Intsy Hardware Module Assembly

Below are assembly instructions for the Intsy hardware module. Assembly requires only basic soldering skills. Anyone with modest experience can do it! Total assembly time is typically about 2-4 hours. The fully assembled module is shown in **Figure 1**.

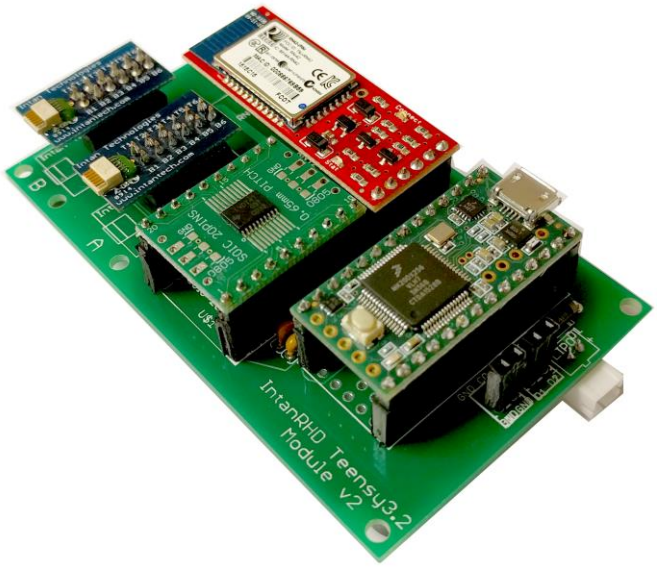


Figure 1. Assembled Intsy PCB. Battery connection and Intan RHD2132 amplifiers not shown.

Tools you'll need

- Soldering iron and solder + flux.
- Box knife of ultrasonic blade to cut rectangular headers to size
- Tweezers and/or needle nose pliers

Solder PCB headers and components

1. Solder power by-pass capacitors into board (yellow blobs in **Figure 2**). Be careful to properly orient electrolytic capacitors, if used. The positive side (longer leg) should connect pins 4 and 8 of the TI LVDT41 chip. These are easily identified as connecting to the thick, light green power rail.

2. Insert Schottky polarity protection diode as shown in **Figure 2**. The cathode (gray stripe) must be oriented toward the top right corner of the board.

Step 1b: Solder female rectangular headers on top side of PCB. Two-row headers may be convenient for the Intan RHD2132 adapter slots. Otherwise, use single row headers cut to the proper number of

contacts. Make sure they remain vertical during soldering; it is recommended to secure pieces with masking tape prior to soldering.

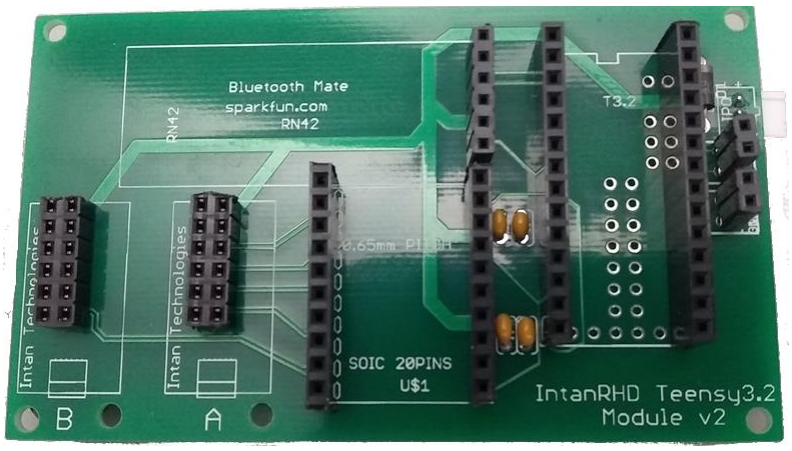


Figure 2. PCB top side

4. Flip the board over, solder and solder in the JST LiPo battery connector (**Figure 3**). The mechanical strength of the solder is weak, so it is recommended to add a drop of epoxy or hot glue to secure the element to the PCB.

5. Optionally add adhesive bumper feet to the corner of the board. Bumper feet reduce slipping. They also reduce the risk of conductive elements shorting out if Intsy is placed on a metal table or other conductive surface.

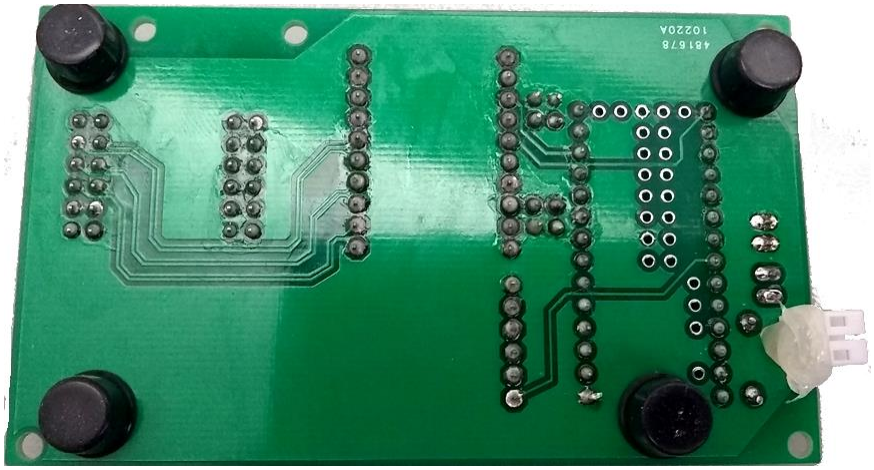


Figure 3. PCB bottom side.

Prepare commercial off-the-shelf components

1. Solder male header pins into the Schmartboard SOIC to DIP adapter. It is recommended to place the male pins in a standard breadboard, then the Schmartboard on top to ensure the pins remain vertical and flush to the board during soldering. Carefully place then surface mount solder the SN65LVDT41

LVDS-CMOS translator and tape it securely in place with masking tape of similar. Flux must be applied with a flux pen prior to soldering. The temperature of the soldering iron must be at least 800 F. Detailed instructions can be found here. The assembled LVDS component is shown in **Figure 4**.

Figure 4. Schmartboard SOIC-DIP 20 pin adapter with mounted LVDS-CMOS translator (SN65LVDT41).

2. Solder male 0.1" pitch pins into the RN-42 Bluetooth Silver Mate module, into the Teensy 3.2 microcontroller (**Figure 5**). The same technique of placing pins in the breadboard to ensure vertical and flush mounting to the component should be used.

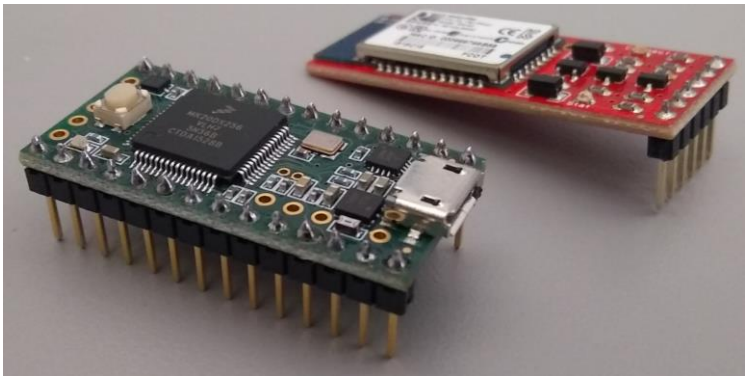


Figure 5. Teensy 3.2 and Sparkfun Bluetooth mate silver with male pins soldered in.

3. Solder dual-row male header pins into the Intan SPI cable adapter boards (see **Figure 1**, slots A and B)

4. Add the 100 MOhm external resistor to the RHD2132 board. The resistor should connect between pins labeled AUX3 and ground, as shown in **Figure 6**.

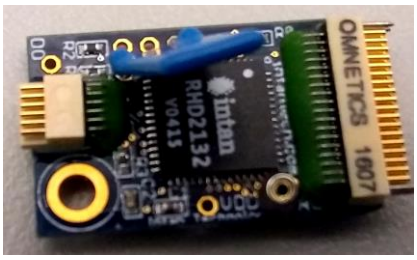


Figure 6. Intan RHD2132 ampliifer board, with external 100 MOhm resistor added (blue).

Mount components on PCB

1. Carefully plug in the Teensy 3.2, oriented as shown in **Figure 1**. The USB connector should face away from the "Intan RHD Teensy 3.2" text printed on the PCB.

2. Carefully plug in the Schmartboard SOIC-DIP adapter board. Make sure pin faces away from the RN-42 bluetooth module. The "SOIC 20PINS" text should be near the PCB edge.

3. Plug in the Intan SPI cable adapters as shown. The Omnetics connectors face outward.

4. Plug in the RN-42 Bluetooth module.

Connecting to amplifiers and host computer

1. It is now time to upload firmware to the Teensy 3.2. It is best to do this using a wired-USB connection (as opposed to over-the-air with Bluetooth). See the separate guide on programming the Teensy for specific instructions.

2. Decide if you are going to use wired or wireless mode. If using wireless mode, plug in a 3.7 V LiPo battery. If using wired mode, you can power the board directly from the USB port. However, it is strongly recommended to use a USB Isolator. See the separate guide on powering the Intsy module for further details.

3. Regardless of wired or wireless configuration, use a multi-meter to verify that 3.3 V is being delivered to all components. This is critically important for the Intan amplifiers which accept only 3.0 – 3.6 V. Anything higher, and you will instantly fry the Intan amplifiers!

4. Assuming power connections are verified to be ~ 3.3 V (absolutely no higher than 3.6 V), proceed…

5. Connect the Intan amplifier(s). These may be connected using a SPI cable from Intan. Alternatively, the Intan RHD2132 bioamplifiers may be directly connected to the board by 'flipping over' the amplifier to mate with the SPI cable adapter boards.

Congratulations - Your Intsy module is now fully assembled and ready to use! See, that was easy ☺

Intsy Power Supply

Created: Jon Erickson 11 May 2017
Modified: 07 Nov 2017

The Teensy 3.2 LDO voltage regulator LP38691 supplies "max 250 mA" continuous current. Dropout voltage is about 65 – 200 mV. In general, we can use the Teensy 3.2's on-board voltage regulator to distribute power to all other module components: Intan RHD2000 amps (note 3.6V hard max!); RN42 Bluetooth; LVDS-CMOS converter. The 3.3V regulated output on the Teensy is be distributed to these components. ***Note that you can NOT substitute Teensy 3.1, which has different power-supply circuitry.***

The power supply for the Insty system depends on the which configuration is implemented:

1) **Wired configuration**: Regulated input to V_USB, or unregulated input. We recommend using a modified Adafruit digital USB isolator + DC-DC converter which supplies regulated 3.3V at Micro-AB-1 USB cable connection. Using the USB isolator is recommended to improve data quality (remove potential ground loop) and for test subject safety/electrical isolation from the mains.
    a. The Adafruit must be modified for 3.3V operation, per instructions below.
2) **Wireless configuration**: 3.7 V LiPo battery connected to V_IN. This is also routed through Teensy's on-board LDO, so we get a 3.3 V output at the 3.3V pin.

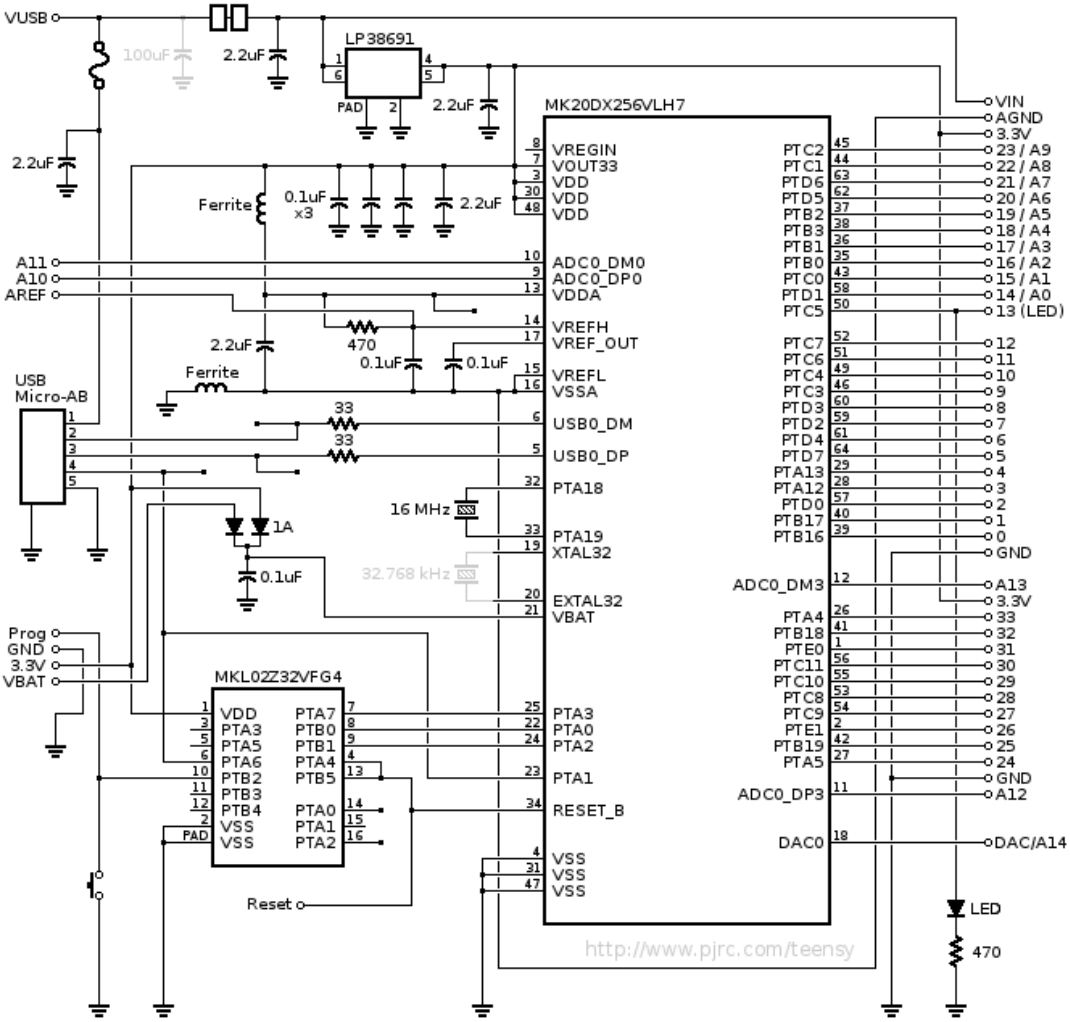
Intended usage is to **use ONLY ONE power source at a time**: either the USB cable OR the LiPO battery, but not both.

Modifying the Adafruit USB isolator

On the ADu4160 USB isolator IC, V_DD2 and V_Bus2 (pins 9 and 11) must be shorted to be compatible with 3.3 V logic. On the ADm5000 (dc-dc converter) V_SEL (pin 13) must be shorted to ground to output regulated 3.3 V. In order to do this, the leg of the ADm5000 IC must be lifted off of the PCB.

TEENSY 3.2 Schematic from PJRC website

Note the LP38691 LDO is driven by either VUSB or Vin. 3.3 V regulated output is available on multiple pins.



Intsy Firmware and Front-End Software Installation: How to connect to a host PC

Jon Erickson
Created: 27 Feb 2017
Updated: 07 Nov 2017

Installing Teensyduino

1. Install Teensy Loader: https://www.pjrc.com/teensy/first_use.html
2. Install Teensyduino from PJRC: https://www.pjrc.com/teensy/td_download.html

Notes: On Windows 10, the Teensy is typically recognized as a USB HID device on first use. Teensy must appear as COM device in order to interface properly with LabView. The remedy is easy, just 'verify' a build for it in the Arduino IDE, then the device changes from USB HID to COM port. See this this thread, especially post #13 in the PJRC forums: https://forum.pjrc.com/threads/28726-new-teensy-3-1-not-recognised-by-windows. Note the same "trick" works with the Teensy 3.2

Specifically, "Plug the fresh Teensy to a new USB port and wait for HID driver install, then [in Arduino IDE] hit program button after 'Verify' build. Then USB #40 showed up ready to go."

3. Verify that firmware for controlling the Intan chip has been uploaded to the Teensy 3.2. Note there are currently 2 firmware libraries to choose from depending on whether you want to use the Intsy as a 32 or 64 channel module. The firmware you upload must match the channel count. The latest firmware versions, respectively, are contained in `TeensyFirmware_32chan` or `TeensyFirmware_64chan`.
4. After selecting your version of firmware, open the main file, which is Intan_LV_interface_32chan.ino or Intan_LV_Interface_64chan_alpha.ino, for 32 or 64 channel modules, respectively.
5. Click the verify then the upload buttons in the Arduino IDE. Doing so should result in compiling and uploading the firmware the Teensy 3.2

Optional Bluetooth connection

1. Open Bluetooth Settings. Let Windows search for new device. The RN-42 module should be detected automatically (in this case as RNBT-9B59 or similar).
2. Identify the COM port number associated with the RN-42. In Windows 7, this is done by opening Bluetooth connections from the Control panel and right clicking to select Properties. In Windows 10, this can be done by scrolling to bottom of Bluetooth Devices > Click "More Bluetooth Options" > Click COM ports tab on Bluetooth Settings Windows that pops open. Look for the "Outgoing" direction COM port. Helpful post on this here.
3. If you are not using windows 10-- i.e., you are using Windows 7-- then you will need to use a USB Bluetooth dongle. (Prior to windows 10, no native Bluetooth API existed).
   Almost any legitimate model should be recognized by most windows 7 and installed. Once the COM port has been identified, it is necessary to program the RN-42 to work in MDM SPP mode. This allows CTS/RTS flow control, essential for stable connections streaming data at a high rate.
   a. Use CoolTerm or TeraTerm to establish a connection with the Bluetooth COM port.
   b. Type `X` to see all settings. The RN-42 should report back a range of settings. Check that the UART speed is set to 115k.
   c. Update the profile to MDM SPP by typing `S~,3`.

d. For more details see page 29 of Roving Network's manual: http://ww1.microchip.com/downloads/en/DeviceDoc/bluetooth_cr_UG-v1.0r.pdf And/or the sparkfun tutorial: https://learn.sparkfun.com/tutorials/using-the-bluesmirf?_ga=1.59342189.1999315961.1476995915

You are now ready to connect to the Intsy module!

Installing LabView library

1. Copy and extract LabView library folder to host machine. The most recent versions of 32 and 64 channel versions are currently in `LabView32chan` and `LabView64chan`.
2. Download a copy of NI's software circular buffer library: http://www.ni.com/tutorial/7188/en/
3. Install the software circular buffer module in *<root>/LVs/swcircularbuffer1.0.19/setup.exe*
4. Open *<root>/LVs/Intan_mainControlPanel.vi.* The first time loading on a new machine, this may take a minute, as labview is locating and compiling all of the modules.
5. Identify the Teensy COM port number in Windows Device Manager. Select it in under "VISA resource name" in *Intan_mainControlPanel.vi.* Select the corresponding COM port type (either USB serial or Bluetooth).
6. Configure all other hardware settings and save file settings.
    a. All registers must be configured upon Intan power up.
    b. Following the initial configuration, configuring reg 0-7 and 14-17 (Apwr) is optional, but recommended.
    c. Hardware filter settings and sampling frequency may be updated independently.
7. Click the white Run arrow. This configures all selected registers with the chosen settings
8. The purple "Start Data Stream" button will become enabled, once Intan register configuration is complete. Click the purple button to start the data streaming. Doing so should open a new window in LV (Intan_streamData_parallelLoop.vi) with various filter controls etc.
9. The user can update all the visualization settings in real-time: LPF, HPF, BPF, channel selection, Y-scale, etc.
10. Saving the file may be discontinued within this window, but it is strongly recommended NOT to do so.