Words: 1407

9/1/23

CE243 – C Programming

Assignment 2

1603905

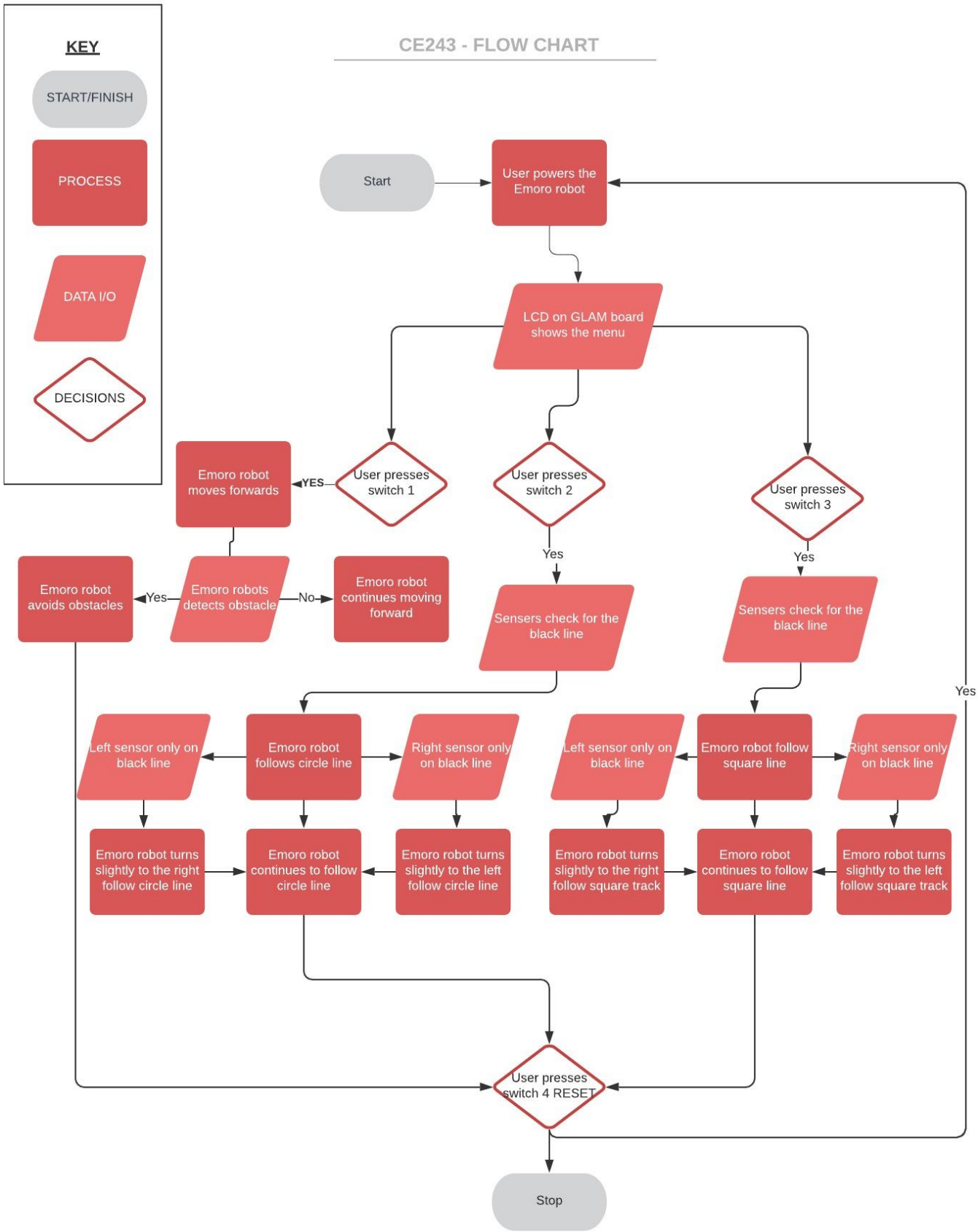
Contents

Introduction

The CE243 assignment 2 is an assignment where the programmer would need to create a C program to control the EMoRo robot to perform a given number of tasks which the user will push buttons on the GLAM board to input commands which there will be four functions as there are only 4 buttons. The first command will make the robot move in a forward motion as well as avoiding obstacles, whist the second command will get the robot following a line which this line will be in a circle. Additionally, the third command will be following a line but in this instance the robot will move along a square track, and the fourth command will stop the motion of the robot and reset the onboard computer for next button input from the user.

Furthermore, the programmer will need to code the LCD display to show “1.Obst”, “2.Circle”, “3.Square” and “4.Reset”, when the EMoRo robot is in use the LCD will change from the 4 options depending on the button the user presses. For example, if the user presses switch 1 then whilst the robot is moving the LCD will displayed “moving forward” and when obstacle is detected the LCD will display obstacle detected. If the user presses switch 2 then the LCD will display “Circle track following” whilst the robot is following the circle track. In addition to this, if the user presses switch 3 then the LCD will display “Square track following” whilst the robot is following the square track and if the robot makes a sharp turn the LCD will display “turning *** degrees” and when the robot moves along the straight line sections the LCD will display “Square track following”.

Flow Chart


KEY
START/FINISH
PROCESS
DATA I/O
DECISIONS
CE243 - FLOW CHART
Start
User powers the Emoro robot
LCD on GLAM board shows the menu
User presses switch 1
YES
Emoro robot moves forwards
User presses switch 2
User presses switch 3
Yes
Yes
Emoro robot avoids obstacles
Yes
Emoro robots detects obstacle
No
Emoro robot continues moving forward
Sensers check for the black line
Sensers check for the black line
Yes
Left sensor only on black line
Emoro robot follows circle line
Right sensor only on black line
Left sensor only on black line
Emoro robot follow square line
Right sensor only on black line
Emoro robot turns slightly to the right follow circle line
Emoro robot continues to follow circle line
Emoro robot turns slightly to the left follow circle line
Emoro robot turns slightly to the right follow square track
Emoro robot continues to follow square line
Emoro robot turns slightly to the left follow square track
User presses switch 4 RESET
Stop

Flow Chart explanation

The above page features the flow chart created. The flow chart includes a key which start/finish is featured with a different shape, whilst the next shape is an square and this represents the processes of the EMoRo robot and the next shape is an indicator for data input and output as well as the diamond shape representing decisions made by the user which the robot will implement.

The first step will be the user powering up the EMoRo robot which this represented as a process and then the robot will show the user menu through it GLAM board on the LCD with options 1.Obst, 2.Circle, 3.Square and 4.Reset. The next step is where the user gets to choose which button they want to press, and this can be in any order. For example, if the user presses the first button then the EMoRo robot will move forward and avoid any obstacles that are less than 40cm away. If there are no obstacles, then the robot will keep moving forward till there is an obstacle in the way. The only way to stop the EMoRo robot is for the user to press the switch four button which will reset the robot and it will take the user back to the menu as shown in the flow chart on page 3.

Additionally, if the user puts the EMoRo robot on the black line that is a circle then presses the switch two button then the EMoRo bot will follow the track as long as both sensors are on the black line, for example if the left sensor is on the black line and the right sensor is off it the black line then EMoRo robot will turn to the right to get back on track. Also, if the right sensor is on the black line but the left is not on the black line then the EMoRo robot will turn slightly to the left to get back on track. Once this is done and both sensors are on the black line then the EMoRo robot will continue to follow the black line for the circle track. To stop the robot following the black line the user just has to press the switch four button to reset the robot back to the menu.

Furthermore, if the user puts the EMoRo robot on the square track black line and then presses the switch three button the robot will follow the square black track and display this on the LCD as well as the angles in degrees. Also, if the right sensor is on the black line but the left is not on the black line then the EMoRo robot will turn slightly to the left to get back on track. Once this is done and both sensors are on the black line then the EMoRo robot will continue to follow the black line for the circle track. To stop the robot following the black line the user just has to press the switch four button to reset the robot back to the menu.

Test results

The results below are the results from the circle track following and the square track following which the tests will be carried out 5 times for both tests.

Circle track results

Five tests were concluded which the results are displayed below.

Test	Time spent (seconds)
1	13.78
2	14.22
3	13.79
4	14.07
5	13.77

The first test concluded that the time spent for the EMoRo robot that it was fast to a certain extent and it did get through the testing completely with the time being 13.78 seconds. The second test was slower due to slower time recording whilst the third test time was similar to the of the first test time. Additionally, the fourth test was slightly slower compared to the other test times it was 0.15 seconds faster than the first test. The fifth test had the fastest time overall, although it was only 0.01 seconds faster than the first test. Various factors could be cause of the times being different such as battery power loss, time keeping issues such as measuring the time wrong/slow. The robot followed the line but made sudden stops for roughly 0.2 seconds then continued moving.

Square track results

5 Tests commenced

Test	Time spent (seconds)
1	38
2	36
3	30
4	34
5	39.42

The tests that commenced were inconclusive as during the test the EMoRo robot may spin on the angle then continue or continue turning then going in the opposite direction then complete the square. Many factors could be the reason for such as not getting the servos correct as well as faulty sensors. Test 3 is different because the EMoRo robot was able to complete the square track with minor issues while in the test 5 the robot get going onto the white line which it kept turning till the black line was found.

Arduino program code

```
/******1603905*******/
int servoPort = SERVO_0;
int servoPort1 = SERVO_1;
int S_Pin;
int pressed=0; //for switches pressed
int lcdpause=0; //to stop the flickrering
int follow=0;
static int Lfwd = 1000;                                          // Servo
values for each motor and each direction
static int Rfwd = 2000;                                          // Servo
values for each motor and each direction
static int Lbck = 2000;                                          // Servo
values for each motor and each direction
static int Rbck = 1000;                                          // Servo
values for each motor and each direction
static int Lstp = 1500;                                          // Servo
values for each motor and each direction
static int Rstp = 1500;                                          // Servo
values for each motor and each direction

void setup() {
  InitEmoro();
  EmoroServo.attach(servoPort);               // initialize servo motor on
selected port
  EmoroServo.attach(servoPort1);
  Serial.begin(9600);                         // initialize serial
communication:
  Serial.print("Pulse width: ");              // print string
  Serial.println(" us\n");                    // print string
  Ultrasonic.attach(S_Pin);
  int follow = 0;
  Gyr.init();
  pinMode(IO_0, INPUT_PULLUP);                                  // Initiates
input with pullup (infrared sensor on IO_1 port)
  pinMode(IO_1, INPUT_PULLUP);                                  // Initiates
input with pullup (infrared sensor on IO_2 port)
  // if LCD available print info:
  if (ReadEmoroHardware() & LCD_AVAILABLE) {                      // if LCD
ia available
    Lcd.locate(0, 0);                                             // set LCD
cursor to the first row, the first character
    Lcd.print("1.Obst 2.Circle");                                    // print
menu options
    Lcd.locate(1, 0);                                             // set LCD
cursor to the second row, the first character
    Lcd.print("3.Square 4.Reset");                                 // print
constant string
    lcdpause=1;

  }
}

void loop() {

    if (ReadSwitch(SW_1) == 1 || (pressed == 1)){ //Indicates that uf
switch 1 is pressed the emoro robot will report switch 1 is pressed
      Serial.println("1.Obst");
```

```
      if (ReadEmoroHardware() & LCD_AVAILABLE) {
        Serial.println("1.Obst");
        Lcd.clear();
        Lcd.locate(0, 1);                                    // set LCD
cursor to the second row, the first character
        Lcd.print("Moving Forward");
        lcdpause=2;
        // Set servo pulse to 1500us. For position servo motors this will
set the middle position,
        // for servo motors with continuous rotation this will stop
rotation
      }
      pressed = 1;
      obstacle();
    }
    if (ReadSwitch(SW_2) == 1 || (pressed == 2)) { //Indicates that if
switch 2 is pressed the emoro robot will report switch 2 is pressed
      Serial.println("2.Circle");
      if (lcdpause!=3) {                // stops the flickring
        Lcd.clear();
        Lcd.locate(0,1);                                     // prints
circle track following
        Lcd.print("Circle track");
        Lcd.locate(1,1);
        Lcd.print("following");
        lcdpause=3;
        }
      pressed = 2;
      circle();
    }



  else if (ReadSwitch(SW_3) == 1 || (pressed == 3)) {
// //Indicates that if switch 3 is pressed the emoro robot will report
switch 3 is pressed
    Serial.println("3.Square");               // print constant string
    if (lcdpause!=4) {
      Lcd.clear();
      Lcd.locate(0,1);
      Lcd.print("Square track");                          // // prints
square track following
      Lcd.locate(1,1);
      Lcd.print("following");
      lcdpause=4;
    }
    pressed = 3;
    square();
  }
  if (ReadSwitch(SW_4) == 1 || (pressed == 4)) {
// if Switch SW_4 is activated then it indicates to the emoro robot will
report switch 4 is pressed
    Serial.println("4.Reset");
    if (lcdpause!=5) {                //To stop the flickering
      Lcd.clear();
      Lcd.locate(0,0);
      Lcd.print("1.Obst 2.Circle");
      Lcd.locate(1, 0);                                          // prints
menu options
      Lcd.print("3.Square 4.Reset");
```

```
      lcdpause=5;
      EmoroServo.write(servoPort, 1500);
      EmoroServo.write(servoPort1, 1500);
    }
    pressed = 4;
  }
}
void obstacle(){
  int ultrasonic;
  ultrasonic = Ultrasonic.read(S_Pin);
  EmoroServo.write(servoPort, 2000);
  EmoroServo.write(servoPort1, 1000);

  if (ultrasonic < 40) { //sensors detect obstacles that are 40cm away
    EmoroServo.write(servoPort, 0);
    EmoroServo.write(servoPort1, 2000);
    Lcd.clear();
    Lcd.locate(0,4); //location of where text is displayed
    Lcd.print("Obstacle");
    Lcd.locate(1,4);
    Lcd.print("detected");
  }
  delay(200);
}

void circle(){
  if (digitalRead(IO_0) == 0 && digitalRead(IO_1) == 0) {       // Both
sensros are on the black, go forward.
      EmoroServo.write(servoPort1, 2450);                          //
left servo forward
      EmoroServo.write(servoPort, 1100);                          //
right servo forward
    }
    else if (digitalRead(IO_0) == 1 && digitalRead(IO_1) == 1) {  // Both
sensors are on the white, stop.
      EmoroServo.write(servoPort, Lstp);                          // left
servo stop
      EmoroServo.write(servoPort1, Rstp);                          //
right servo stop
    }
    else if (digitalRead(IO_0) == 0 && digitalRead(IO_1) == 1) {
// Only left sensor is on the black, turn right.
      EmoroServo.write(servoPort1, 1250);                          //
left servo forward
      EmoroServo.write(servoPort, 1250);                          //
right servo backwards

    }
    else if (digitalRead(IO_1) == 0 && digitalRead(IO_0)==1) {
// Only right sensor is on the black, turn left.
      EmoroServo.write(servoPort1, 1750);                          //
left servo backwards
      EmoroServo.write(servoPort, 1750);                          //
right servo forwawrd

    }

}

void square(){
```

```
    if (digitalRead(IO_0) == 0 && digitalRead(IO_1) == 0) { //Checks if
both the left and right sensors are on the black line

       EmoroServo.write(servoPort, 1050);
       EmoroServo.write(servoPort1, 1950); // moves backwords

       }
  else if (digitalRead(IO_0) == 1 && digitalRead(IO_1) == 1) {  //Checks if
both the sensors are detecting white
        Gyr.resetDegrees(); // resets the X.Y AND Z angles
        EmoroServo.write(servoPort, 1250);       //spins both wheels
clockwise and anti clockwise to turn at sharp turns
        EmoroServo.write(servoPort1, 1250);
        Lcd.clear();
        while (Gyr.readDegreesZ()<90){  //Set up the gyroscope reading for
turning 90 degrees
             Lcd.locate(0,0);
             Lcd.print("Sharp Turning");
             Lcd.locate(1,0);
             Lcd.print(Gyr.readDegreesZ()); // gives off the gyroscope Z
reading
           }
         Lcd.clear();
         Lcd.locate(0,0);
         Lcd.print("Square Track");
         Lcd.locate(1,0);
         Lcd.print("Following");  //Printed when the robot is going
straight

         EmoroServo.write(servoPort, 1500);
         EmoroServo.write(servoPort1, 1500);  //Stops motion on both wheels
         delay(100);


     }
   else if (digitalRead(IO_0) ==0 && digitalRead(IO_1) == 1) {   //Checks
if left the sensor is off black line
        EmoroServo.write(servoPort, 1250);
        EmoroServo.write(servoPort1, 1250);
        delay(1); //delays for 1 second
        }
   else if (digitalRead(IO_1) == 0 && digitalRead(IO_0) == 1) { //Checks if
the right sensor is off black line
        EmoroServo.write(servoPort, 1750);
        EmoroServo.write(servoPort1, 1750);
        delay(1); //delays for 1 second
        }
}
```

Conclusion

To conclude the project went well as all the specifications were met mostly as the obstacle, circle tracking and reset as well as the LCD display were a success, but the square tracking could have been done better as the turns did not always work the first time. A few parts of the Arduino code was researched online as well as through lectures which this was helpful as I got to use the gyro scope [1] and it gave me a minor scope on how the line follower [1] worked. For example

```
“    if (digitalRead(IO_0) == 0 && digitalRead(IO_1) == 0) { //Checks
if both the left and right sensors are on the black line

       EmoroServo.write(servoPort, 1050);
       EmoroServo.write(servoPort1, 1950); //moves backwords“ [1]
```

The code above represents some of code used from the lecture which this for both sensors and it checks if both the left and right sensors are on the black line. The EMoRo servos

In addition to this, the EMoRo library was helpful with understanding certain code as it made the code more effective such as using another loop for my obstacle, circle and square parts of the code [2]. I have cited the works used that help me with programming as well as using the EMoRo library examples to help further my understanding with the functions used.

Works Cited

[1] D. Zhai, “Unit 8: Week 9 Structures & Unions in C & Digital input/output,” 11 November 2019. [Online]. Available: https://moodle.essex.ac.uk/pluginfile.php/812541/mod_resource/content/2/CE243-T8-Structures-Unions-in-C-Digital-input-output.pdf. [Accessed 1 12 2019].

[2] J. Stivic, “List of functions of EMoRo 2560 library (Arduino) Function prototype Function description Page,” EMoRo, 2007.