Instructions for Ball Motion

for Generation II Ball Printer

Prepared by George H. VanHoose Sr.
For Mike Litscher
V1
01/04/2023

- Wiring
  - Refer to the wiring diagram and pin assignment
  - This device utilizes 3 of the same model of motor driver
    - DRV8825
      - by Pololu Corporation
      - https://www.pololu.com/product/2982
    - These drivers do not require any power source, since their supply voltage is from voltage available on other pins. That is why there is no VCC pin dedicated to the board itself.
    - These drivers utilize what is called "Complementary Logic"
      - All complementary pins on the DRV8825 are
        - Sleep (SLP)
          - 0V input de-energizes the motor and driver
          - 3.3V input energizes the motor and driver
          - Note: When supplied with 3.3V on this pin, the stepper motor has the potential to become very hot.
        - Reset (RST) – for our purposes this is not critical, but it has worked thus far being tied to the complementary-SLP pin.
        - Enable (ENA)
          - 0V input turns the driver on
          - 3.3V input turns the driver off
          - This should be tied to ground for our immediate purposes, and probable long-term as well
      - This "Complementary Logic" is not evident simply by looking at the board, so it may cause confusion if it were not explained here.
    - These drivers utilize a current-attenuating screw
      - Used to manage power available to the motors
      - This helps prevent power loss and overheating
      - A good rule of thumb is to adjust the screw of a driver until the motor just barely works. There
    - The DRV8825 should have a heat sink attached.
    - The FLT pin on the motor driver is unused, it can remain open and unattached.
  - Motor Power
    - Use a 12VDC power source, limited to 2 Amperes
    - A 100 microfarad capacitor should be placed across the positive and negative wires from the power source.
      - if using an electrolytic "barrel-type" capacitor, ensure the negative/cathode/shorter//"-" lead is tied to the ground of the power source
      - electrolytic capacitors can explode if wired backwards
  - Wire requirements

        - Assumption for this section is that the Teensy4.1 and motor drivers are being mounted on a prototyping breadboard
        - multi-colored jumper wires, typically 20AWG, are recommended
        - 45 jumper wires are required to connect the motor drivers
            - 33 male-to-male
            - 12 male-to-female
            - Recommended each jumper wire is 8" or longer
                - Such long wires can be eliminated in production models
                - Long wires are easier to move out of the way during development and testing
        - 4 additional jumper wires, male-to-male, are required to supply power to the power rails of the breadboard.
            - 2 to supply GND and 3.3V from the Teensy
            - 2 to supply 12V and GND from the motor power supply
        - In total:
            - 37 male-to-male jumper wires
            - 12 male-to-female jumper wires
            - Recommend having extras on-hand
    - Teensy4.1 should draw enough power through its serial connection with the user's computer

● Operating the device

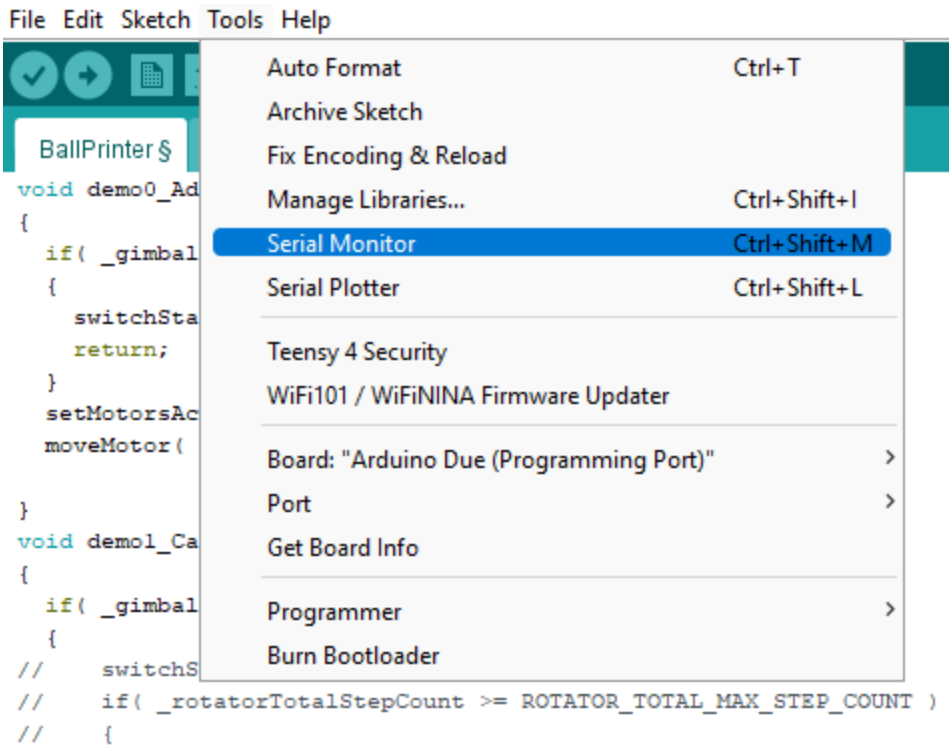
- Assumption is that instructions for loading a program onto a Teensy4.1 are not required. Assumption is that the program has been flashed and is running.
- Device Modes & "BEGINNING_STATE"
  - Use "ctrl + F" to find a variable called "BEGINNING_STATE" (Line 81 of the program)
  - "BEGINNING_STATE" determines the behavior of the device at runtime.
  - For development and testing purposes 5 demo states were created and labeled as "DEMO0," "DEMO1," and so-on.
  - "BEGINNING_STATE" appears in code as such:

    ```
    BEGINNING_STATE = DEMO5,
    ```

  - "BEGINNING_STATE" can also be changed to:

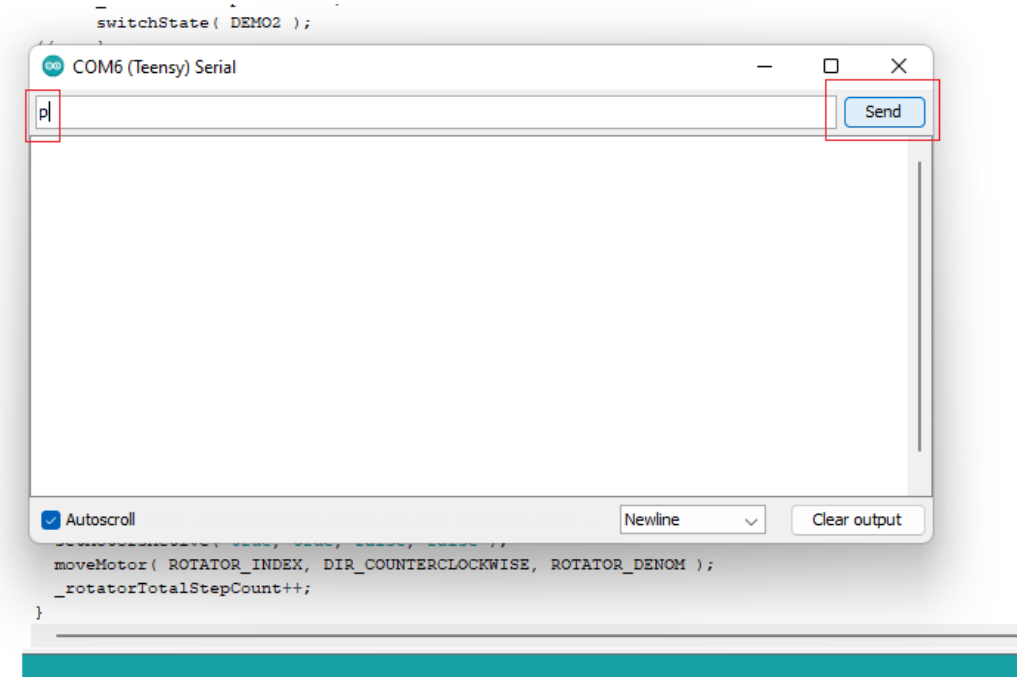
    ```
    BEGINNING_STATE = DEMO0,
    ```

  - Demo Modes
    - When "BEGINNING_STATE" is initialized to DEMO0
      - Device will move the ball through a full sweep of the gimbal motor, then back, then rotate the ball a certain amount before resuming another full sweep of the gimbal motor
      - This cycle will continue indefinitely – very useful for testing how the ball holds center.
      - This is likely to be the mode used-most for developing a solution to the current (as of 1/4/2023) "hold-stable" issues
    - When "BEGINNING_STATE" is set to DEMO5
      - Clamp motor will run for 1 second, then stop
      - DEMO5 state does nothing else
      - Run device with "BEGINNING_STATE" set to DEMO5 in order to clamp the ball, or to tighten the clamp if it gets loose over time
- Serial Monitor
  - Assumption for this section is that the device is currently running in DEMO5 state
  - In the Arduino IDE, under "Tools" in the Menu Bar, is an option for "Serial Monitor"


File Edit Sketch Tools Help
BallPrinter §
void demo0_Ad
{
if(_gimbal
{
switchSta
return;
}
setMotorsAc
moveMotor(
}
void demo1_Ca
{
if(_gimbal
{
// switchS
// if(_rotatorTotalStepCount >= ROTATOR_TOTAL_MAX_STEP_COUNT)
// {
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
Teensy 4 Security
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Due (Programming Port)"
Port
Get Board Info
Programmer
Burn Bootloader

(next page)

- When the Serial Monitor window appears, typing 'p' into the serial input and clicking "Send," like so:

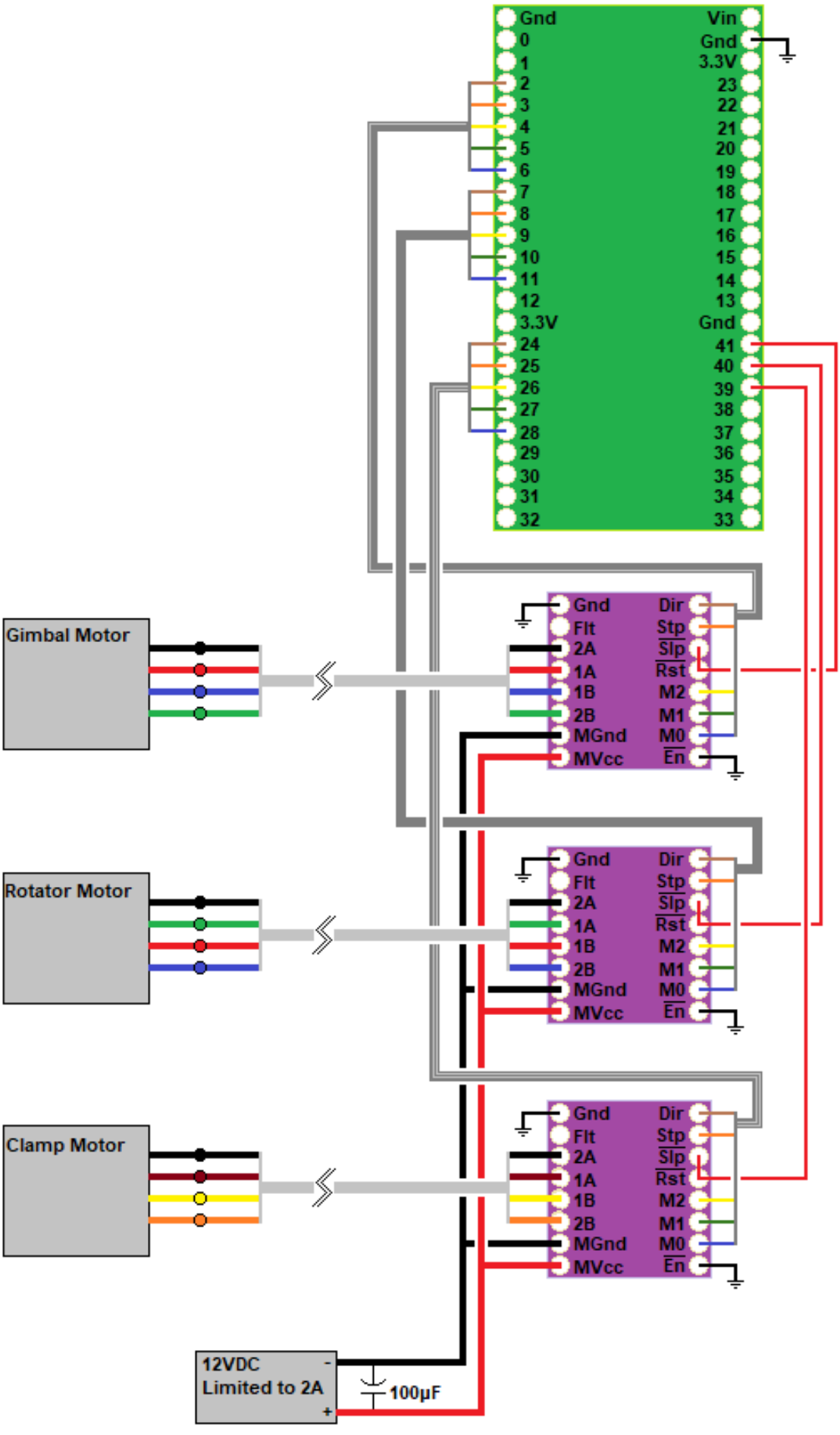


  after the clamp motor stops moving will make it move for another second.

- BEWARE: The software provided for this functionality includes several aspects which are not related to motor movement. This device also manages data being sent to the printhead controller, including buffering and communicating with the printhead. As such, modifications to the program as-is should be avoided except changing the assignment of the "BEGINNING_STATE" variable between "DEMO0" and "DEMO5"
- NOTE: Many lines of code are commented-out
  - preceeded by a `//` or enclosed between `/*` and `*/` demarkers.
  - These are either developer notes, debugging tools, or code that has been modularized elsewhere in the program.
  - When Commented-out, these lines are not built in to the program that is executing on the Teensy4.1

- Motor heat management
    - Motors are programmed to de-energize if their function is not critical for the current state of the device. For instance:
        - Clamp motor will de-energize unless it is turning the worm gear;
        - This prevents excessive power loss (which would be substantial) and prevents heating of the motors
    - De-energizing is achieved by providing a logic-high signal on the complementary SLP pin
        - The complementary on-off signal is controlled by the Ball Printer software
        - If "always energized" is desired
            - simply tie the complementary SLP pins directly to the 3.3V pin of the Teensy
            - it should only be for a short period of time
            - Be sure to disconnect the SLP/RST pins from the 3.3V pin once the motor is no longer required to be "always energized," else they will heat up and waste significant amounts of power.

- Appendix
    - Wiring diagram (next page)


Gnd
0
1
2
3
4
5
6
7
8
9
10
11
12
3.3V
24
25
26
27
28
29
30
31
32
Vin
Gnd
3.3V
23
22
21
20
19
18
17
16
15
14
13
Gnd
41
40
39
38
37
36
35
34
33
Gimbal Motor
Rotator Motor
Clamp Motor
Gnd
Flt
2A
1A
1B
2B
MGnd
MVcc
Dir
Stp
Slp
Rst
M2
M1
M0
En
12VDC
Limited to 2A
-
+
100µF