CECS 346 Spring 2024

Project 3 - A Smart House with a Stepper Motor Car

By Nicolas de Greef, Tylor Franca, Anthony Keroles

May 10, 2024

This project is a smart house and car that use sensors to interact with each other.

Introduction

There are two comments of this project, the smart house and smart car. There car has two operation modes, entering and leaving. In the coming mode, the smart car uses an infrared sensor to detect when an object is in front of it which is used in combination with the garage to not run into it. It runs forward then turn to the house. The house also has an infrared sensor meant to detect when the car is waiting for the garage to open which when detected will use the stepper motor to open the door allowing the car to enter. The car sensing the door is no longer in the way will go into the garage. When the car enters the leaving mode it will exit, turn, and leave the garage.

Operation

VIDEO DEMO-

20240517_124935.mp4

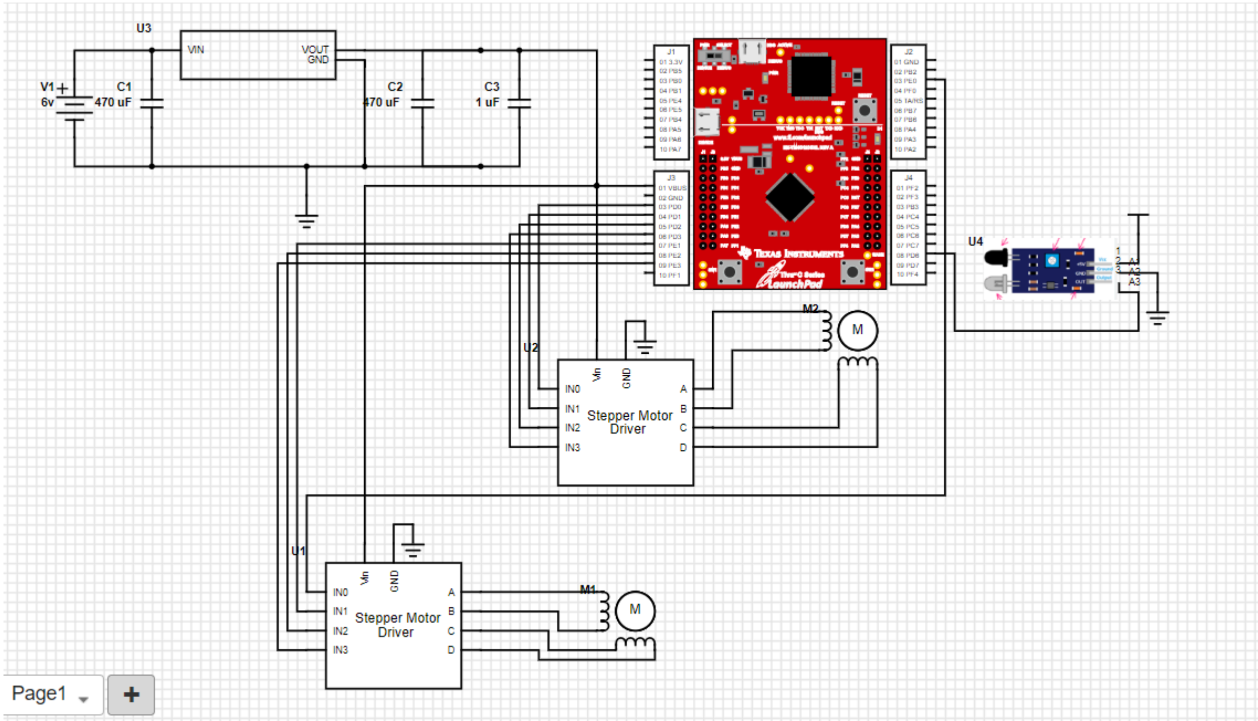
https://drive.google.com/file/d/1d-cMdVRQ0Sl9efi6i3eT-06Kth9WA355/view?usp=drive_link

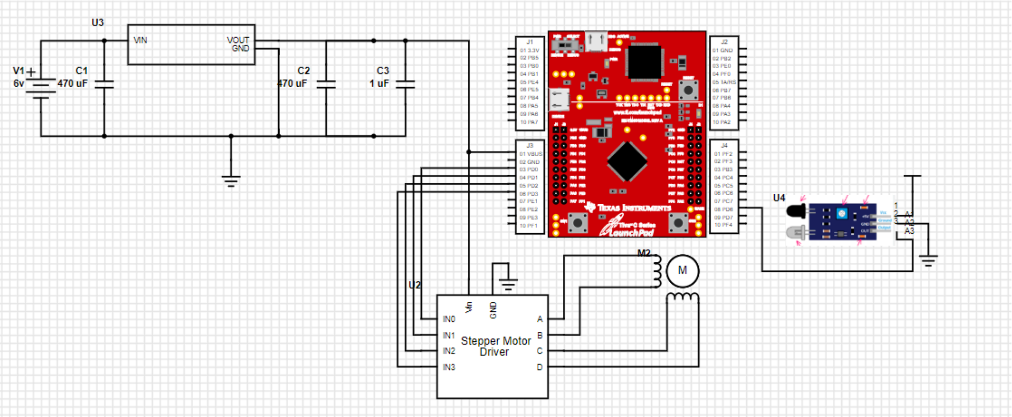
In this lab the operating modes of the car are controlled by two buttons the first sets it to enter the garage where it will move for 720 degrees forward, turn 90 degrees then move forward until the sensor detects it is in front of the door. When the door opens the car moves forward 260 degrees*** .

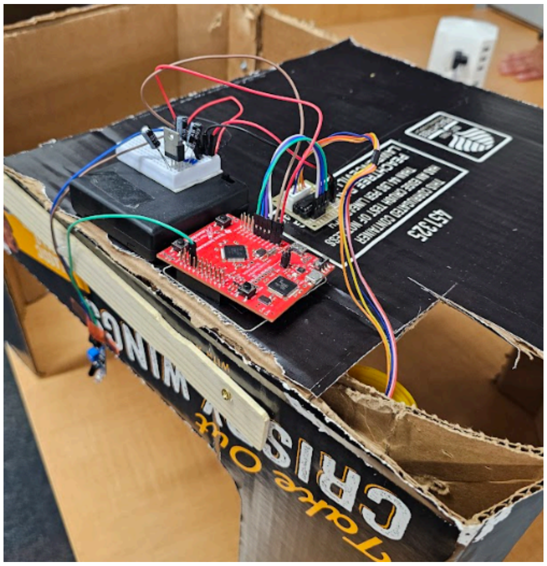
Theory

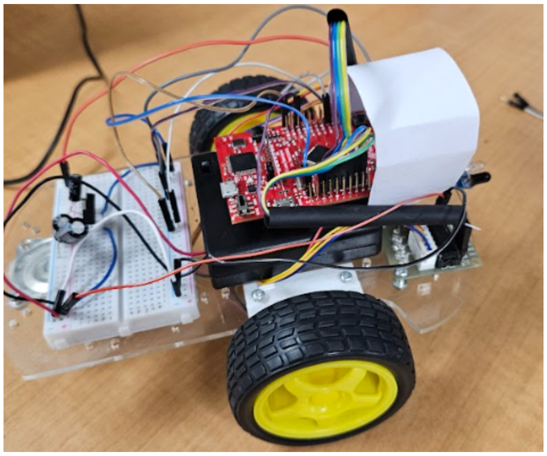
This project uses everything we have learned in this class so far to make the car run and the garage door open. Immediately when approaching this project we needed to understand the micro controller so we could properly initialize all of the port's pins. Our next concern was getting the stepper motor to run. This was dependent on a power supply unit with an LM2940 voltage regulator that allowed the stepper motor driver to run off of 5v rather than from the micro controller itself, which could be dangerous. We needed to be proficient in finite state machines and systick to ensure we could properly wave drive the stepper motor in a way that allowed us to alter the speed and degree that the motor would rotate at. Infrared sensors were also pivotal to our design and we needed knowledge of how waves are transmitted and received by this device to make the system function. Debugging was very useful because the interrupts needed to be configured in a very particular way to ensure they weren't being incorrectly called or not called when they should have been.

Hardware design


U3
VIN
VOUT
GND
V1
6v
C1
470 uF
C2
470 uF
C3
1 uF
J1
01 3.3V
02 PB5
03 PB0
04 PB1
05 PE4
06 PE5
07 PB4
08 PA5
09 PA6
10 PA7
J2
01 GND
02 PB2
03 PE0
04 PF0
05 RESET
06 PB7
07 PB6
08 PA4
09 PA3
10 PA2
J3
01 VBUS
02 GND
03 PD0
04 PD1
05 PD2
06 PD3
07 PE1
08 PE2
09 PE3
10 PF1
J4
01 PF2
02 PF3
03 PB3
04 PC4
05 PC5
06 PC6
07 PC7
08 PD6
09 PD7
10 PF4
Texas Instruments
Tiva C Series LaunchPad
U4
1
2
3
A1
A2
A3
U2
M2
M
Vin
GND
IN0
IN1
IN2
IN3
Stepper Motor Driver
A
B
C
D
U1
M1
M
Vin
GND
IN0
IN1
IN2
IN3
Stepper Motor Driver
A
B
C
D
Page1
+


U3
VIN
VOUT
GND
V1
6v
C1
470 uF
C2
470 uF
C3
1 uF
J1
J2
J3
J4
U4
1
2
3
A1
A2
A3
U2
M2
M
Vin
GND
IN0
IN1
IN2
IN3
Stepper Motor Driver
A
B
C
D





Software design

In this Project we used two tm4c microcontrollers with two different programs. This first program is to operate the car. This works by usings an FSM to control whether the car moves forward, backward, left, or right. Then we have two interrupts. One is for the onboard switches that start the car moving either forward or backward. Then we have an infrared sensor set up as an interrupt. When the sensor detects something the code is interrupted and the car stops. Until there is nothing in front of the car it will not move.

For the second part we programmed a smart house. In this project we used virtually the same method as the previous code but with LED indicators added. The house does not turn on until the tm4c switch 1 interrupt is activated. Then the house waits to detect an object approaching with an infrared sensor. When the sensor detects a car coming the interrupt is called and switches the flag so that the garage begins to open. Once the car door is open we add a small delay until it is called to close again. Then we can open /close the garage with the switch 2 interrupt. Once the door is back open the car can leave. When the garage senses the car leaving with the infrared it will call the interrupt again and close the garage door. When the garage door is moving we use a systick timer that reloads every 200 systick calls to output a blinking red LED. When the garage door is all the way down it will have a solid Green LED indicator and when it is closed it will have a solid Blue LED indicator.

Conclusion

One of the things we struggled with when programming the smart home was initializing the interrupts for the switches and the LEDs. This was because they were both on board and we needed to use 5 pre-defined pins on port F. We immediately ran into problems because PF1-PF3 needed to be outputs while PF0 and PF4 were inputs, so we were very meticulous with our definition of the SwLedInit function. Even once the interrupts were correctly initialized the code was still not functioning how we expected it to. This was because the interrupt started triggering whenever we changed the LED color, not only when buttons were pressed. We fixed this by checking "GPIO_PORTF_RIS_R" in the port F handler. Using if statements to check this register, we were able to create a separate interrupt service routine for each push button, while also masking changes to the LED.

Another challenge we faced was making the LED blink at the right frequency based off of the systick rate necessary to drive the stepper motor. It was easy to make the red LED flash at the desired speed when systick was dedicated to the LED, but the timing was immediately off when we tried to run the motor and the LED at the same time. We overcame this by using a count variable to divide the frequency of the motor in the systick ISR which gave us the desired rate of blinking on the led.