

Moving the robot forward and reversed was code ported over from the previous lab but with a separation into blocking and non blocking functions. The blocking function was necessary to get the robot to not turn until it had reversed away from the object. The non blocking function *move_forward_in_inches_stnb()* was needed to allow priority of the object detections *IR_AVOID()* function over the *CRUISE()*.

Results

When testing the IR sensor against the 4 different surfaces the robot would perform in a different manner depending on the surface. The fist surface that was tested was a light colored surface. When driving the robot directly at this surface the robot would respond almost immediately when reaching 4 inches from the surface indicating that the sensors were able to successfully detect the object. When it came to testing dark colored surfaces, the robot had a similar response and was able to consistently detect the dark colored objects. To test the response of a soft surface (a sweater being the surface) the robot would occasionally not identify the surface with able time. To test the accuracy the robot was directed at this surface 10 times. Of the 10 tests performed, the robot would not identify the surface 1 out of 10 times. The final surface tested was a light colored object that was angled in relation to the front of the robot. Performing these tests, the robot would only detect the surface about 1 out of 10 teams. These tests exemplified a pitfall of the IR sensors where the sensor's emitted signal would reflect away from the sensor preventing it from detecting the object.

After testing the sensor's accuracy and reliability the robot was set out into open space. The robot would begin cruising until an object was placed in front of its path. During which either the left, right, or both sensors would detect an object and the robot would maneuver around the object.

Conclusion

The objective of this lab was introducing an IR sensor and how it could be used for object detection for mobile robotics. This sensor was tested against dark, light, soft, and angled distance validating that objects at 4 inches would be detected reliably on all surfaces except on angled surfaces. The angled surfaces produced a detection rate of 1/10 while the other averaged 9/10 or above. Built on top of this was object avoidance when objects were detected on the left, right, or both sensors. Using functionality the robot was able to successfully avoid objects in its path and cruise in open space.