D.A.R.A Sensor Board - Small (SBS) 1.0

Digital multiplexer for analog and digital sensors
Datasheet Revision Date: 20220820

This product was made in cooperation with:
D.A.R.A. http://www.dararobots.com/
JWSmythe https://jwsmythe.com/
Makerspace Pinellas https://makerspacepinellas.com/

Features

- 16 pins - 2 banks of 8 sensor pins - bidirectional analog or digital on all ports.
- Common pullup/pulldown pin connections for each sensor pin
- Selectable pullup or pulldown per bank, or user defined electrical source
- Focus on minimum size for the application
- Smallest 16 port pluggable analog sensor board at 25.4mm x 25.4mm
- Will work on Arduino, Espressif, Raspberry Pi, and similar boards

Input Selection

There are 4 selection pins, S0, S1, S2, and S3. By selecting the appropriate combination of pins, your analog pin will have access to that pin's channel.

The TI 74HC4067 datasheet shows pins in MSB ordering. Many people learned to count in binary with LSB ordering. We're including both charts, but they convey identical information, the rows are simply rearranged for ease of reading.

MSB Ordering

Channel	S0	S1	S2	S3	E
None	X	X	X	X	**1**
0	0	0	0	0	0
1	**1**	0	0	0	0
2	0	**1**	0	0	0
3	**1**	**1**	0	0	0
4	0	0	**1**	0	0
5	**1**	0	**1**	0	0
6	0	**1**	**1**	0	0
7	**1**	**1**	**1**	0	0
8	0	0	**0**	**1**	0
9	**1**	0	0	**1**	0
10	0	**1**	0	**1**	0
11	**1**	**1**	0	**1**	0
12	0	0	**1**	**1**	0
13	**1**	0	**1**	**1**	0
14	0	**1**	**1**	**1**	0
15	**1**	**1**	**1**	**1**	0

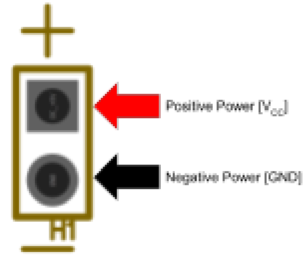
0=low **1**=high **X**= doesn't matter

LSB Ordering

Channel	S3	S2	S1	S0	E
None	X	X	X	X	**1**
0	0	0	0	0	0
1	0	0	0	**1**	0
2	0	0	**1**	0	0
3	0	0	**1**	**1**	0
4	0	**1**	0	0	0
5	0	**1**	0	**1**	0
6	0	**1**	**1**	0	0
7	0	**1**	**1**	**1**	0
8	**1**	**0**	0	0	0
9	**1**	0	0	**1**	0
10	**1**	0	**1**	0	0
11	**1**	0	**1**	**1**	0
12	**1**	**1**	0	0	0
13	**1**	**1**	0	**1**	0
14	**1**	**1**	**1**	0	0
15	**1**	**1**	**1**	**1**	0

0=low **1**=high **X**= doesn't matter

Power Connections (V_{cc})



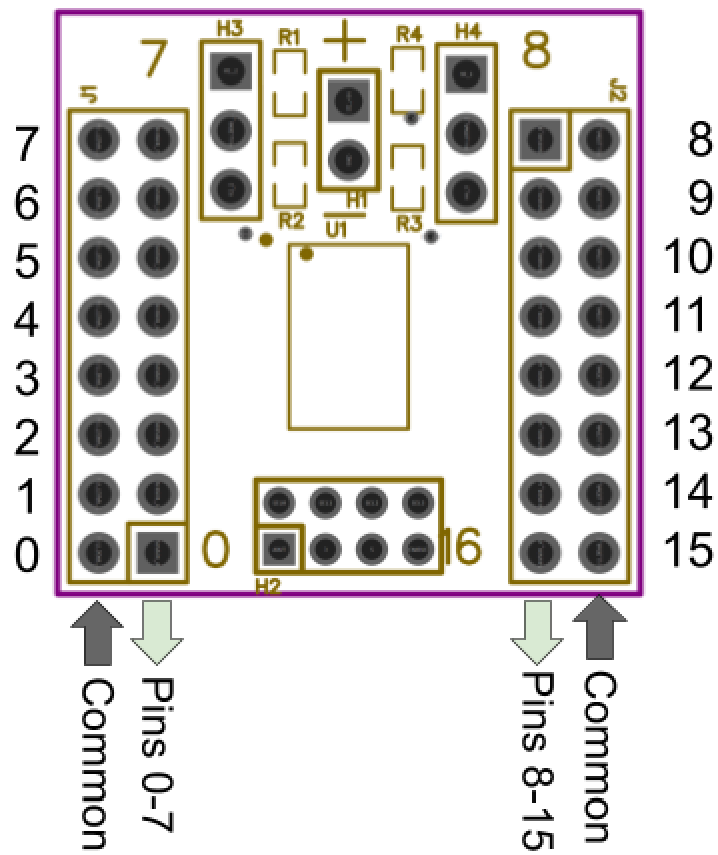
Attach **positive** power (V_{CC}) to the top pin of H1.

Attach **negative** (common/ground) from your microcontroller to the bottom pin of H1.

This Vcc provides power to the multiplexer chip, as well as the common options, as selected. See the section **Common Side Options** for more information about the common side options.

If you forget to attach the negative wire correctly, or it isn't common with your microcontroller, you won't get any useful data. Double check this if you have communications errors.

Sensor Connections



Attach your sensors to the innermost column of each 8 pin connector.

Pins 0-7 are numbered from the bottom on J1.

Pins 8-15 are numbered from the top on J2.

> **Note**: The "16" mark is erroneous,it should say "15". It is not present in SBS v1.1

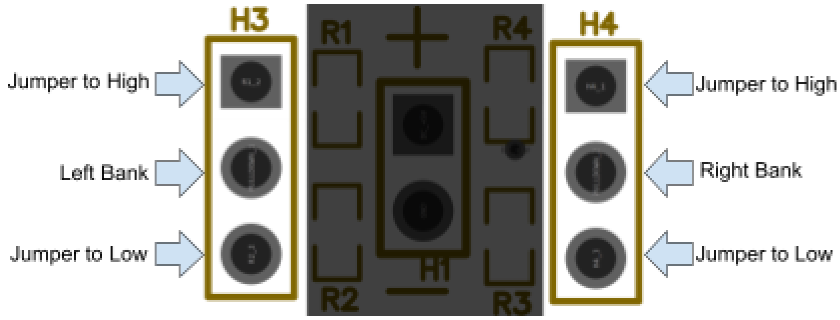
Your **Common** pin, typically +5V or GND, are attached to the outer pins. They are all connected together on each bank, but the banks are independent.

All of the left bank pins lead to the center pin of H3

All of the right bank pins lead to the center pin of H4

It is set up so you can use both pins, for your 2 pin sensors, without needing to find another source for your sensor power/ground pin. See the images in the "Sample Code and Wiring" section below. You aren't required to use these common pins, but it should help keep your wiring clean.

Common Side Options



The header blocks H3 and H4 allow you to select what kind of connection you will have to the common rails on the Left and Right bank common pins.

If you have a specific power requirement, you can simply connect the **center pin** of block **H3** and **H4**, marked **Left Bank** and **Right Bank**, to that power source.

The Left and Right Banks can have different configurations. They only impact the left or right side common pins respectively. The common pins are only there to simplify your sensor wiring. You can choose not to use them at all.

If you use a jumper to connect the center pin to the pin above or below it, it will get that voltage or ground for the V_{cc}. See the **Power Connects (V_{cc})** for more details.

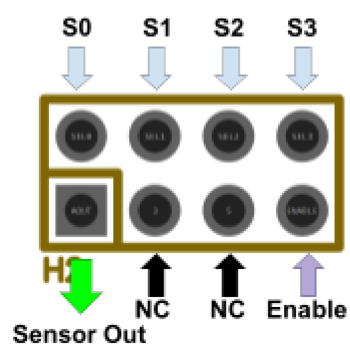
For example, one of our test boards, the ESP8266 chip, can only handle 1.0 vdc for analog sensors. Simply connect the Left and/or Right bank to a 1.0 vdc power source. Some ESP8266 dev boards already have a voltage divider on the analog ports, which can then support a range of 1.0 to 3.3 vdc. You'd simply jumper the Left Bank and/or Right Bank directly to the 3V3 pin, or another 3.3 vdc source.

You can use a common header jumper to select the left bank for pullup to V_{CC} or pulldown to GND.

The **High** pins goes through a 10kΩ resistor to your V_{CC}.
The **Low** pins go through a 10kΩ resistor to GND.

You can also attach your own wire to the center pin of H3 and/or H4, dictated by the kind of sensors you are using. If you have high power requirements for the sensors, use a direct wire to the center pin, and handle your pullup/pulldown externally.

Data Block



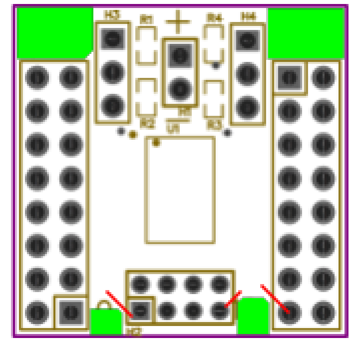
This is where you connect the sensor board to your microcontroller.

The data block, H2, has the select pins across the top (S0 to S3) which are the Selection pins referenced in the "Input Selection" section above. These let you choose which sensor port to read, and the output is available on the Sensor Out pin (bottom left).

The Enable pin is the bottom right.
Pins 3 and 5 are not connected, and are reserved for future use.

Important Note: On SBS v1.0, the pins in this block are **2mm pitch**, 2.8mm long, and 0.5mm thick. Common "Dupont" jumpers are 2.54mm pitch.

Drilling



Due to the small size of this board, we have not pre-drilled any holes. Most of the board is utilized, but there are still a few small places where you can safely drill. Those places are marked in green in this picture. Non-conductive double sided tape or plastic cable ties would be preferred if possible.

You should not only be sure the hole is in the green areas, but the head of the screws too. It may be wise to use one of the cardboard or plastic screw insulators under the head of the screws, so they won't possibly short or tear anything.

If you want to use the two smaller spots at the bottom, over the 0 and 15 (or 16 on v1.0), be VERY careful so you don't accidentally hit the traces marked in red. They can render the board completely unusable.

Usage

Using this board is fairly easy to do. There are numerous tutorials for the "**74HC4067**". Because of the acceptable power range, you can safely use this with Arduino boards (5v), Espressif boards (3.3v), Raspberry Pi (3.3v), and other similar microcontroller boards.   Just set the selector pins as high (1) or low (0) as appropriate, and read your analog port.

Setting the Enable pin LOW (GND) enables the board.
Setting the Enable pin HIGH (V_{CC}) disables all channels.

Sample Code and Wiring

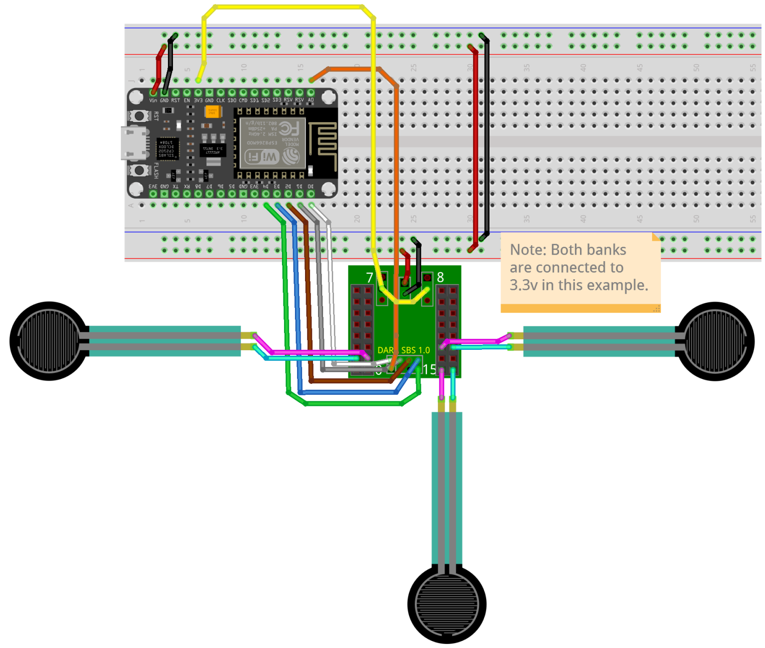
See the sample code at: https://github.com/JWSmythe/Hardware

We have tested on these platforms, with the code available at the repo above. All of these boards are cheap commodity development boards, not custom application boards. It should be enough to guide you to implementing it on other platforms. No special libraries are required.

Board	SoC/MCU	Select Pins				Enable	Sensor	Analog
		S0	S1	S2	S3		Pin	Voltage
ESP8266	AI Thinker ESP8266MOD	D0	D1	D2	D3	D4	A0[1]	3.3
ESP32	ESP-WROOM-32	32	33	25	26	27	35[2]	3.3
Arduino Nano/Uno	Atmel ATmega328P	D6	D5	D4	D3	D2	A0	5.0

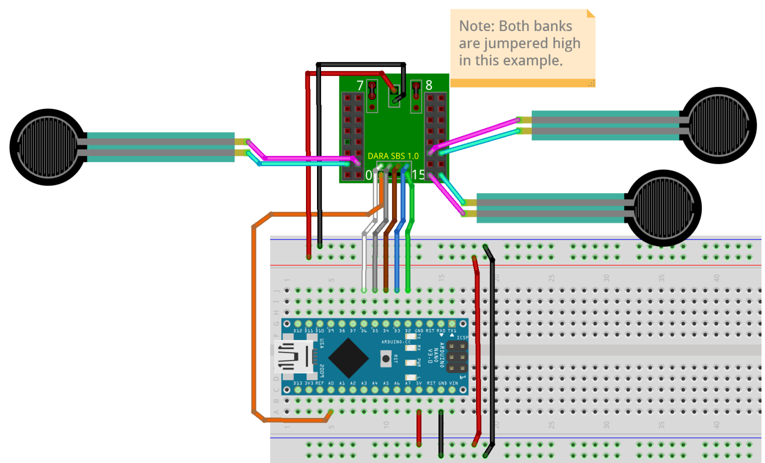
[1] A0 is the only analog input on the ESP8266.
[2] ESP32 has 18 12-bit analog pins. 15 of those are exposed on most dev boards.

ESP8266 has a good mid-range ADC resolution. It has a 0 to 3.3vdc voltage range. The ADC itself is 0 to 1.0vdc, but most dev modules have a built in voltage divider to support 0 to 3.3vdc. The top and bottom are flattened out. If you need high resolution across the entire range, use an external ADC. The range was more than adequate for testing, even without any smoothing or filtering code.

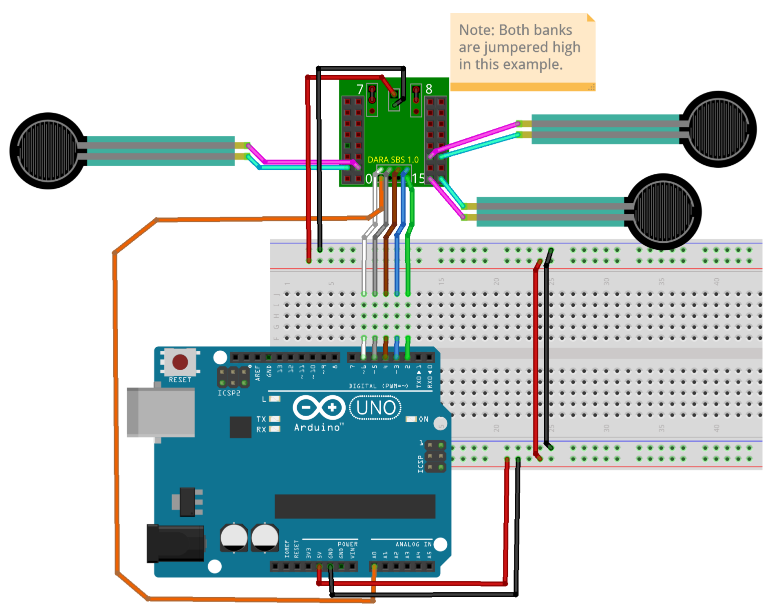


ESP32 as provided on dev modules by default has poor ADC filtering. Your signals will be heavily influenced by noise. This may be correctable with hardware noise filtering, or advanced settings when coding with the ESP-IDF framework. The necessary features are not available with the Arduino framework.

Arduino can use the same sample code as the ESP8266. You may also use the DDR and port register manipulation features, to control banks of pins with one shot. Search Google for "[arduino ddr port register manipulation]". The pins above are in reverse order from the ESP8266, simply because it allowed for cleaner wiring for the example schematic.



Raspberry Pi does not have an integrated ADC, so you cannot read analog signals directly. You can use it to read binary switches. Sample wiring diagram and code pending.



Operating Conditions

Parameter	Normal		Absolute Maximum		Notes
	Min	Max	Min	Max	
DC Supply (V_{CC})[3]	2	6	-0.5v	7v	Typical 3.3 to 5v microcontroller voltage
Sensor[4]	0	V_{CC}	0	V_{CC}	
Sensor Current	-20mA	+20mA	-25mA	+25mA	
Temperature (C)	-55°C	+125°C	-65°C	+150°C	See note 1
Temperature (F)	-67°F	+257°F	-85°F	+302°F	See note 1

Note 1: Despite the rated extreme temperatures, **no electronics like overheating**, and will **fail prematurely**. If you can't comfortably touch it, it's too hot. It is best to keep your electronics at room temperatures, and they will keep working for longer than you will. This applies to all electronics. Put the chips and boards in a cool environment, and run wires to your high temperature sensors.

More Modules, More Ports

You can further extend this by adding more modules. You'd simply use something to select which module to enable, and wire all the Select and Sensor Out pins in parallel.

Only have ONE module's Enable set to LOW (active) at a time. Multiple chips enabled (Enable LOW) at the same time would result in undesirable garbage received.

Included Important Components

Texas Instruments (**TI**) CD**74HC4067**SM96 [datasheet]
See the TI datasheet for further details on voltage and current limits.

[3] All voltages are in DC unless otherwise specified.
[4] Do not exceed V_{CC} limits. The analog/digital input port on your microcontrontroller will likely have stricter limits.

History

Board Version	Design	First Build	End of Life
SBS v1.0	20220423	20220509	TBD
SBS v1.1	TBD	TBD	TBD

The current version of this doc, sample code, and other JWSmythe projects are maintained at: https://github.com/JWSmythe/Hardware

Legal

This document contains information on a product under development. We reserve the right to change or discontinue this product without notice. The information included herein is believed to be accurate and reliable. However, we assume no responsibility for its use; nor for any infringements of patents or other rights of third parties which may result from its use. 3rd party components in this device may have their own additional legal restrictions and/or changes that may not have been disclosed. Please reference their datasheets for that information.

Copyrights and trademarks may be referenced in this document, and are copyright or trademarked by the applicable parties.

Addendums

v1.0

- **Right side Pins should be numbered 8-15. Corrected in v1.1.**

©Copyright 2022, JWSmythe and D.A.R.A. Robots