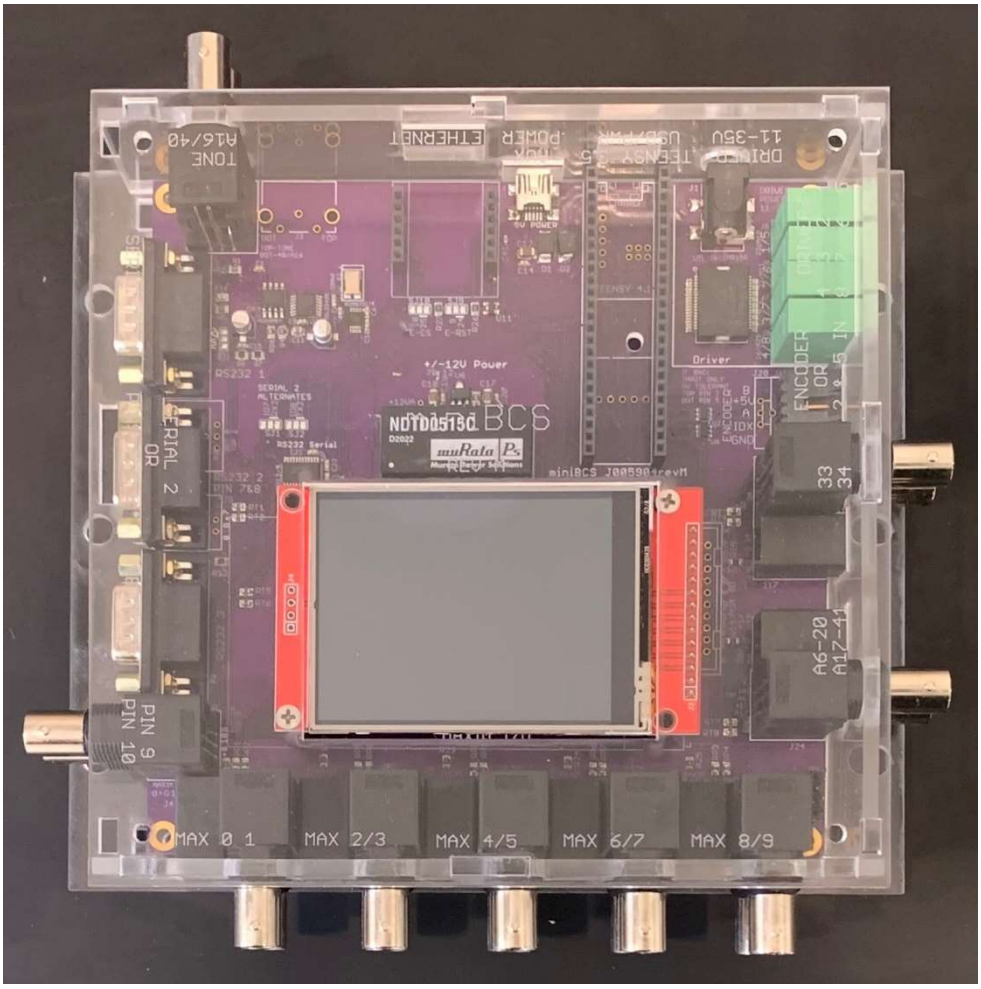
hhmi | janelia

# miniBCS-4




Steve Sawtelle
Engineer, ID&F

hhmi | janelia

# Contents



## System Overview

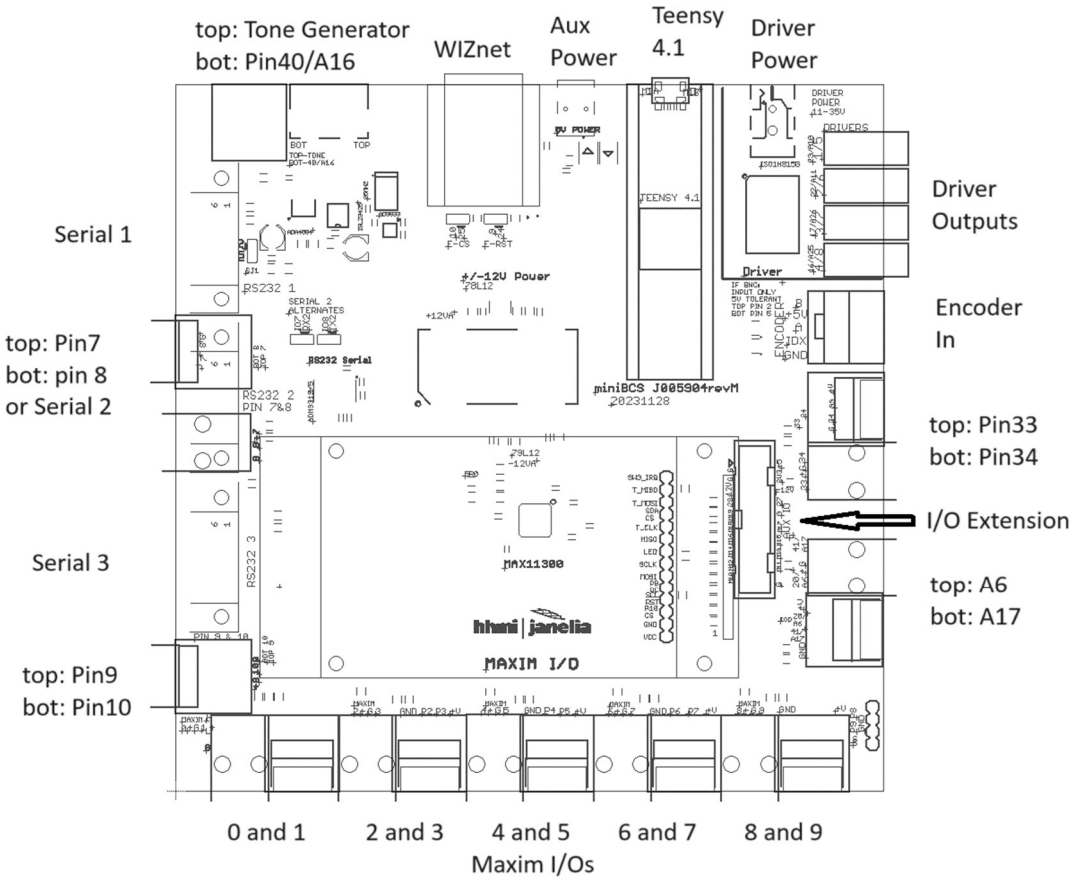
The miniBCS-4 is a compact, general purpose multi-function I/O system that makes it easy to access various peripherals found on the Teensy 4.1 and provides a powerful platform to control real time experiments. It is highly adaptable in terms of types of outputs and connectors and is designed to be easy to use, inexpensive, and easy to customize.

## Hardware Development

The miniBCS-4 is designed around the Teensy processor system. Currently, the Teensy 4.1 processor board is being used. The Arduino programming environment makes writing and downloading code easy. A MAX11300 integrated circuit is used to provide 12-bit bipolar ADC/DAC and other configurable I/O not available on the Teensy. The various input/outputs (IOs) are brought out to various connector options including DB9, BNC, latch connectors, and RJ9. A waveform generator is also built in. A color graphic LCD with touchscreen can be added to provide a simple user interface. The system is designed so that only the required features and connectors needed for a particular application are added to the base board, minimizing cost and assembly time. The enclosure can be made from laser cut acrylic, which makes it easy to configure the unit, house the electronics, make attachment points, and engrave connector labels.

Capabilities (some are mutually exclusive):

- Up to three RS232 serial ports
- Host USB serial port (also used for programming)
- Ethernet via WIZnet module
- Incremental Encoder (optional index pin) with 5V tolerant inputs
- Up to 10 12-bit DACs with +/-5V, 0-10V, or -10 to 10V ranges (MAX11300)
- Up to 10 12-bit ADCs with 0-2.5, +/-5V, 0-10V, or -10 to 0V ranges (MAX11300)
- Up to 10 I/Os with programmable thresholds or output ranges (MAX11300)
- 8 isolated solenoid drivers with 11-35 Volt power supply range
- Frequency Generator with range of 1Hz to 10MHz, sine, triangle, or square wave
- I/O Extension providing two direct processor I/Os and ten Maxim I/Os
- Up to 9 I/Os that connect directly to the processor that can be programmed for:
    - Interrupts
    - 0-3.3V ADC
    - 3.3V digital output
    - 3.3 digital input
    - PWM or servo drive
    - Various timer/counter needs



# Assembly

Due to the relative low cost and relative assembly difficulty of the surface mount parts on the board, it is advisable to have the basic board assembled by a small run assembly house. For 10 boards, this can be done for about $300 a board. For hand assembling partial boards, there is information at the end of the document. The real customization of the board is done by hardware jumpers, choice of connectors, and firmware. The jumpers and connectors can be easily soldered by hand by novices.

**Note on Jumpers**

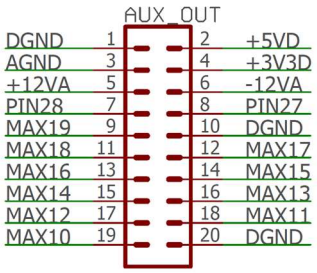
Jumpers are included on the board to allow various configurations. There are two types of jumpers; I/O select, and tone select.

**Pin selection jumpers** have a connection to the default I/O. If the alternate function is needed, then the default connection between the pads needs to be cut and a short between the other function and the middle pad needs to be soldered in. These are used to select alternate CPU pins to be available at each connector to maximize I/O options. The following selections are available:

| I/O | Jumper Name | Default | Option (cut and jumper) |
|---|---|---|---|
| RX2/IO7 | SJ1 | RS232 #2 Receive | Pin 7 (PWM, digital I/O) |
| TX2/IO8 | SJ2 | RS232 #2 Transmit | Pin 8 (PWM, digital I/O) |
| Eth-RST/IO9 | SJ9 | Pin 9 (PWM, digital I/O) | Alternate Ethernet Reset on Pin 9 |
| Eth-CS/IO10 | SJ10 | Pin 10 (PWM, digital I/O) | Alternate Ethernet CS Pin 10 |

**Tone Generator jumper** (SJ1) allows the tone generator to bypass AC coupling and gain to provide a TTL square wave output. The default supports sine/triangle tones and AC couples the signal and provides variable gain. If a TTL square wave output is needed, then cut the jumper and solder the pads for square wave output. This bypasses the AC coupling and variable gain.

**Auxiliary Input/Output** (X2) Provides additional I/O through a 20-pin ribbon cable. Two unused Teensy I/O pins are brought out, along with the upper ten Maxim I/O pins, and power.



The available pins are:

1. Digital ground
2. +5 volts
3. Analog ground
4. +3.3 volts
5. +12 volts
6. -12 volts
7. Teensy pin 28
8. Teensy pin 27
9. Maxim I/O pin 19
10. Digital ground
11. Maxim I/O pin 18
12. Maxim I/O pin 17
13. Maxim I/O pin 16
14. Maxim I/O pin 15
15. Maxim I/O pin 14
16. Maxim I/O pin 13
17. Maxim I/O pin 12
18. Maxim I/O pin 11
19. Maxim I/O pin 10
20. Digital ground

## Firmware

The firmware is written under the Arduino environment. Download and install the Arduino software and add in the Teensy support. See the Arduino and PJRC websites for current implementations.

Add the miniBCS-4 library to your Arduino environment. This is available on the jET Experimental Technologies GitHub repository. This provides high level access to the drivers, tone generator, and encoder.

If the MAX11300 is used, add the MAX11300 library, also on the jET Experimental Technologies GitHub repository.

If the LCD is used, you will need to add the following Adafruit LCD libraries:

XPT2046_Touchscreen for Touchscreen functions.
ILI9341_t3.h for TFT Display functions.

The Adafruit website has good tutorial on how to use their functions.

Finally, if it is desired to have a command interface, then it may be worthwhile to look at the CmdArduino library, also found on the jET Experimental Technologies GitHub repository.

## miniBCS-4 Library Functions

These functions are available through the miniBCS-4 Arduino library.

```
void begin();
        sets up hardware for I/O and interrupts
void driverOn(uint8_t ch);
        turn on solenoid driver channel 'ch'
void driverOff(uint8_t ch);
        turn off solenoid driver 'ch'
uint32_t getVersion(void);
        return firmware version (year, month, day concatenated into a single integer – ex: 20190230
void setToneGain(uint8_t level);
        set gain of tone output (0-255)
void toneReset(void);
        reset frequency generator chip
int8_t setTone( float freq, uint16_t mode);
        set tone to frequency 'freq' (1 to 20 MHz), and mode:
                TONE_SINE     sine wave
                TONE_TRIANGLE triangle wave
                TONE_SQUARE   square wave
void toneOn(void);
        enable tone output
void toneOff(void);
        disable tone output
int32_t getPosition(void);
        get encoder counts
void  setPosition(uint32_t newpos);
        set current value for encoder counts.
```

## Enclosure

An enclosure can be made from 1/8" laser cut acrylic. This allows the enclosure to be easily customized for the particular set of I/O connections mounted on the PCB. It also allows engraving unique labels for each connector.

A DXF file is provided that shows all the possible cutouts for various connector combinations.

## Minimal Assembly of Bare PCB

Only the parts needed for the application need to be assembled to the printed circuit board (PCB). At a minimum, the Teensy must be installed; everything else is optional. However, it is probably better to buy boards with all surface mount parts mounted and then add any required connectors and jumpers.

To decide what parts to install, first list the IOs needed for the application, then determine which IOs on the miniBCS-4 can fill the requirement. If only a few basic 0-3.3V digital and analog IOs are needed, the MAX11300 IC can be eliminated along with the DC-DC converter, power USB connector, and regulators. The parts that need to be added when the MAX11300 functions are needed are outlined on the PCB. If no solenoids, relays, or other power actuators are needed, the parts outlined in the solenoid driver area can be eliminated. If RS232 serial is not needed, then do not install the parts outlined in the RS232 area and change the jumper defaults for SJ1 and SJ2 by cutting the TX2 and RX2 track to the center pad and soldering from IO7 and IO8 to the center pad. This allows these pins to be used for additional IO.

Each IO will require a connector and in-line resistor (typically 100 ohms) or thermal cutout (PTC – such as the Murata PRG18BB series.

Minimal assembly:
Solder the headers for the Teensy

If any other ICs will be mounted, then D1 and D2 need to be installed.

If the MAX11300 functions are needed,
Add in the DC-DC converter, regulators, and filter caps, and resistors in that area.
Solder in the components within that area
The connectors for each port pair can be:
1 Molex 0855135016 right angle RJ9
1 Amphenol 112705 right angle dual BNC

Both can coexist on the circuit board, but only one or the other should be connected to external circuits.

If the encoder is needed,

Solder in a TE 103635-4 right angle header

Mount U2. If the index pin is needed, then also add U8.

If the LCD is required

Solder a header in at LCD1. This can be a pluggable header or a flex cable to allow the LCD to be positioned elsewhere. A flex cable such as A9AAT-090xF can be used, where 'x' is the length in inches, 2, 3, 4, 5, 6, or 8.

If the solenoid driver is required

Solder the components in the Solenoid Parts area.

If RS232 serial ports are required

Solder in the parts in the RS232 Parts area

Solder in, for each port:

1 Norcomp 190-009-163R001 right angle DB9

If serial port 2 is not need, the pins can be brought out as additional I/O by cutting the serial jumpers SJ1 and SJ2 and solder jumper the alternate configuration pads.

If the tone generator is required

Solder in the parts in the TONE area. Also, if a TTL square wave is required, instead of the sine/triangle output cut the jumper at SJ3 and solder the alternative configuration.