ECEN 345/ECEN 3450-001

Mobile Robotics I

Prof. A. Gilmore (Instructor)
Carl Endrulat (TA)

Laboratory Assignment #3

Lab 3 - Obstacle Avoidance with IR Sensors [Report]

Written by :
Nelson Lefebvre 03137910

Department of Electrical and
Computer Engineering (ECE)
University of Nebraska
Lincoln/Omaha

Due: 09/17/2022

• *Overview:*

Ressources :

- CEENBoT-API – Getting Started guide
- CEENBoT-API: Programming Fundamentals
- Ceenbot + Microchip studio + Ceenbot Utility tools (LAB 1)

Time invest in this project:
I spent around 2h30 to make this LAB. I quickly found the different functions in the guides provided (in 1h30) and I established a strategy that I found rather coherent (in 1h). I had a slight difficulty finding a solution to the obstacle coming from the front.

Team assignment:
I did this project on my own but I received some advice from students in the class.

Goal of this lab:
The main lesson of this LAB is to understand how to implement an exteroceptive sensor, here we will use a rather common sensor in robotics: IR sensor. To allow CeenBot to guide itself to avoid obstacles. For this we will have to use two functions.

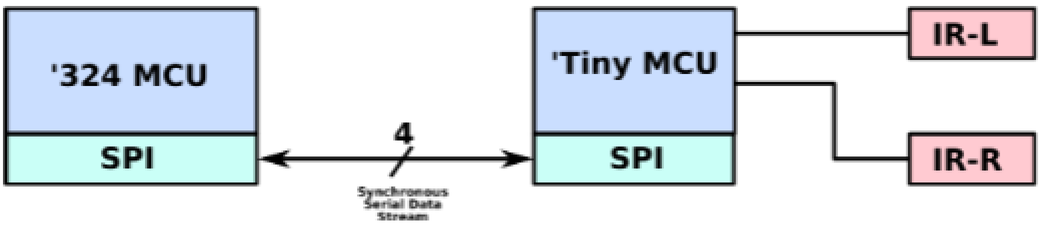
Cruise : To make the robot go forward as long as the ir sensor is not stimulated.

IR-AVOID: Function that will distinguish an obstacle with the reception of a signal previously sent, indicating an obstacle. Once the signal is received we will have to establish a strategy to make the robot move to avoid the obstacle and continue its way.

Difficulties:
Problem encountered during the LAB, the robot when it meets a corner of wall the sensor does not detect the wall. One of the two solutions that I think is possible is either to move the sensors a little closer to the center of the robot (positive point: more precision for angled walls, negative point: a smaller angle of vision). Or add two other sensors in the middle of the robot (IR or other type of exteroceptive sensor).

• *Background :*

The IR sensor that emits and detects infrared light reflected from any surface within the detection range of the unit.
Here for the background in this lab we used IR proximity sensor to detect the obstacle to avoid them. Here is the schema of the module :



The IR state information will be transferred via the SPI serial interface to the MCU which will then be decrypted by the ATmega324 library.

We will use the ATTINY_get_IR_state() function of the CEENBoT API, which returns a value of type BOOL, which will be TRUE if the IR sensor whose state you are requesting is "triggered" (by an obstacle) or FALSE (zero) otherwise. This function has 3 possible parameters: ATTINY_IR_BOTH (both IR sensors are triggered), ATTINY_IR_LEFT (only the left sensor is triggered), ATTINY_IR_RIGHT (only the right sensor is triggered). And so we need to establish a strategy to dodge obstacles with an IRAVOID() function and make a continuous movement that can be influenced with the IRAVOID() function with the CRUISE() function.

In the experimentation you can see that the green LED indicates that the IR sensor is stimulated.

Procedure and discussion about the source code:

During this lab we had 5 problem questions to solve:

- We needed to experiment with different ways to implement avoidance behavior (stop, back up or turn, move forward) using the IR sensors.
  But also determine how long it takes the robot to stop at different speeds in order to determine the maximum speed of CRUISE to prevent the robot from running into the objects it detects. Prevent the robot from running into the objects it detects.
  Determine which shock switch is activated (LEFT only, RIGHT only or BOTH) to determine how you will move the robot away from an obstacle.

  To determine how to stop and move forward we use a function set up for this purpose called STEPPER_move_stnb which allows the robot to continue running as long as nothing is detected (here as long as no function is activated). With our eyes we can notice that the robot makes very fast forward micro movements. This function is implemented in an infinite while loop which will be followed by the IR-AVOID function. The IR-AVOID function is executed here just after and contains the movement functions we used in the previous LAB like :
  STEPPER_move_stwt(STEPPER_BOTH,STEPPER_REV,
  400,200,400,STEPPER_BRK_OFF,STEPPER_REV,400,
  200,400,STEPPER_BRK_OFF); to move backwards etc..

  Which are activated with the one that looks at the state of the IR sensors (TRUE or FALSE) with the different parameters ATTINY_IR_BOTH, ATTINY_IR_RIGHT, ATTINY_IR_LEFT that are related to the respective detection of the two, right and left.

- Create a function of your chosen avoidance behavior and name it IR-AVOID. Be sure to show what happens when LEFT, RIGHT and BOTH IR sensors are triggered.

  In this function we have chosen to turn 90* in the direction where the IR sensor is activated. And when both sensors are active we go backwards and do a 90* to the right. This is one of the strategies that I found interesting because I had the idea to create a variable that changes according to the previous movement but it tended to return to the obstacle and therefore it was unproductive.

- Create a CRUISE behavior that will move the robot forward in some manner. Take advantage of the CEENBoT API functions.

  As said before we use the function STEPPER_move_stnb which advances as long as other functions are not activated.

- Write a C program that implements the IR-AVOID behavior and the CRUISE behavior. (When working together the two behaviors will cause the robot to move forward, STOP before hitting an obstacle; TURN in some manner away from the obstacle; and continue moving forward until another obstacle is detected.) The IR-AVOID behavior should be coded so that it has precedence in sending commands to the motors whenever an obstacle is sensed over the CRUISE behavior.

  I had the idea to realize a recursive function but this idea was abandoned following the discovery of movement with stnb.
  The goal is to create the two functions CRUISE and IR-AVOID, the function IR-AVOID will always be called in the while loop so that it is always executed and in this one we look if the sensors are activated or not activate Cruise which makes us advance. Once executed we return to the while loop and start the cycle again indefinitely.

- Experiment with the IR-AVOID behavior and observe sensor response with various obstacles in front of the robot (light, dark, at an angle, shiny, soft, rough). Observe and record both the range and effectiveness of IR reflectance sensing on at least 4 different types of surfaces.

  Here is the different experimentation with différent type of obstacle (5ft distance) :

White wall light on	5sec
White wall light off (or really low)	5.3sec
Black wall light on	5sec
Round obstacle light on	Undetected

We can observe that the parameter of the obstacle will make the detection more easily or not. The most easily detected obstacle will be a white lightly flat obstacle and the worst one will be one that is different from black without light because the IR signal will be reflected badly.

Results :
We had rather conclusive results, because the robot is extremely well reactive and detects the obstacle well before touching it. Our strategy to rotate 90* in the direction where the

sensor was activated is rather conclusive. One of the only problems is the detection of the wall angles which are not necessarily in the perimeter of detection of the IR sensor.

Conclusion :

We can therefore conclude that obstacle detection by exteroceptive sensors can be influenced by the type of environment. But also the strategy used influences the management of the program goal. We have chosen to use 90* movements, but this can remain very archaic and prevent the robot from achieving its goal. Let's imagine if the wall is inclined and blocks the goal then the robot risks indefinitely the way to its goal.

Source Code :

```
/*    Auth: Nelson Lefebvre
*    Course: ECEN3450-Mobile Robotics I
*    Lab: Lab#3
*/

#include "capi324v221.h"

      void CBOT_main( void )
      {
            STEPPER_open(); // Open stepper services.
            // Enter the infinite loop to make the robot move
infinitly...

            //Make the screen available to write wich IR SENSOR is
detected
            LCD_open();
            LCD_clear();
            //Infinite Loop to make the robot move continuously
            while(1){
                  //Cruise fonction who make the robot move forward
                  void CRUISE(){
                        // Move foreward if nothing detected
                        STEPPER_move_stnb( STEPPER_BOTH,
                        STEPPER_FWD, 100, 200, 400, STEPPER_BRK_OFF, //
Left
                        STEPPER_FWD, 100, 200, 400, STEPPER_BRK_OFF ); //
Right
                  }
                  //IR-AVOID fonction who try to detect the obstacle and
act by moving in another direction or call the function cruise
                  void IRAVOID(){
                        // Check if BOTH IR's are "triggered".
                        if ( ATTINY_get_IR_state( ATTINY_IR_BOTH ) )
                        {
                              // Back up!
                              STEPPER_move_stwt( STEPPER_BOTH,
                              STEPPER_REV, 400, 200, 400,
STEPPER_BRK_OFF,
                              STEPPER_REV, 400, 200, 400, STEPPER_BRK_OFF
);
                              LCD_printf("Both");
                        // ... if not, check if just the LEFT IR is
'tripped'...
                        else if ( ATTINY_get_IR_state( ATTINY_IR_LEFT ) )
                        {
                              // Turn Left!
                              STEPPER_move_stwt( STEPPER_BOTH,
                              STEPPER_REV, 150, 250, 400,
STEPPER_BRK_OFF,
                              STEPPER_FWD, 150, 250, 400, STEPPER_BRK_OFF
);
                              LCD_printf("Left");
                        } // end else-if()
                        // ... if not, then finally check if the RIGHT IR
is 'tripped'.
                        else if ( ATTINY_get_IR_state( ATTINY_IR_RIGHT )
)
                        {
                              // Turn Right!
                              STEPPER_move_stwt( STEPPER_BOTH,
                              STEPPER_FWD, 150, 250, 400,
STEPPER_BRK_OFF,
                              STEPPER_REV, 150, 250, 400, STEPPER_BRK_OFF
);
                              LCD_printf("Right");
                        } // end else-if()
                        else{
                              CRUISE();
                        }
            }
            IRAVOID();
            }
      } // end CBOT_main()
```