SC2107 Lab2 Assignment Sheet (to be submitted to NTULearn before next lab)

Name: \_\_\_Ganesh Rudra Prasadh\_\_\_ Lab Group: SCE1 Date: 05/10/2023 \_\_\_

1. Section 6.2. Give a short 2-3 lines description on concept behind the reflectance reading process. Why does the black surface result in slower voltage decay?

Answer: An IR sensor consists of two things – An LED that emits IR (Infrared) light, and a phototransistor that receives and transforms the reflected light into electrical energy. The more light that the phototransistor receives, the more current flows, therefore the reflectance affects the effective resistance (as seen in the voltage decay of the pins on port 7).

In general, dark surfaces have the property to absorb any rays (such as infrared) better than lighter surfaces. Therefore, the amount of IR rays that is actually reflected onto the phototransistor is very low. This leads to less current being conducted by the transistor. As more and more current is conducted overtime, the effective resistance of the material decreases significantly. This reduces the RC value (since R is going down) which governs the speed of discharge – more commonly known as the ‘Time Constant’. Since the time constant reduces, the voltage decays at a slower rate for black surfaces since they are darker than any other material.

1. Section 6.2. Which parameter do you need to tweak in the Reflectance\_Read() if the reflectance sensor reading is not accurate? Hint: check the 8 steps for Reflectance reading.

Answer: If the reflectance sensor reading is not accurate, the time parameter in the Reflectance\_Read() function must be tweaked. This parameter controls the capacitor discharge duration, which is a key factor in the sensitivity of the sensor. Adjusting the time parameter helps in fine tuning the response of the sensor for all the possible surface reflectance.

If the sensor values are higher than the actual value, we can decrease the time parameter – which reduces the time for capacitor discharge, thereby reducing sensitivity which is useful for extremely reflective surfaces. Similarly, if the sensor values are lower than the actual value, we can increase the time parameter, increasing the time for capacitor discharge and thereby increasing sensitivity which is useful for less reflective surfaces.

1. Section 6.2. Write down the procedure to initialise P7.3 to be an input pin without internal pull-up resistor

Answer:

P7->SEL0 &= ~(0x08); // Set SEL0 pin to 0

P7->SEL1 &= ~(0x08); // Set SEL1 pin to 0 as well (GPIO configuration)

P7->DIR &= ~(0x08); // Set P7.3 as an input pin

P7->REN &= ~(0x08); // Disable internal pull-up (and pull-down) resistor for P7.3

1. Section 6.3. Where are the sources of the offset error between actual distance and the estimated distance return by the function Reflectance\_Position()?

Answer: Since there are not many sensors on the MSP432, it has to resort to weighing each sensor accordingly to determine the position of the black line. Due to that there might be a slight inaccuracy, since working with discrete values on a sensor is bound to only be a close prediction to the values.

Other ways could also be if there were other infrared rays in the area, it might affect the accuracy of the readings. Also, if the sensors were not calibrated accurately, there could be some fault. Finally, there might be some angle inconsistencies, in case the sensors are slightly tilted, and/or they don’t read the same angle due to a slightly different angle of incidence, which could all be possible due to slight defects.

1. Section 7.2.  Figure 7. The robot state toggled between LEFT and CENTER state repeatedly when it is detected that the robot is off to the left of the line (input: ‘01’). If the outputs of the FSM states are connect to the input of the DC motor, how would the input signals to the DC motor looks like? Which wheel will move at a slower speed?

Shape

Description automatically generated with medium confidence

Answer: In order to move to the right, the left wheel would have to move faster than the right wheel to simulate a turning effect. At left state, the FSM outputs a control signal that would drive the left motor to turn faster than the right motor (simply increase the left wheel speed and/or reduce right wheel speed). This is done to aid the robot in returning back to the center of the line. While in the center state, the FSM outputs a control signal to both motors to move forward, with both wheels moving at the same speed.

1. Section 7.3. Fix the bug in the 11-state FSM design.  
   A picture containing table

   Description automatically generated

State\_t fsm[11] = {

{0x03, 500, { Right1, Left1, Right1, Center }},

{0x02, 500, { Left\_off1, Left2, Right1, Center }},

{0x03, 500, { Left\_off1, Left1, Right1, Center }},

{0x02, 5000, { Left\_off2, Left\_off2, Left\_off2, Left\_off2 }}, // 1000 -> 5000

{0x03, 5000, { Left\_stop, Left1, Right1, Center }}, // 1000 -> 5000

{0x00, 0, { Left\_stop, Left\_stop, Left\_stop, Left\_stop }}, // 500 -> 0

{0x01, 500, { Right\_off1, Left1, Right2, Center }},

{0x03, 500, { Right\_off1, Left1, Right1, Center }}, // 1000 -> 500

{0x01, 5000, { Right\_off2, Right\_off2, Right\_off2, Right\_off2 }}, // 1000 -> 5000

{0x03, 5000, { Right\_stop, Left1, Right1, Center }}, // 500 -> 5000

{0x00, 0, { Right\_stop, Right\_stop, Right\_stop, Right\_stop }} // 500 -> 0

};

1. Section 7.3. What is the purpose of toggling LED within the main routine or ISR?

Answer: Toggling the LED is done as a form of debugging since it works as a confirmation that the Output pins are configured properly. This toggling gives the user a clear portrayal of that microcontroller being active. A lack of this toggling LED will clearly suggest that there is some software issue, which will cause the program to not run. The output LEDs will either be all 1s or all 0s, and the user can then debug it. Similarly, if the LEDs blink repeatedly, it means that there isn’t an issue here and the program will run smoothly (unless obviously there is some error on runtime).

1. Section 7.4. What hardware and software modifications are required in order for the robot to move within a lane, i.e. between two black lines, instead of following a line? Detail algorithm not required. Just one bullet point each for hardware and software.

Answer:

Hardware: Adding reflectance sensors on the left and right side of the robot. By placing them on the outer boundary of the robot on either side, the robot can then sense if either of the sensors sense a black line, and move in the opposite direction to stay in between the lines.

Software: Using the two additional reflectance sensors, we can modify the algorithm to move such that it considers the relative positions of the lines based on the new sensors. Such that, if both sensors sense a line, it’s in the center and therefore moves forward. If only one of the sensors sense a line, it will adjust its rotation and stay in the lane without moving towards the line.

These two modifications is the basis of an algorithm that will help enable the robot to move in between two lines instead of just simply following a line.