Presentation

This is a shorter section but it is about the overall presentation of my project

Challenge

There was some conflict with this, because with all of my sensors and breadboard extension module, There was little to no space for the carbon monoxide sensor, so I had to cut that, also because the circuit was barely running, as there was only 5v supplied. I didn't realize I could use a power source.

The Challenge

Design an addition to a robot where the addition can take temperature and humidity readings, detect light, detect carbon monoxide, and display information on a screen, using the Arduino Uno as the controller.

Abstract

Abstract

In this design challenge, there were no obstacles until my original rover controller, a Raspberry Pi, was fried because a voltage regulator is needed when you power the Pi with batteries. As a substitute for the Pi, the Arduino Uno was brought in. The Uno is a microcontroller good for beginners in coding and electronics. After assembling the robot and adding the top layer, there was no display/notification for the sensors and their readings so an LCD screen was added for the temperature and humidity sensor along with a buzzer for the CO sensor and a LED for the photoresistor module. The next problem was uploading the code to the Arduino. The computer used originally was not recognizing the Uno for what it was, meaning the code was not able to be uploaded. To take care of that, another Pi was brought into the equation. This time, the Pi was not fried and worked perfectly to upload the code. (Raspberry Pis are small computers) The robot was receiving code just fine. When I tried to upload the project code, I was getting an ‘avrdude: content does not match: verification error’ . Then, all I could do was to double check everything. I disconnected the LCD and tried to upload the code, same error. I disconnected the CO monitor next, uploading the code, and this time there was no error! I rewired everything I had taken off really carefully and there was no error, but the monitor was not doing its job monitoring. After more testing, I found that the CO monitor was not getting enough power because of the amount of things on the breadboard,so it only had enough power to turn on its LED that signifies it is on. At that point I decided adding more power would probably not be safe for the other components so I got rid of the CO monitor and buzzer to make sure the LCD and photoresistor were getting enough power. I uploaded the code again and it worked! The rover was not quite what I imagined it to be, but it still accomplished most of what I wanted it to do, which is a success to me, because after the first error in uploading , I thought it might never work at all, but after a bunch of rewiring, it did.

FINAL VERSION Materials and Procedure

Materials

- lcd (Liquid Crystal Display) screen
- M7 photoresistor module
- DHT11 Temperature and humidity sensor
- Flying Fish CO detector module
- Green LED
- Potentiometer
- 220□□ resistor
- Active buzzer
- Arduino Uno
- Jumper wires
- Small chalkboard
- Various hardware
- Breadboard
- Battery pack
- 8 AA batteries
- Elegoo smart robot car v 4.0 kit
- Arduino cord
- A drill
- A computer

Procedure

1. Assemble the Elegoo kit
2. Drill holes in the top platform and the chalkboard
3. Secure the chalkboard on top of the platform using four screws, eight washers, and eight nuts
4. Secure the breadboard on to the surface of the chalkboard with double-sided tape
5. Put the batteries in the battery box
6. Put double-sided tape on the bottom of the battery box and fix it to the top of the robot car battery
7. Secure the Arduino to the bottom of the Raspberry Pi case using two small screws
8. Fix the Raspberry Pi case with the Arduino inside across from the breadboard using double-sided tape
9. Wire the photoresistor module and LED
    - Photoresistor pin AO to analog zero (A0) on Arduino

- Pin DO to LED anode (positive/longer leg) pin
- Pin GND to negative column on the breadboard
- Pin VCC to positive column on the breadboard
- LED cathode (negative/shorter leg) to negative column on the breadboard

10. Wire the DHT11 temperature and humidity sensor
    - (-) pin to negative column on breadboard
    - OUT pin to digital pin 7 on Arduino
    - (+) pin to positive column on breadboard

11. Wire the CO module and active buzzer
    - Pin DO to Arduino digital pin 8
    - GND pin to negative column on breadboard
    - VCC pin to positive column on breadboard
    - Buzzer positive pin (longer leg) to digital pin ~9 on Arduino
    - Buzzer negative pin (shorter leg) to negative column on breadboard

12. Wire the LCD, resistor, and potentiometer
    - LCD pin one to negative column on the breadboard
    - LCD pin two to positive column on the breadboard
    - LCD pin three to the center pin on the potentiometer
    - LCD pin four to digital pin twelve on the Arduino
    - LCD pin five to negative column on breadboard
    - LCD pin six to digital pin ~11 on the Arduino
    - LCD pins seven to ten; nothing
    - LCD pin eleven to digital pin ~5 on the Arduino
    - LCD pin twelve to digital pin 4 on the Arduino
    - LCD pin thirteen to digital pin ~3 on the Arduino
    - LCD pin fourteen to digital pin 2 on the Arduino
    - LCD pin fifteen to resistor pin
    - LCD pin sixteen to negative column on the breadboard
    - Potentiometer right pin to negative column on the breadboard
    - Potentiometer center pin to LCD pin three
    - Potentiometer left pin to positive column on the breadboard
    - Remaining resistor pin to the positive column on the breadboard

13. Connect power to the breadboard
    - GND pin on Arduino to top of negative column on the breadboard
    - 5V pin on Arduino to the top of the positive column on the breadboard

14. Code the sensors, LCD, LED, and buzzer
15. Upload the code to the Arduino
16. Connect the Arduino to the battery pack with the Arduino cord
17. Test the robot (drive it, light a match near the CO sensor, test the LED and photoresistor, and the LCD screen and DHT11 sensor)

18. Record results

Result

The rover met three out of four standards for this challenge: temperature and humidity readings, detect light, and display information on a screen. There was not enough power to control all of the sensors, so I had to either add another power source or cut a sensor. I had to go with the latter because too much power could damage your components, a fact I learned all too well when the first Pi got fried. Overall, the project did better than I thought it would but it did not achieve what I wanted it to be.

I personally, looking back, don't like this conclusion, but oh well. That was for the deadline.