UNIVERSITY OF REGINA

ENEL 387 Project
Smart Robot Car Final Report

Mar. 30, 2019

Li Pan

200371281

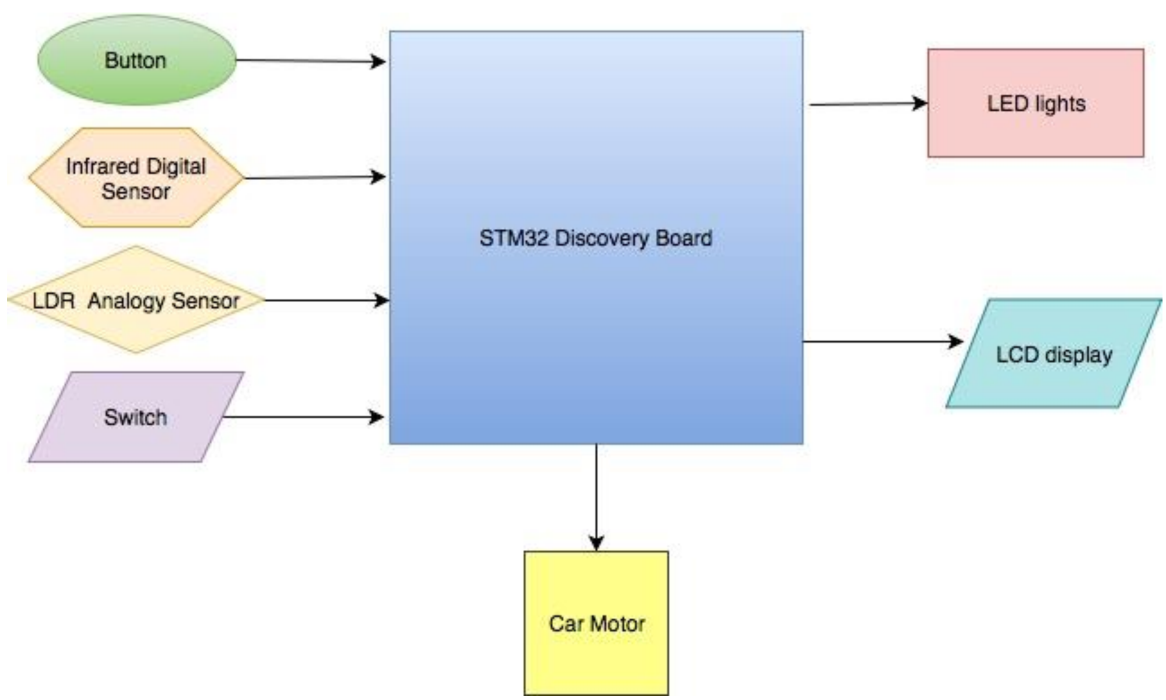
Instructor: Dave Duguid

Introduction

This project implements a smart car to deal with different driving situations, using the device of STM32F100RB as a micro-controller. The smart car has four wheels, with each being controlled by an individual motor. Once the power button is turned on, the car will drive automatically and handle different driving situations. Two infrared digital sensors are set on at front of the car to detect upcoming obstacles- one on the left, and the other on the right. If the left infrared sensor detects an object, it will send a signal back to the processor. By altering the speed of the wheels, the car will turn to the right. Similarly, the car can turn left when the right side sensor detects an object in the detection range. Also, the car will stop whenever both sides sensors have detected an obstacle. The LDR as an analogy sensor can control the four LED lights according to the brightness of the environment.

Hardware Description

Block Diagram:



System Inputs:

- Power control Button
  Power to the device is controlled by a switch on the 9 V battery case.

- Digital Infrared Sensor
  The distance sensor will detect an upcoming obstacle on the driving path from a certain distance. Depending on which side of the infrared sensor has an output signal, the processor will decide the speed of motors. This helps to reduces the chances of the car having an accident. It also protects itself from having physical damage occur.

- Analogy Light Sensor
  Light Dependent Resistor (LDR) changes the resistance in a circuit according to the brightness of the environment. Using Port A3 to read a value, the LED lights will turn on according to different values.

System Outputs:

- 4 Motors
  The two left side motors are controlled by Port A8 PWM, and two right motors are controlled by Port A9 PWM. The speed of the wheels can be changed by using different PWM to control the voltage going to the motors.

- LCD Display
  The LCD screen will display the driving condition. It will display a reminder message whenever it begins turning. The screen will also display the driving condition.

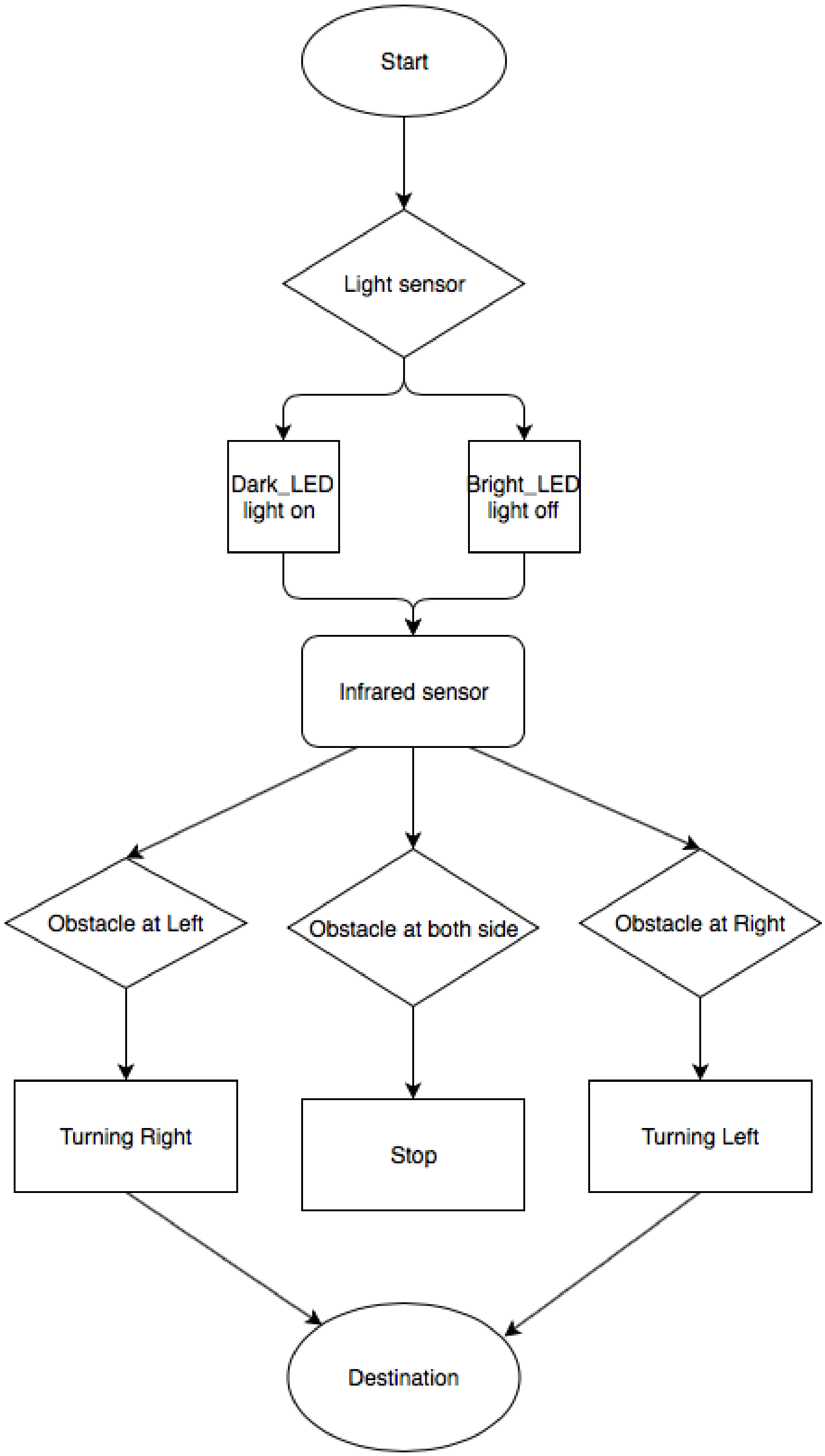
- LED lighting lights
  The LED lights up as another output device is added to the board, it will turn according to the brightness of the environment. There is two levels of darkness set for LED lights. When the environment is a bit dark the green LED lights turn on. When the environment is very dark, the green LED lights and yellow LED lights will all turn on.

- Turning Signal LED lights
  The two LED lights on the two rear sides of the car indicate the direction the car is turning.

Software Description

- The power button can control the device on or off.
- The self-driving car can drive at the same speed when there is no object in its path.
- The left side infrared sensor can detect if an object is near, and then react by having the left motors speed up and the right motors slow down - making the car turn itself towards the right.
- The right side infrared sensor function just like the left, only reversed.
- When both infrared sensors detect an object, the both motors will turn off, and the car will stop.
- The analogy sensor LDR will change resistance with the light. Port A3 will read the analogy value, and it will turn on the led lights according to the level of darkness.

System State Diagram



Testing Documentation

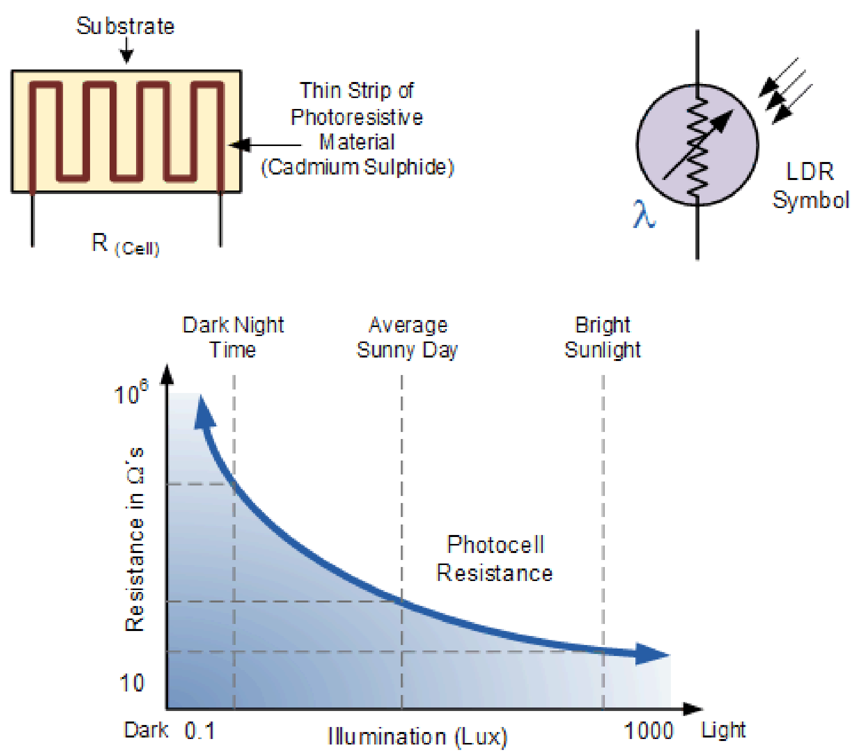
[Mar.5, 2019]

Finished purchasing all the necessary components to build a robot car. Lab instructor checked all the components. Made a list of function specification. Finished hand drawing the circuit for the motors.

[Mar.7, 2019]

The project requires an analogy sensor. Refer to the reference manual and STM32F100RB Platform table, only a few pin can be used as a ADC input pin. We setup PA0 and PA1 as ADC in a past lab, however, the PA0 connects to a resistor, and PA1 connects to the temperature analogy sensor on the board. I ended up using the similar way that I used to enable the Port A3 as ADC pin. I then built a circuit with 3.3 voltage power supply, a LDR, and a 1.1K resistor. Since we already used Port A3 as ADC, I connected the Port A3 with the Light Dependent Resistor(LDR) in the circuit.

According to the description of the “Light sensor”, the Light Dependent Resistor resistance decreases when of brightness increases.



[The light dependent resistor cell]

I enabled the register for the LCD screen, which can display a value according to the brightness. After recording the value range I then set up the value to send a warning signal from LED lights when the light is block around LDR. I finished the analogy sensor program.

[Mar.14, 2019]

I was planning to use ultrasound digital sensors. I noticed the ultrasound digital sensors has four pins. This was an issue, as I had bought an infrared collision avoidance sensor that only has three pins. One pin is for Vcc, one pin is for connecting the GND, and one is for sending back a signal. When it connects to 3.3V and the ground, the power LED light will turn on. I enabled the Port A4 and A5 as input. After connecting the output pin with them, the obstacle LED light will turn on when I put my finger close to the infrared sensor.

I set the detection distance through distance adjusted on the sensors. According to data sheet of the infrared sensor, the distance range is 2~ 30 cm. However, it was difficult to make the sensor work efficiently at such a large range. I found that the best working range is 7~10 cm.

[Mar.15, 2019]

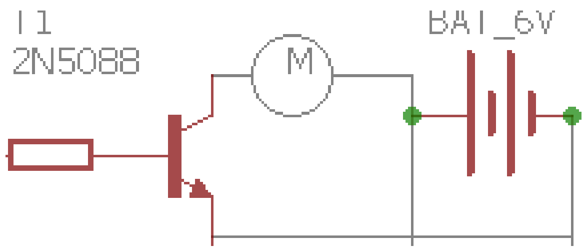
I was planning to use just one PWM to control four motors. The robot car will stop whenever the sensors detect an object. However, this plan soon seemed too simple, and would not make my robot car very remarkable. I talked with my lab instructor, and he suggested that I make the robot car able to turn around. The turning can be controlled by the different speed of two side motors. The voltage of the motor can control the speed of motors. Pulse Width Modulation can set the voltage by setting the width period of timer. According to the STM32 Pins Platform table, Timer1 is only available on PA8~PA11. In the preview lab I enabled the channel one on timer one. Enable two channels of Timer#1 can be used to check the time diagram with Oscilloscope.

[Mar.17, 2019]

I set the circuit for controlling the motor with two channels of Timer#1. I then connected two channels with 5088 transistors and 1.1K resistors. The emit pin of transistor connects to the ground. The basic pin connects with a resistor. The collect pin connects with one polar of motor. Also, another polar of motor is connected to the external power supplier. The emit pin of transistor also has to connect to the GND on the board. This circuit can guarantee the current through the motor can over the limit current of board. However, the speed of motors is still too slow to properly function. I used the DMM to check many times on the motors. But I cannot find a working solution to this issue.

[Mar.19, 2019]

After analyses from the lab instructor, I found the errors in my circuit. Without analyzing the circuit, I randomly select components. This is a big error I made. The speed of the motor is too small because the current going through the motor is too small. The data sheet is helpful when analyzing data in a circuit. However, I did not have the datasheet when I purchased the motors. My lab instructor taught me how to check the maximum current of motor without a data sheet. Since the voltage on the motor is 3~6 voltage. I assigned 6 voltage to a motor without turning the wheel, then we will be able to detect the maximum current can go through the motor. The current is 100mA without load, and the current is close to 1A with stall.



After analyzing the whole circuit, I was clear about choosing circuit components. First of all, the transistor I used is a 5088 transistor, which mean that amplifier β is too small. The collection pin only allows 50 mA current through. I need to use a big amplifier β transistor. My lab instructor then suggested that I could use TIP22 which β is 200. Secondly, I used a small size of resistor. The current in base pin should 5mA, I need to choose the size of 330 Ω resistor instead of 680 Ω.

[Mar.22, 2019]

Although I set up ADC pins the four LED lights on the board will not be able to be used as warning signals. Because one of my PWM pin is also controls one LED light. I have to find another way to send the information when an object in front of car. I chose the LCD screen. I added more code inside the condition loop. Whenever the left side sensor detects an object, the message “Turning right” will display on LCD screen. Similarly, whenever the right sensor detects an object, the LCD screen will display “Turning left”.

The 3.3 V pin on the board used for a few branch circuits. According to the datasheet of infrared sensor, the applied voltage range is 3.3V~6V. So I chose the 5V pin to connect the infrared sensor instead of use 3.3V pin.

[Mar.23, 2019]

I finished building the car and all circuitry. I used two small size breadboards to connect all devices. I tried to let the car move on the floor, but the car still moves very slow. This was because of the added weight from the new devices. However, if the car runs at a high speed, it could be damaged from an accident. This is because the sensor only detects objects within a range of 10cm. The car would not able to stop immediately if it was running at a high speed. Considering these two main reasons I adjusted the period width for the timer.

[Mar.28, 2019]

Enabled another two output pins PB12 and PB13. Added signal LED lights to the car. Set the LED lights to react whenever the car makes a turn. If the car needs to turn left, the left signal light will turn on, if the car needs to turn right, the right signal light will turn on. It's not hard to enable another two GPIO output pins, but it causes more wires to be brought to the circuit. The ground pin on the board is not enough. I made all LED lights share the same ground spot.

[APR.3, 2019]

Soldered the circuit. I was plan not to use breadboard, however, it's hard to solder all components with wires without burning the board. I have to keep the breadboard in the car. Reorganized the wires for the circuit. Made sure all the joins connected properly. Made the LED lights set at the right spot for the car. Also used proper length of wires in the circuit.

Summary

Function Checklist

FEATURE	COMPLETED
Mobile car drives straight without hiting objects	√
Mobile car turns left when an object at right side	√
Mobile car turns right when an object at left side	√
Mobile car stops when an object at front	√
LCD screen displays the driving condition	√
Green LED lights turn on when the environment is dark	√
All front LED lights turn on when the environment is totally dark	√
Signal LED lights show the turning direction	√

Recommendation

- The speed of the car is too slow. By reducing the burden of weight, the car could speed up. Alternatively, the motors could be replaced with ones of a higher voltage. However, the problem cannot be solved without considering other factors. The car requires a certain window of time to react to upcoming objects. If the speed is too high an accident might happen.
- The detection function is not good enough. Firstly, the range of detection is too small. This is can be solved by choosing a wide range of digital sensors. Secondly, the sensor without much trouble, detects stationary objects - but not moving objects. More factors need be considered.
- It is better if the car can adjust its direction after it has finished turning. The program should make another timer to count the turning time.

References

The light dependent resistor cell, 2019, *Light Sensors* , https://www.electronics-tutorials.ws/io/io_4.html