# Science Board Description and Pinout

(Datasheets at the bottom)

Current Board Dimensions: 147.828mm x 53.594mm, 20mm height

This is an alternative design to the science board where an external regulator is used instead of an onboard IC regulator. (Kudos to Sam for this idea!)

Pros:
- Comes out to be cheaper if we're looking to manufacture a backup board since the boards can use the same power supply
- The external supply can protect itself better than the onboard solution if there is a fault on the board, saving us another expense in a bad situation
- Gets around the problem of not having enough current to run both steppers at the same time (6A max. for the onboard solution vs. 10A max for external supply)
- The onboard solution is being run very close to its maximum all the time, which could be a problem
- The board would have less heat stress
- The planes simplify a lot since there aren't two channels to worry about
- The code controlling the steppers doesn't have to waste time turning the steppers on and off
- We have less control over the external regulator which means less places to mess up

Cons:
- The external regulator is more expensive if we will only manufacture one board
- External regulator takes up most of the space available to us in the science module
- We have less control over the external regulator since it handles fault conditions on its own (if there is a fault, the board simply loses 5V power as opposed to the Arduino shutting the 5V IC's down itself)

External Regulator: SD-50A-5 https://www.meanwell.com/Upload/PDF/SD-50/SD-50-SPEC.PDF

**Update(10/05/2021):** The voltage for the Peltier cooler and the CCD sensor is now handled by a linear regulator.

**Update(06/05/2021):** The board has connectors for a LED strip and a solenoid.

LED Strip: BXEB-L0340U-40E0750-C-C3 (the model is different from the link, check the datasheet)
**https://www.digikey.com/en/products/detail/bridgelux/BXEB-L0590U-50E1500-C-C3/10279841**

D32 - LED_CONTROL : Active low LED control

Solenoid: https://bc-robotics.com/shop/large-push-pull-solenoid/

D22 - SOLENOID_ON: Active high solenoid control

*Fault Detection/Shutdown Lines:*

**Update(18/03/2021):** The board now has an onboard relay driven by a beefy BJT.

D51 - POWER_ON: Set high to close relay, enabling power to reach the rest of the board. (The relay is normally open.)

*Steppers*

The steppers are driven with DRV8834 controllers, with each having one fault indication line. The numbering is arbitrary for now, and the steppers should be named later on depending on where exactly they will be.

D38 - notFAULT1: High when stepper #1 is not in a fault condition.

D36 - notFAULT2: High when stepper #2 is not in a fault condition.

D44 - STEPPER1_notENABLE: Stepper #1 is enabled when this pin is low.

D50 - STEPPER2_notENABLE: Stepper #2 is enabled when this pin is low.

*Stepper Control:*

The DRV8834 stepper drivers are being used in "Indexer Mode" and wired for 16 microsteps/step resolution. When in this mode, the chip sorts out the PWM internally and only needs a step signal and direction data. Again, the numbering is arbitrary for now and the steppers should be named later depending on what exactly they will do.

Stepper #1:

D42 - STEPPER1_STEP: Stepper will advance one step on the rising edge of this pin.

D40 - STEPPER1_DIRECTION: Reverses stepper direction on rising edge.

Stepper #2:

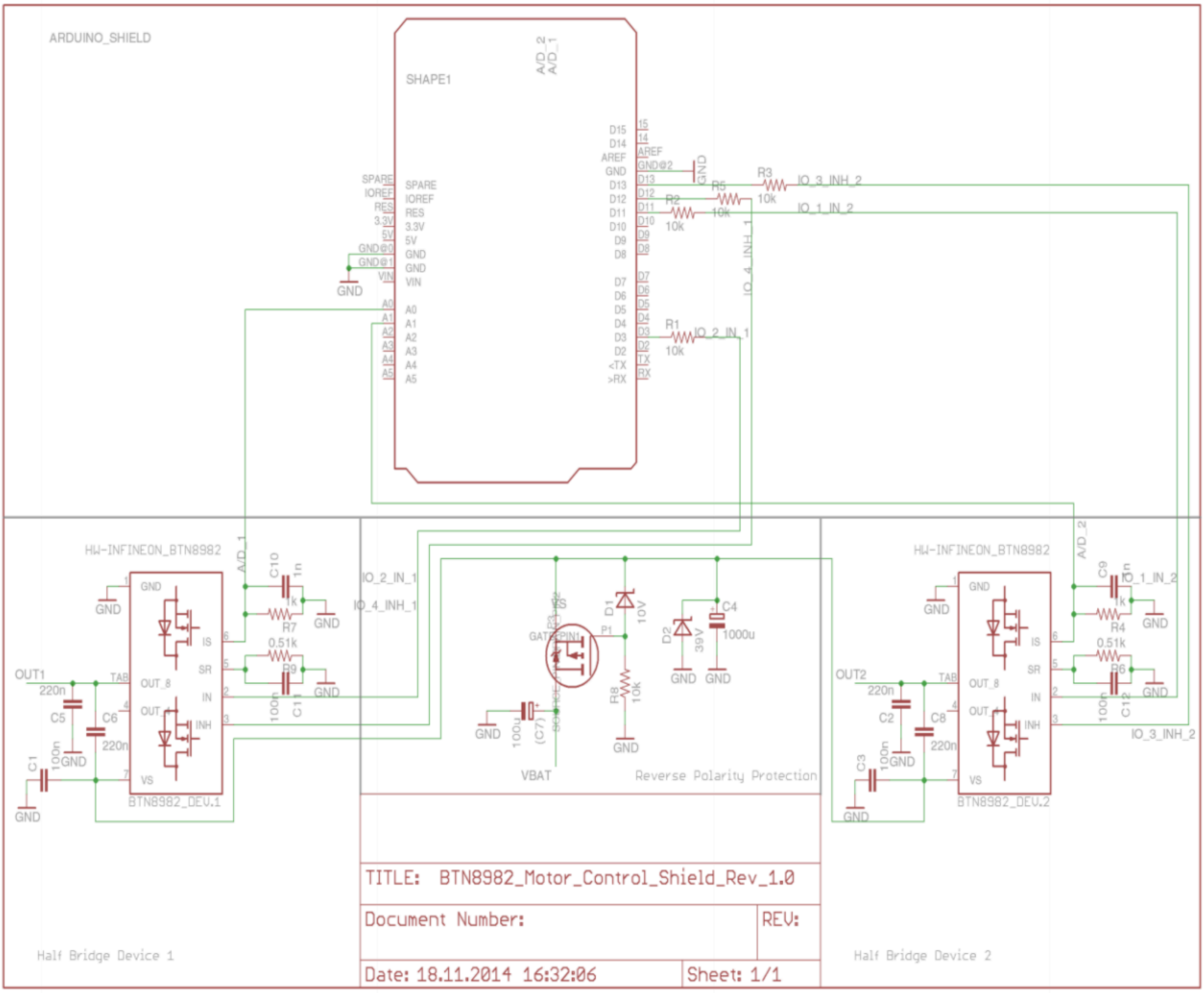
D48 - STEPPER1_STEP: Stepper will advance one step on the rising edge of this pin.

D46 - STEPPER1_DIRECTION: Reverses stepper direction on rising edge.

*Brushed DC Motor Control:*

To future proof the board against different motor choices, two high current bridges (BTN8982TA) are used. The manufacturer (Infineon) actually has an Arduino shield, so the pinout is exactly the same as what they used in their product. See the schematic: https://www.infineon.com/dgdl/Infineon-Motor_Control_Shield_with_BTN8982TA_for_Arduino-UserManual-v02_00-EN.pdf?fileId=5546d4624ca27d02014cb20b89867eed

The pinout is the same as that schematic:




*Laser:*

Since the laser has a 250mA current requirement and no control logic, it is switched with a BJT transistor in a common emitter configuration. This means that the pin is **active low.**

D8 - LASER1_CONTROL: Laser is on when this pin is low.

*CCD Sensor Control*

**Update (10/05/2021):** As we looked into the clock, there have been pinout changes. (This is the new version)

The CCD sensor picked by Science is TCD1304DG with a cryptic datasheet. There is much work to be done to figure out what we need to do in order to get data from it, but for now they are wired like so:

A3 - OutputSignal_out: “out” as in “out to Arduino”.

D2 - phiM(CCD_Clock): Clock output to CCD sensor.

D10 - SH: “Shift Gate” (???)

D9 - ICG: “Integration Clear Gate” (???)

*Datasheets:*

(Stepper Driver) DRV8834: *https://www.ti.com/lit/ds/symlink/drv8834.pdf?ts=1613165364751&ref_url=https%253A%252F%252Fwww.google.at%252F*

(Motor Control) BTN8982TA: *https://www.infineon.com/dgdl/Infineon-BTS8982TA-DS-v01_00-en.pdf?fileId=db3a30433fa9412f013fbe32289b7c17&sd=t*

(Lasers) CPS532: *https://www.thorlabs.com/thorproduct.cfm?partnumber=CPS532*

(CCD Sensor) TCD1304DG: *https://toshiba.semicon-storage.com/ap-en/semiconductor/product/linear-image-sensors/detail.TCD1304DG.html*

(Relay) J115F11AH12VDCS61.5U: *https://www.citrelay.com/Catalog%20Pages/RelayCatalog/J115F%2050amp.pdf*

(Linear Regulator for Peltier Cooler + Sensor) NCP57302: https://www.onsemi.com/pdf/datasheet/ncp57302-d.pdf