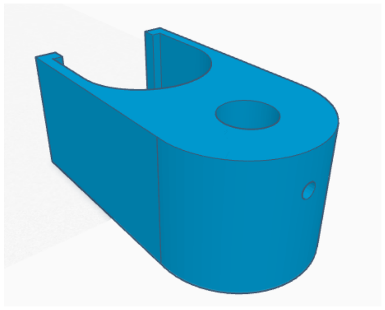
CS7.1 AirDrop Mod

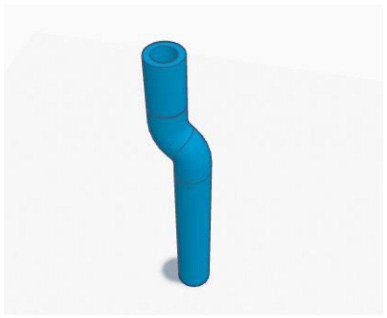
The airdrop mod consists of two 3d printed parts, an air solenoid valve and a switch/relay to active the solenoid. The air pressure on the compressor seems to work best when set between 25 – 40psi.

AirDrop – Bracket



This can be printed in any semi-flexible filament such as PLA+ or PLA. PETG may be too brittle and cause the clips to break. This module is printed with 20% infill with layer height of .2.   It is designed to snap on to the front of the feed bracket.

AirDrop – Air Nozzle



This part connects to the .25"/6mm air hose using a friction fit. The part can be printed in any material and because it is so narrow and tall, it is recommended to use at least a 10 line brim to help it stick to the print surface

Air Solonoid

There are many options for air solonoids but I recommend using a 12v Normally Closed solenoid so you can use the same power source as that currently feeding your electronics system.

https://amzn.to/41BXuTY

You will also need a length of airhose and the connectors to attach the solenoid. I used the following kit but you can probably source the parts you need for your specific compressor setup

6mm pneumatic air hose pipe kit.

https://amzn.to/3MVGpjE

Mosfet Switch or Relay

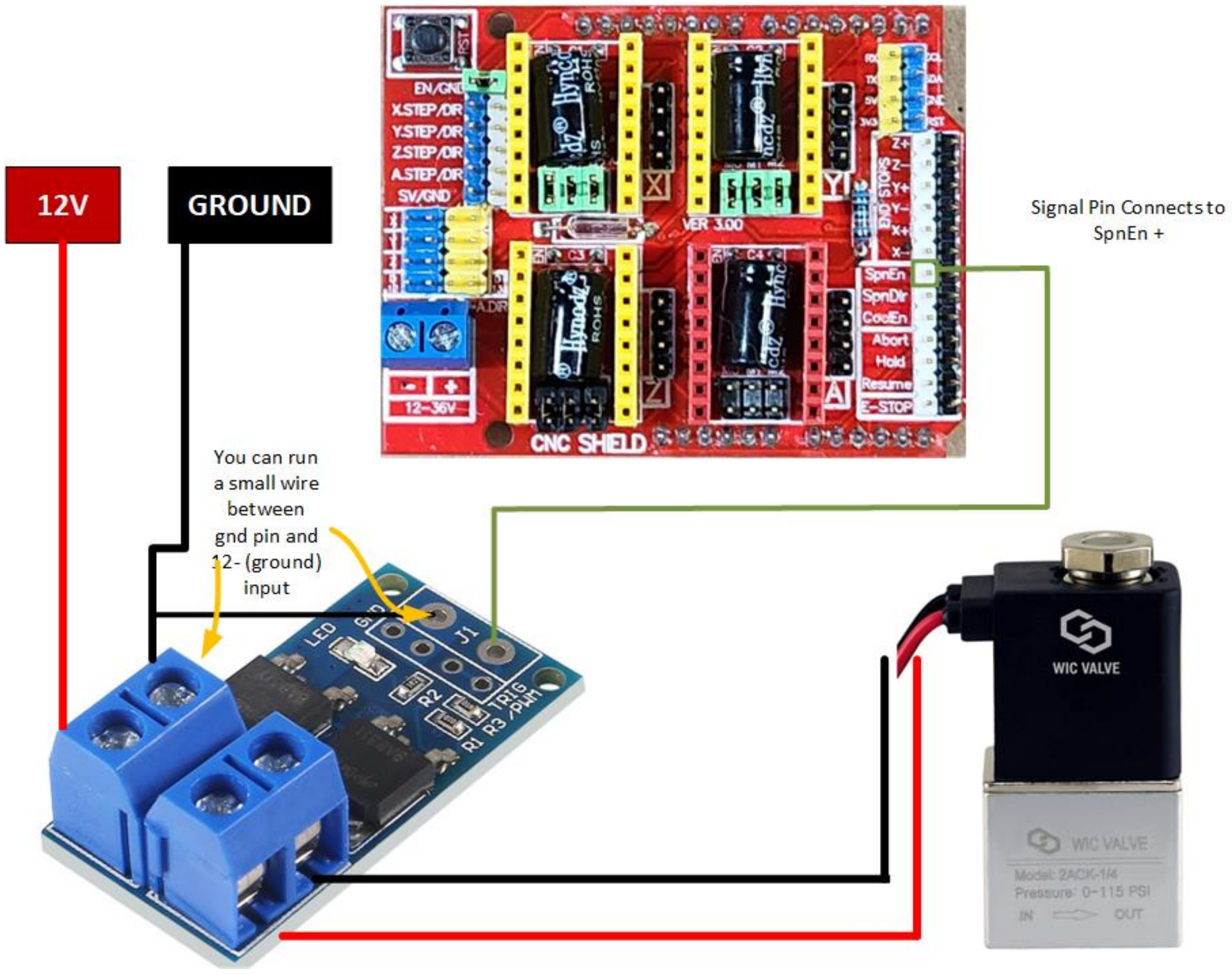
This is another choice to make as there are many solutions here. The Arduino has a signal pin on pin 12 which sends a 5v signal at the end of each feed cycle. The solution you use should be controlled a 5v signal and be able to handle a 12v 10w load.

I used a very inexpensive MOSFET Transistor switch and it works very well and was small enough to cram into the electronics housing.

https://amzn.to/3Ac9f7Y

Wiring Schematic

The below schematic is the wiring diagram for this particular switch but wiring should be similar for any relay. Note, we are running 12v power and Ground from our 12v rail and ground rails.

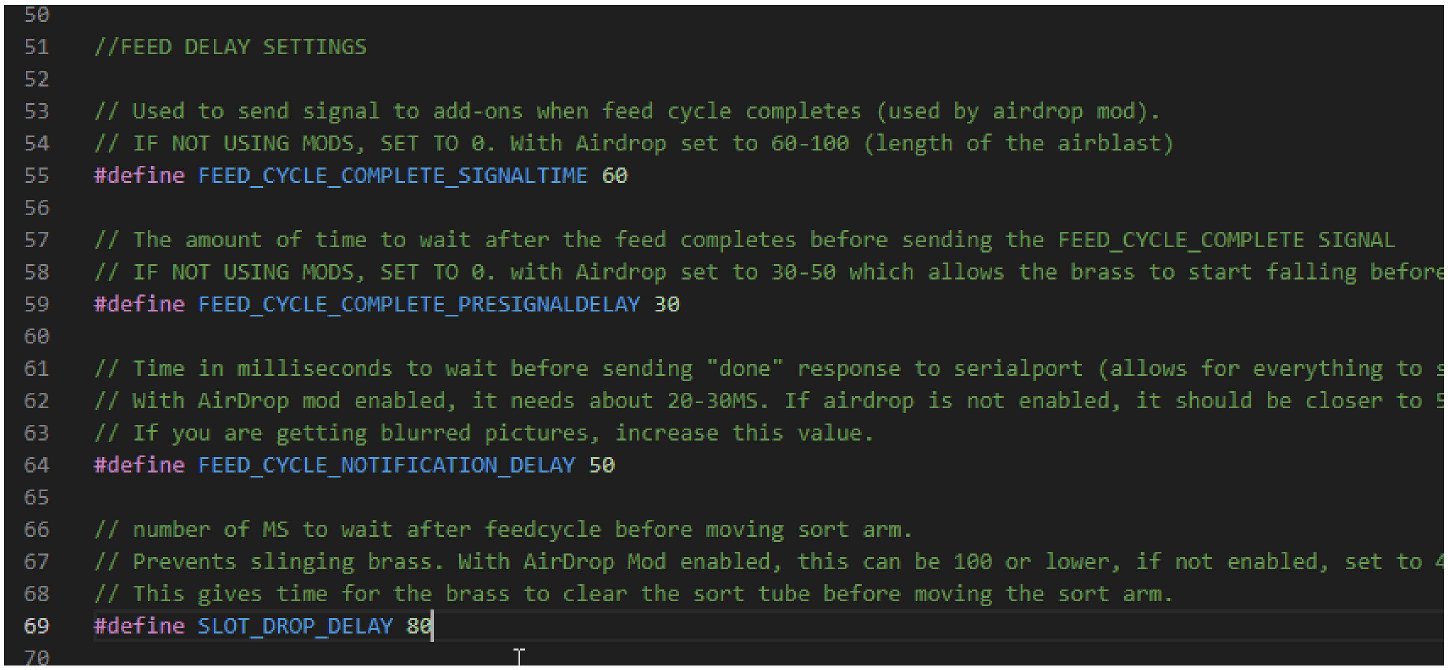


Configuring the Arduino

Finally, you will want to adjust the Arduino code to maximize performance in your system.

The following four parameters control the delay settings.

- The length of the air blast is controlled by **FEED_CYCLE_COMPLETE_SIGNALTIME** (line 55) and is in milliseconds
- The time to wait after feed stops before issuing the blast is **FEED_CYCLE_PRESIGNALDELAY.** Adding a bit of time here allows for the brass to start dropping before sending the air blast.
- **FEED_CYCLE_NOTIFICATION_DELAY** is the time to wait after the blast before sending the "done" signal back to the software. This is used to allow time for vibrations to clear before taking the next picture. Setting this too low may result in blurry pictures and too high is just a waste of time.
- **SLOT_DROP_DELAY** is the minimum amount of time to wait before moving the feed arm in between sorts. Without airdrop enabled this is usually set to about 450-500ms to allow brass to clear the tube before moving the arm.



```
50
51    //FEED DELAY SETTINGS
52
53    // Used to send signal to add-ons when feed cycle completes (used by airdrop mod).
54    // IF NOT USING MODS, SET TO 0. With Airdrop set to 60-100 (length of the airblast)
55    #define FEED_CYCLE_COMPLETE_SIGNALTIME 60
56
57    // The amount of time to wait after the feed completes before sending the FEED_CYCLE_COMPLETE SIGNAL
58    // IF NOT USING MODS, SET TO 0. with Airdrop set to 30-50 which allows the brass to start falling before
59    #define FEED_CYCLE_COMPLETE_PRESIGNALDELAY 30
60
61    // Time in milliseconds to wait before sending "done" response to serialport (allows for everything to s
62    // With AirDrop mod enabled, it needs about 20-30MS. If airdrop is not enabled, it should be closer to 5
63    // If you are getting blurred pictures, increase this value.
64    #define FEED_CYCLE_NOTIFICATION_DELAY 50
65
66    // number of MS to wait after feedcycle before moving sort arm.
67    // Prevents slinging brass. With AirDrop Mod enabled, this can be 100 or lower, if not enabled, set to 4
68    // This gives time for the brass to clear the sort tube before moving the sort arm.
69    #define SLOT_DROP_DELAY 80
70
```

Once you have set your parameters, you can test in the Arduino console by sending commands like 0 or any number between 0 – 7. Also commands like "test:20" can be helpful as it will run 20 brass through as quickly as possible.