

Vending Machine Project

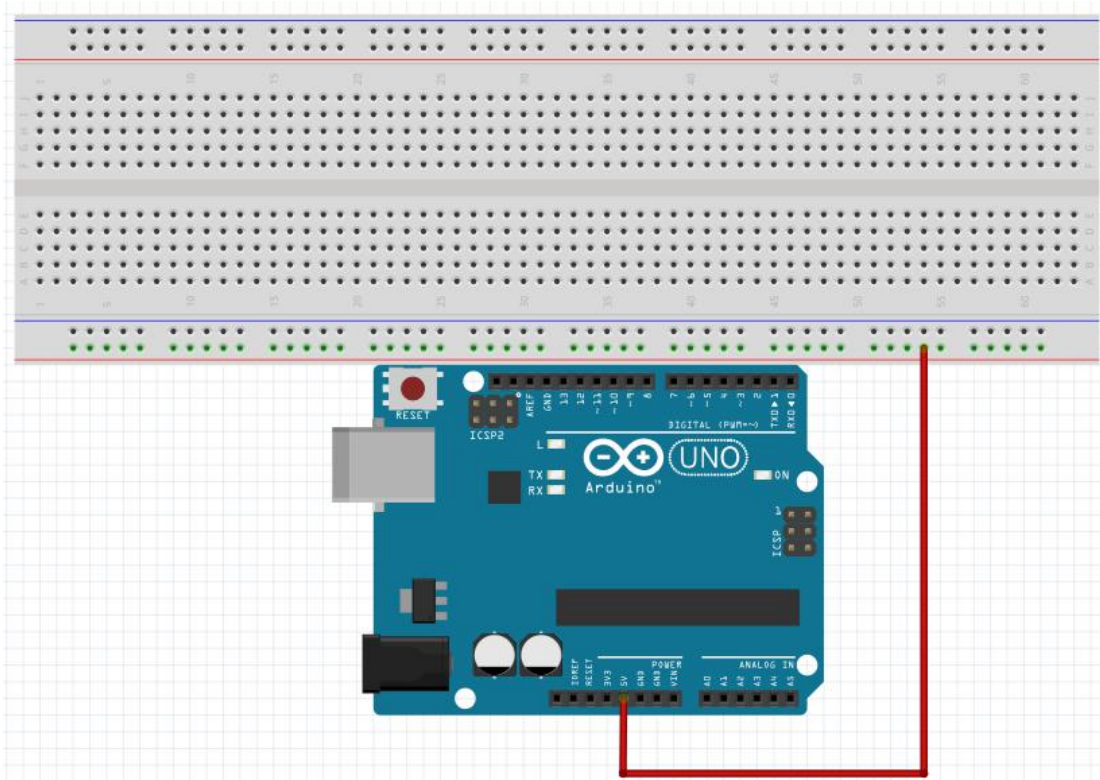
In this project, we will use the Arduino Uno to construct a vending machine type interface. We will use (4) push buttons to signify money inserted. One for a dollar, a quarter, a dime, and a nickel. We will then use (4) push buttons to signify a snack being selected. One for a Snickers, M&Ms, Doritos, and gum. We will then have (1) push button acting as the change return.

Materials

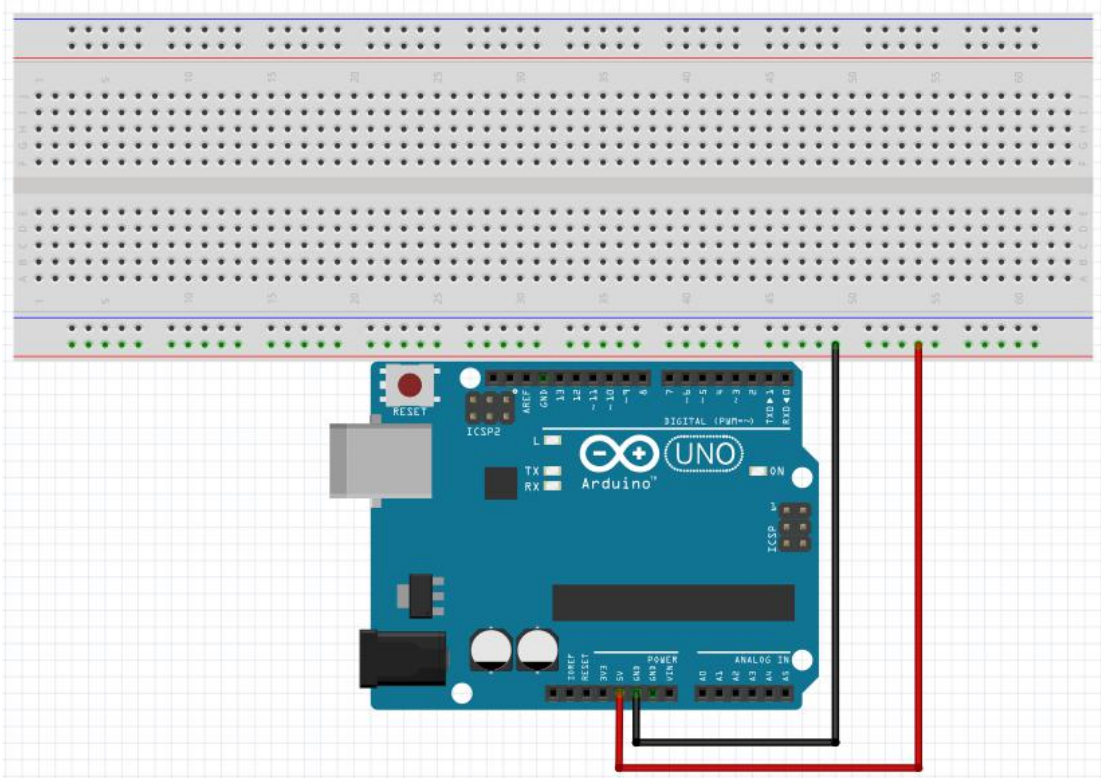
- Arduino Uno or equivalent (Nano has same pinning)
- Breadboard (400 tie point or larger)
- Jumper Wires (x20+) 3-6 inches each
- 10 kohm resistors (x9)
- Push Buttons (x9) <https://www.sparkfun.com/products/97>

Hookup Instructions

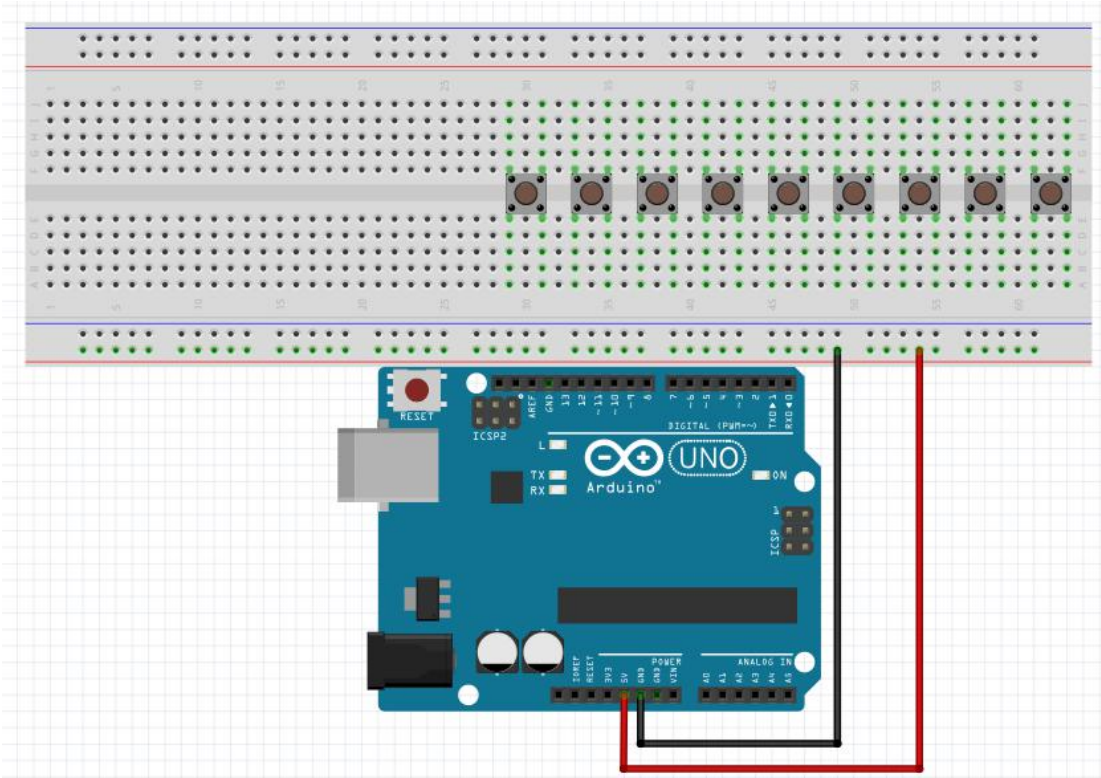
1. Connect a jumper wire from the 5V pin on the Arduino to one of the + rails on the breadboard.



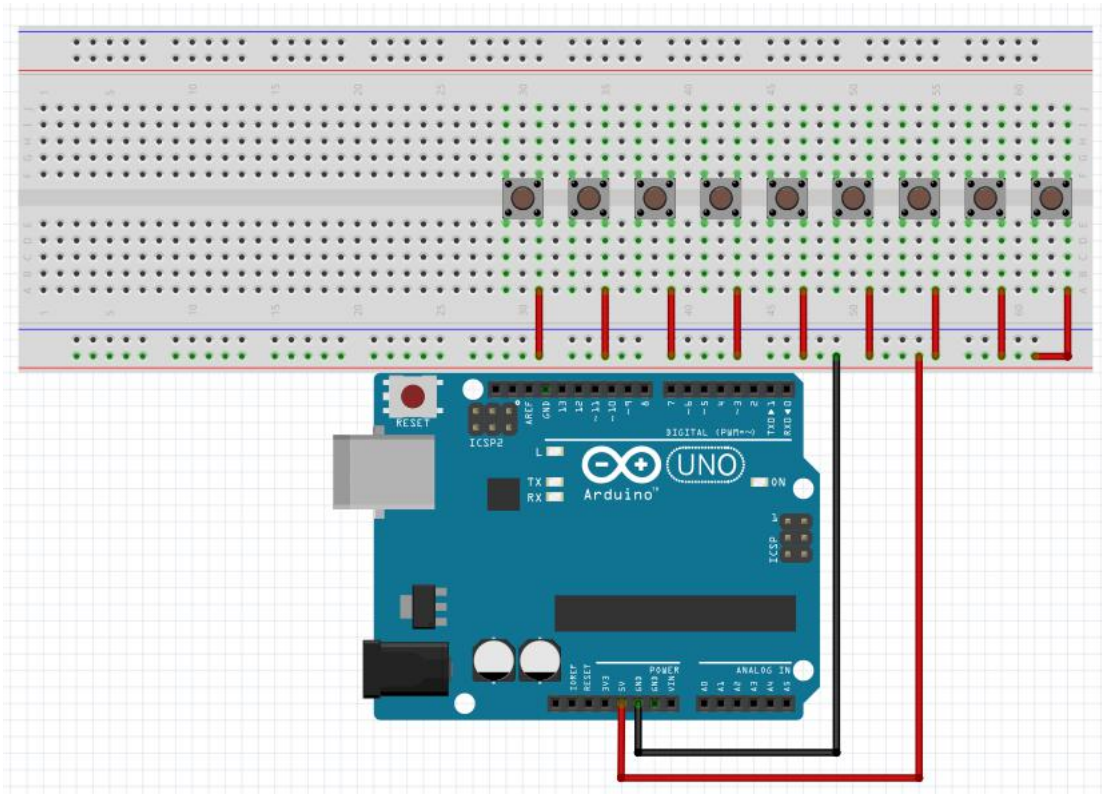
2. Connect a jumper wire from the GND pin on the Arduino to the - rail next to the + rail you chose on the breadboard.



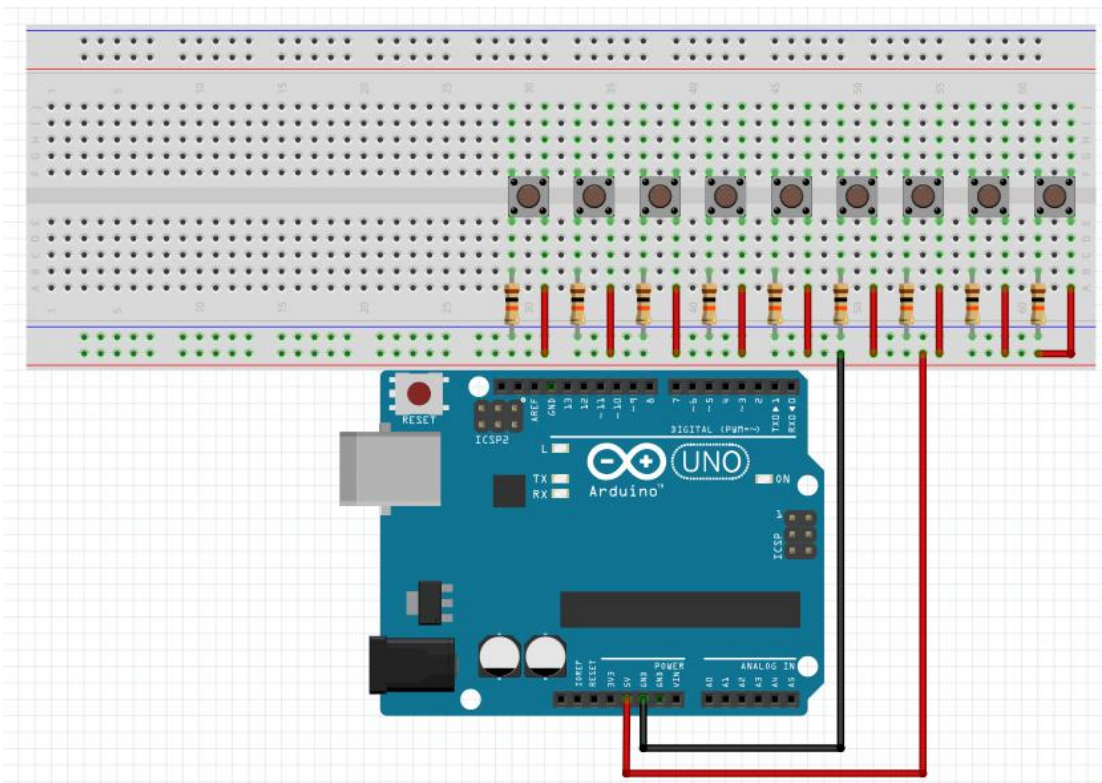
3. Place your buttons on the breadboard straddling the gap between the two sides.



4. Place small jumper wires going from the + rail with the 5V connected to one pin of the buttons.

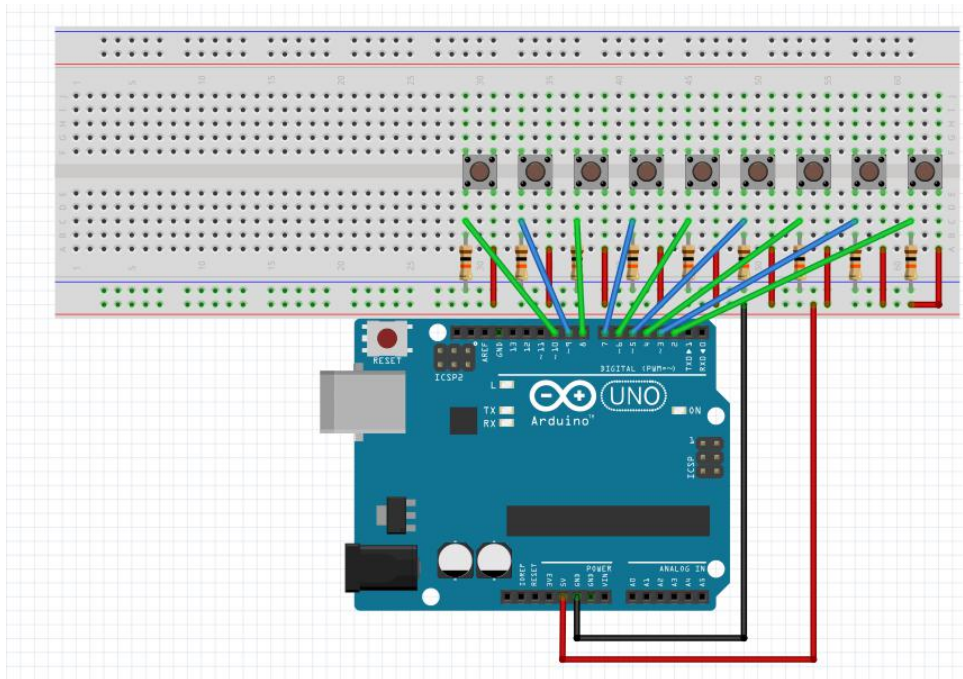


5. Place 10 kohm resistors from the - rail with the GND pin connected to the other pin on the same side of the gap as the 5V is connected to.



6. Place jumper wires from digital pins 2-10 on the Arduino to the same pins that the 10 kohm resistors are connected to, preferable in a sensible order. (The 5 points on the breadboard going across in the

same row are all connected. So on row 15, points 15a, 15b, 15c, 15d, and 15e are all connected.)



Programming Instructions

Overview: There are several ways to program a vending machine type interface utilizing our push buttons. These instructions will give you an idea of what needs to be done, and leave you with figuring out how. I have provided my method, but it's not the only (or easiest) correct way. I highly recommend using comments extensively so you know what each section does and can come back to it later to troubleshoot and know what each pin is, what each state is, etc.

1. First, decide how you want to keep track of each buttons value (dollar, quarter, Snickers, etc.) and how you want to store that (numbers, strings, both, neither?).
2. Next, create any libraries for custom classes you may want (I created a class for Money and a class for Snack, but that's not required).
3. Next, declare any global variables, or variables that you want to be able to use anywhere in the program. These can be your variables for the money and snacks and the value they hold, variables to name your pins to allow you to use things such as `digitalRead(dollar_pin)` instead of the number of the pin (this makes it easier to keep track of), total money received, and any other status indicators you may think of along the way.
4. Then, in the setup routine, initialize the serial interface to allow you to print to the serial monitor. Also, this is a good place to declare what

type each pin is (`pinMode(dollar_pin, INPUT)` for example). This is also a place where you can set initial values for any variables and give the user initial instructions (Enter money).

5. Next, decide if you want to create individual routines for different functions (reading the money pins, snack pins, change return, etc.) or do it all as one large routine in the loop routine. I created several separate routines and would advise this. It allows you to type a set of instructions once then call them whenever you want to execute them.
6. Next, you need to read what money is being put in, whether this be a separate routine or the beginning of your loop function. You need to check one pin at a time to see if a dollar, quarter, dime, or nickel has been inserted. But, what about if two buttons are pressed at once? You need to make sure only one button is pressed at a time and let the user know if two are pressed at a time. (`Serial.println("Only one button at a time please!")`)
7. Then, what about if the user holds the button down. You don't want to add \$X.XX every time the Arduino scans! (This is in the microsecond range by the way.) Find a way to only register each button press once. (Check out the while function)
8. Now that you've read the button being pressed, you need to add that money value to the total and store it in your total money in variable!
9. Keep in mind it might be a good idea to let the user know that they inserted X amount and what their total is! (`Serial.println("You inserted X, your total is Y")`)
10. After you figure all this out for one button and money value, it can be copy and pasted and the pin being read changed and the amount added to the total changed to expedite the other three buttons.
11. Next, you need to see if the user has pressed a snack button! Keep in mind the same considerations as the money buttons, one at a time and only register it once.
12. Along with the considerations of the money button, when a snack button is pressed, you need to check to make sure they've inserted enough money. (`Total - snackCost >= 0`)
13. If they haven't inserted enough money, let them know! (`Serial.println("You haven't inserted enough for that")`)
14. If they have, tell them the item was vended and "return" the change!
15. To simulate returning the change, take `Total - snackCost` and tell them that was returned. Then, set the total to 0.
16. Next, you need to check if the change return button is pressed, again, only registering one press. If it has, return the total and set the total to 0.