Ag Irrigation-Pump Remote Control
Using Arduino-UNO and Digi-Xbee RF

Table of Contents

1. INTRODUCTION

The "***Ag Irrigation Pump Remote Control***" project's goal is to provide RF remote control and monitoring features of an Agriculture-Industry Irrigation Pump; typically used in wheel-line and hand-line irrigation setups. The project uses Xbee RF radio's with up to a 28-mile range (More practically a mile or two at ground level). The project consists of a Hand-Held "***Hand Remote***" and a pole mounted "***Pump Controller***" as well as high voltage control electrical components to be mounted inside the pumps electrical panel.

The project is Open Source and Resides at: https://github.com/tgit23/AgIrrigationRemoteControl
The most recent version of this document can be gotten as a PDF at: https://github.com/tgit23/AgIrrigationRemoteControl/AgIrrigationPumpRemoteControl.pdf

Illustration 1: HandRemote   *Illustration 2: PumpController*

2. INSTALLATION

Most 3-phase irrigation pumps that do not have variable frequency drives(VFD) or inverters are setup similarly. Below is the details of two pump panels as an ***example or reference*** to the pumps power circuit. This information was gathered to determine the wiring and power needed to control the pumps power.

2.1 Verify Your Current Setup

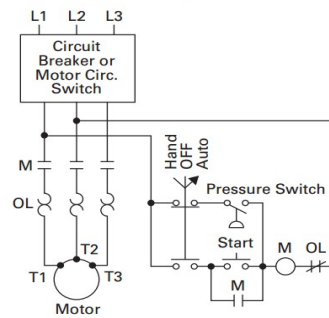### 2.1.1 Example Ditch Pump

✔ Details

- ○ Box   - Eaton Irrigation Pump Control Type 3R Enclosure Rainproof
  - ▪ http://www.eaton.com/Eaton/ProductsServices/Electrical/ProductsandServices/AutomationandControl/EnclosedControl/NEMA/PumpPanels/index.htm

Type 3R Industrial Pump Panel—C440 SSOL

Wiring Diagram

Typical Wiring Diagram

○ Wiring Diagram Description
  ▪ Labels
    ▫ L1, L2, L3   - Are the three HOT wires of the 3-Phase Power
    ▫ M   - The Starter Contactor (see Panel picture; white block with a blue center) contains:
        ◦ Activation Coil   - Shown as a circle in the wiring diagram just right of the Start Button (Activates the "Hammer")
        ◦ Main Contacts   - The Contacts that connect the L1,L2,L3 lines → OL → Motor (Sometimes called the "Hammer")
        ◦ Hold Contact   - The Contact wired in parallel with the Start Button and mounted on the Right-Side of the Starter Contactor Block

  When **HAND** is selected the "control circuit" (L1 → L2 loop) is still open until the Start Button is pressed. When the Start button is pressed it
  activates the (M) Coil which in-turn activates the "Hold Contact" (the --|/-- in parallel to the Start Button). This right-side contactor holds the
  "control circuit" closed until there is a power failure bump or the Hand-Off-Auto Switch breaks open the circuit loading the (M) activate coil.

  This Projects SSR-Relay will be connected where the "Pressure Switch" is shown in this wiring diagram and will use the **AUTO** switch selection.
  This setup will require that the Arduino **NOT re-activate** power after a power failure has occurred until a specified time delay or turned back on.
  The click in HAND circuit is setup so the power company can re-activate power after a power failure without the over-load of all motors kicking
  on at the same time.

○ Starter Contactor   - Eaton-Cuttler/Hammer 45-Amp Model # AN16GNO Series B1   (Need to check Contactor Model #)
  ▪ MODEL # Decoding
    ▫ A = Starter
    ▫ N = NEMA
    ▫ 1 = Non-Reversing
    ▫ 6 = Starter w/C306 Over Load Relay
    ▫ G = NEMA Size #2 (45-Amp)
    ▫ N = Normally Open
    ▫ O = Horizontal Mount
  ▪ https://www.galco.com/buy/Cutler-Hammer-Div-of-Eaton-Corp/AN16GN0AB
  ▪ http://www.eaton.com/Eaton/index.htm
  ▪ http://www.eaton.com/Eaton/ProductsServices/Electrical/ProductsandServices/AutomationandControl/ContactorsStarters/ElectromechanicalContactorsandStarters/FreedomSeries/F
        reedomSeries/index.htm#tabs-2

○ Starter Contactor (Coil) Amperage
  ▪ See Page 9 "AC COIL DATA" for NEMA Size #2 Shows 230VA(Volt-amperes which is VA=V*A) so 230VA/480Vac = 0.479Amps @ 480V
  ▪ http://www.eaton.com/ecm/idcplg?
        IdcService=GET_FILE&allowInterrupt=1&RevisionSelectionMethod=LatestReleased&noSaveAs=0&Rendition=Primary&dDocName=998056282226
○ Auxiliary Contact   - Cutler Hammer Model #C320KGS1 10-Series A2 (480VAC @ 1.5 Amps?)

2.1.2 Example Canal Pump

✔ Details
  ○ Box by Cutler-Hammer
  ○ Need to get more details later after opening the enclosure
  ○ Capacitor = Model #9L18BBB301 650VAC 1-1-1 MFD

2.2 Apply Power Control

BE SURE THE MAIN BREAKER IS TURNED OFF AND CAREFULLY CHECK THAT THERE IS NO VOLTAGE ON THE BREAKER WITH A PROPER METER!!!
480VAC is deadly; and can cause severe flash burn (e.g. A meter on the wrong setting can "flash" explode causing "flash" burns on you even at a distance away)
See https://www.youtube.com/watch?v=6hpE5LYj-CY

2.2.1 Install Power Supply

1. Cut DIN Rail to length that will comfortably fit the Power Supply, Solid State Relay, and Fulree DC/DC Converter
2. Using tapped screws fasten the DIN rail horizontally in an open space within the Pump Panel
3. Attache the RHINO power supply to the DIN rail
4. Wire L1, L2, L3 of the RHINO Power Supply to the Top Terminals of the Starter Contactor

2.2.2 Install Solid State Relay

1. Attach the Solid State Relay next to the Power Supply on the DIN Rail
2. Wire Connector 15 → Auto Position of the HAND-OFF-AUTO switch
3. Wire Connector 16 → M auxiliary contact on the right side of the Starter Contactor

2.2.3 Wire in Pump Controller

1. Mount the 'Pump Controller' Box up high on the telephone pole or on-top of the Pump Panel

2. Wire the Solid State Relay (A1) Connector to the Pump Controller D5 Pin
3. Wire the Solid State Relay (A2) Connector to the GND on the Pump Controller
4. Wire the 24Vdc(+) from the Power Supply to the Pump Controller
5. Wire the 24Vdc(-) from the Power Supply to the Pump Controller

3. OPERATING INSTRUCTIONS

3.1 Pump Control

3.2 Alarms

3.3 Customizing

4. FIRMWARE & DEBUG

✔
✔

4.1 Uploading Firmware

1. Install Arduino Sketch IDE *(If NOT installed already)* https://www.arduino.cc/en/Main/Software

2. Install the PeerIOSerialControl Library *(If NOT installed already)*
    a) In a web-browser; go to https://github.com/tgit23/PeerIOSerialControl
    b) On Green Clone or download Choose Download Zip
    c) Save to a place you'll remember – like Documents or Desktop
    d) Run the Arduino Sketch IDE
    e) Choose Menu Item Sketch → Include Library → Add .zip Library
    f) Select the PeerIOSerialControl Library file saved in step 'b' above

3. Download the Firmware
    a) In a web-browser; go to https://github.com/tgit23/AgIrrigationRemoteControl
    b) On Green Clone or download Choose Download Zip
    c) Save to a place you'll remember – like Documents or Desktop
    d) Unzip the Folder
    e) In Arduino Sketch IDE choose File → Open and Select
        • File HandRemote.ino for the HandRemote Firmware
          ◦ Location Example: \Downloads\AgIrrigationRemoteControl-master\AgIrrigationRemoteControl-master\HandRemote
        • File PumpController.ino for the PumpController Firmware
          ◦ Location Example: \Downloads\AgIrrigationRemoteControl-master\AgIrrigationRemoteControl-master\PumpController\PumpRemote

4. Compile and Upload the Firmware
    a) Setup the unit to "**Upload-Mode**" by removing both Zigbee Shield jumpers
    b) Plug in a USB cable from the Computer to the unit
    c) Select the Port the units USB is connected to; in Sketch menu Tools → Port
        • To determine Port; Open Windows Device Manager → Ports (COM & LPT) a new COM?? port appears right after plugging in the cable
    d) Select the Board; Sketch menu Tools → Board → Arduino/Genuino UNO
    e) In Arduino Sketch IDE; Press the Right-Arrow next to the Check mark in the Top-Left Corner to upload the firmware onto the unit

4.2 Customizing Firmware

✔ Overview
  ○ Downloaded firmware includes menu items for;
    ▪ Pump Controller Selection   - Ability to control multiple Pump Controllers from one Hand-Remote Unit
    ▪ Power   - Sets or Read the Pump Power as either ON or OFF
    ▪ Water Level   - An Accessory IO-Monitor to check the canal/ditch water level
    ▪ Battery   - Reads the Battery Voltage Level
    ▪ Pressure   - An Accessory IO-Monitor to check the water pressure
    ▪ Alarms   - Various Alarms for the monitoring readings above

The menu items are defined in the Setup-Menu Function identified by line 'void SetupMenu() {'. Each item in the Menu has a numeric index (i.e. Menu[index-#]).

✔ To delete a Menu Item
  ○ Find the menu items index (ie. Menu[index]) you'd like to delete by identifying it by its' Menu[Index].Text setting
  ○ Select ALL Menu[index] lines with the same index
  ○ Press delete
  ○ Starting from the top make sure the indexes are in order; for example if you deleted [3], change [4] to [3], [5] to [4] and etc.. for all entries
  ○ Up towards the top of the program (~line 50) change value of '#define MENUITEMS ?' to the number of menu items that exist now

✔ To add a Menu Item
  ○ Start a new line below the last one defined and enter the attributes that need to be set
    ▪ Menu[?].Text   - The text that will display on the LCD
    ▪ Menu[?].Value   - Typically not set; holds the value of the item
    ▪ Menu[?].ValueSettable   - TRUE/FALSE determines if the user can change the value
    ▪ Menu[?].ValueLocation   - PROG/LOCALPIN/REMOTEPIN/EPROM; Determines where the value of the item is set/retrieved
    ▪ Menu[?].Pin   - Arduino Pin# associated with the item

- ▪ Menu[?].Analog   - TRUE/FALSE; Default is FALSE, Set to 'true' if using an Analog pin for the value
- ▪ Menu[?].AlarmCompare   - LESS/GREATER/EQUAL; Defines how an alarm will compare its limit value to the ValueIdx Value
- ▪ Menu[?].AlarmValueIdx   - The Menu[Index].Value this alarm item will compare itself with
- ▪ Menu[?].Option[?].Poll   - TRUE/FALSE; Default is true. Set to false if the value doesn't need to be refreshed and displayed

Options are how a limited amount of select-able items are defined for a value. For example ON or OFF would be two options.

- ▪ Menu[?].Option[?].LastOption   - Counting from 0-> ; How many options are defined
- ▪ Menu[?].Option[?].Text   - Option[?].Text defines an option that can be selected like 'ON'
- ▪ Menu[?].Option[?].DrivePin   - Option[?].DrivePin LOW or HIGH when this option gets selected

○ Up towards the top of the program (~line 50) change value of '#define MENUITEMS ?' to the number of menu items that exist now

5. OPTIONAL ACCESSORIES

5.1 Pressure Transducer ($9.59)

✔ 5V 0-1.2 MPa Pressure Transducer Sensor Oil Fuel Diesel Gas Water Air Sensor
   ○ http://www.banggood.com/Pressure-Transducer-Sensor-Oil-Fuel-Diesel-Gas-Water-Air-Sensor-p-1007341.html   - $9.59

5.2 Ultrasonic Water Level ($10.77)

✔ DC 5V Waterproof Ultrasonic Module Distance Measuring Transducer Sensor
   ○ http://www.banggood.com/DC-5V-Waterproof-Ultrasonic-Module-Distance-Measuring-Transducer-Sensor-p-1094462.html   - $ 10.77

5.3 Water Meter

6. BUILDING INSTRUCTIONS

The following tools are required to build the project components

✔ 3D-Printer (*may be optional if other casing solutions are available*)
✔ Soldering Iron / Solder / Wire
✔ Screw Driver

6.1 MATERIALS (BOM) - $206.69

6.1.1 Pump-Panel Components ($95.74)

Illustration 3: (1) Power Supply

Illustration 4: (1) Solid State Relay

Illustration 5: (1) DIN Rail

○ (1) 12VDC Power Supply – Main power for each PUMPCONTROLLER
  ▪ Buying Options (Recommended; *the RHINO & Fulree Converter for practical usage and pricing*)
    ▫ RHINO switching Power supply, 24 VDC output, 2.5A, 60W, 320-600 VAC input, 3-phase, 35mm DIN rail mount.
      ▫ https://www.automationdirect.com/adc/Shopping/Catalog/Power_Products_(Electrical)/DC_Power_Supplies/24VDC,_3-Phase_Input/PSB24-060S-3 ($59.00/EA)
      ▫ RHINO Requires; (24 → 12VDC) Fulree Converter (*see Electronic Control Section below*) ($7.99/EA)
    ▫ TDK Lamda Rail Power Supplies 120W 12V 10A DIN Rail 380-480VAC
      ▫ www.mouser.com/ProductDetail/TDK-Lambda/DPP120-12-3/ ($104.04/EA)
    ▫ Car Battery
      ▫ If seriously concerned about money; A used/charged car battery can also be used to supply power to the PUMPCONTROLLER for a while

○ (1) Solid state relay, 35mm DIN-rail, 3-32 VDC input, SPST, normally open SCR, 10A contact rating, 48-480 VAC load voltage – for each Pump
  ▪ https://www.automationdirect.com/adc/Shopping/Catalog/Relays_-z-_Timers/Solid_State_Relays/DIN_Mount_Relays_(AD-SSR8_-z-_AD-SSR2_-z-_AD-SSR6_-z-_AD-HSSR8_Series)/General_Purpose_10A_-_65A,_(AD-SSR8_-z-_AD-SSR2_-z-_AD-SSR6_Series)/AD-SSR810-DC-48Z ($20.75/EA)

○ (1) 6" of DIN rail, slotted, 35mm, 10 mm height – for mounting Power Supply and Solid State Relay of each Pump
  ▪ https://www.automationdirect.com/adc/Shopping/Catalog/Enclosures_-z-_Subpanels_-z-_Thermal_Management_-z-_Lighting/Enclosure_Subpanels_-a-_Internal_Mounting_Accessories/DIN_Rails/DN-R35SAL1-2 ($8.00/2x 1-Meter Pieces)
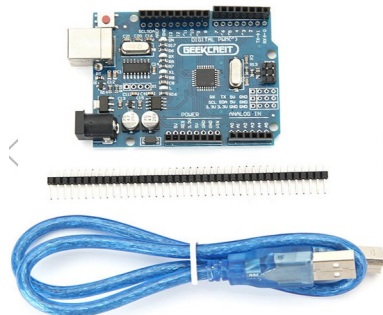
6.1.2 Electronic Control ($30.07)

Illustration 6: (2) Arduino UNO R3

Illustration 7: (1) LCD Keypad Shield

Illustration 8: (2) Zigbee Shield

○ (2) UNO R3 Atmega328P Development Board with Male Pin Header and USB Cable - (1) for each HANDREMOTE and (1) for each PUMPCONTROLLER
  ▪ https://www.aliexpress.com/item/Free-shipping-high-quality-UNO-R3-MEGA328P-CH340-CH340G-for-Arduino-UNO-R3-USB-CABLE/32353698284.html ($3.80/ea)

○ (1) Keypad Shield Blue Backlight for Arduino Robot LCD 1602 Board – Display for each HANDREMOTE
  ▪ https://www.aliexpress.com/item/Free-Shipping1PCS-LCD-Keypad-Shield-LCD1602-LCD-1602-Module-Display-for-arduino-ATMEGA328-ATMEGA2560-raspberry-NO/32787414048.html
    ($2.19 / ea)
  ▪ DFRobot Wiki LCD @ https://www.dfrobot.com/wiki/index.php/Arduino_LCD_KeyPad_Shield_(SKU:_DFR0009)
  ▪ Schematic @ http://image.dfrobot.com/image/data/DFR0009/LCDKeypad%20Shield%20V1.0%20SCH.pdf
  ▪ Layout @ https://raw.githubusercontent.com/Arduinolibrary/DFRobot_LCD_keypad/master/DFR0009_Layout.png

○ (2) Zigbee Shield RF Wireless Module Expansion Board for Arduino Xbee - (1) for each HANDREMOTE and (1) for each PUMPCONTROLLER
  ▪ http://www.banggood.com/Zigbee-Shield-RF-Wireless-Module-Expansion-Board-For-Arduino-XBee-p-916226.htm l ($4.83/ea)
  ▪ Schematic @ https://www.arduino.cc/en/uploads/Main/XbeeShieldSchematic.pdf
  ▪ Tutorial @ https://www.cooking-hacks.com/documentation/tutorials/xbee-arduino-raspberry-pi-tutorial/

Illustration 9: (1) 9V Backup Power Case   *Illustration 10: (15)5.08mm Screw Terminal* *Illustration 11: (1) DC-to-DC Converter*

○ (1) DIY 9V Battery Storage Container Box Case Holder With ON/OFF Toggle Switch - Emergency backup power for each PUMPCONTROLLER
  ▪ http://us.banggood.com/Wholesale-Warehouse-DIY-9V-Battery-Storage-Container-Box-Case-Holder-With-ONOFF-Toggle-Switch-wp-Usa-1086131.html ($4.66/EA)

○ (15) 2x Pin Plug-in Screw Terminal Block Connector 5.08mm Pitch - External I/O connections for each PUMPCONTROLLER
  ▪ http://www.banggood.com/20pcs-2-Pin-Plug-in-Screw-Terminal-Block-Connector-5_08mm-Pitch-p-993197.html ($2.15 / 20pcs)

○ (1) Fulree DC 8-35V to 1.5-24V Adjustable Buck Converter Power Supply Voltage Regulator – For the RHINO Power Supply purchase ONLY
  ▪ http://www.banggood.com/DC-8-35V-to-1_5-24V-Adjustable-Buck-Converter-Power-Supply-Voltage-Regulator-For-Car-Truck-Boat-p-1087176.html (Price listed with Pump Panel)
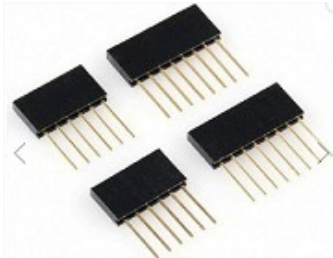

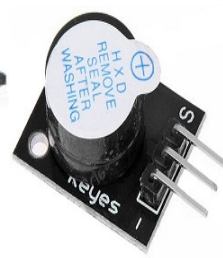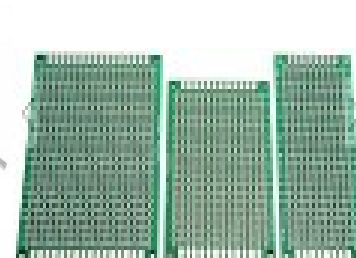*Illustration 12: Female Pin Headers*   *Illustration 13: Rocker Switch* *Illustration 14: Active Buzzer* *Illustration 15: 50 / 30cm x 70cm Perfboard*

○ (4) Pin Headers 2.54mm pitch - For each PUMPCONTROLLER
  ▪ (1) 4-PIN Male   - Zigbee->Perfboard POWER ($NA / Included with Arduino Kit - Male headers can be cut to length)
  ▪ (2) 6-PIN Male   - Zigbee->Perfboard (1) for Analog (1) for Digital ($NA)
  ▪ (1) 4-PIN Female   - Zigbee->Perfboard RX/TX
    ◦ https://www.aliexpress.com/item/10-pcs-4P-4pin-Female-Single-Row-Straight-Header-Strip-Socket-Connector-Pitch-2-54mm/32442087495.html ($0.79 / 10pcs)

○ (1) Black Snap-in On/Off Rocker Switch – For each HANDREMOTE
  ▪ https://www.aliexpress.com/item/New-Black-Push-Button-Mini-Switch-6A-10A-110V-250V-KCD1-2Pin-Snap-in-On-Off/32670615106.html ($0.86 / 5pcs)

○ (1) Active Speaker Buzzer Alarm – For each HANDREMOTE
  ▪ https://www.aliexpress.com/item/KEYES-KY-012-Active-Buzzer-Module-FOR-The-ARDUINO-AVR-PIC/32740686896.html ($0.60 / ea)

○ (2) FR-4 Double Side Prototype PCB Printed Circuit Board - (1) 30cm x 70cm and (1) 50cm x 70cm for each PUMPCONTROLLER
  ▪ https://www.aliexpress.com/item/Free-Shipping-4pcs-5x7-4x6-3x7-2x8-cm-double-Side-Copper-prototype-pcb-Universal-Board-for/32730636287.html ($1.56/ 4pcs)

6.1.3 RF Electronics ($80.88)

- ○ (2) Xbee-Pro 900Hp S3B Digimesh, 900Mhz, 250Mw, Rpsma Connector, 200Kbps – (1) for each HANDREMOTE and (1) for each PUMPCONTROLLER
  - ▪ https://www.arrow.com/en/products/xbp9b-dmst-002/digi-international ($36.11/EA)

- ○ (2) Antenna Helical 3dB Gain 900MHz – (1) for each HANDREMOTE and (1) for each PUMPCONTROLLER
  - ▪ https://www.arrow.com/en/products/w1063/pulse-electronics-corporation ($4.33/ea)

6.1.4 Other Components

- ○ (7) Small Screws no longer than 10mm or bigger than 3.5mm for Display and Case covers
- ○ (4) Resistors; Half valued at 1/3 the other - Recommend above 10K-Ohms – (2) for each HANDREMOTE and (2) for each PUMPCONTROLLER
- ○ (1) Right-Angle 9Vdc Power Plug for Arduino Power Jack
- ○ (2) Diodes

6.2 HAND REMOTE

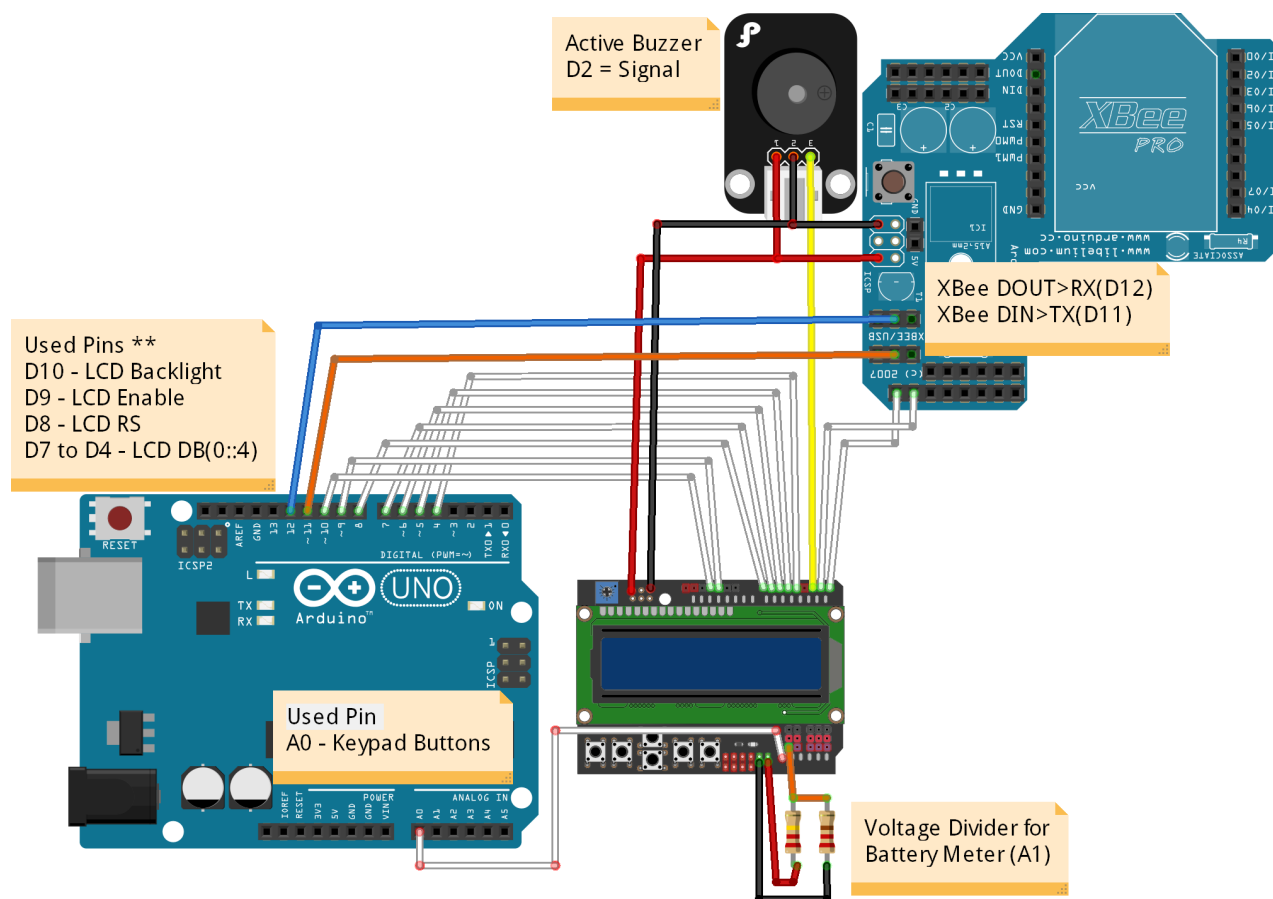The HANDREMOTE is the remote controller unit used to control and monitor the pump remotely.

✔ Features
- ○ Buzzer alarm settings for monitored values
- ○ Battery level indicator (9V-Battery Powered)
- ○ Programmable Interface through USB

6.2.1 Electronics Assembly

✔ The HANDREMOTE uses the following electronic parts
- ○ 1- Arduino UNO R3 Board
- ○ 1- LCD Keypad Shield
- ○ 1- Zigbee Shield
- ○ 1- Xbee-Pro 900Hp S3B Module and High Gain Helical Antenna
- ○ 1- 9Vdc Battery Connector
- ○ 1- Active Buzzer
- ○ 2- Resistors

Active Buzzer
D2 = Signal

XBee DOUT>RX(D12)
XBee DIN>TX(D11)

Used Pins **
D10 - LCD Backlight
D9 - LCD Enable
D8 - LCD RS
D7 to D4 - LCD DB(0::4)

Used Pin
A0 - Keypad Buttons

Voltage Divider for
Battery Meter (A1)

fritzing

A. LCD Keypad Shield

    1. Wire the Battery-Meter Voltage-Divider
        a) Select (2) Resistors; One-Value being 1/3 the other (10K-ohm up to 1M-ohm is recommended)
        b) Solder First Resistor (Full Value) on the top of the LCD Shield from **VIN → A1**
        c) Solder Other Resistor (1/3 Value) from **A1 → GND** (Solder hole just left of VIN)
   d) Snip off the excess back-side leads

The two resistors setup a voltage divider from the 9Vdc battery (which we will attach to VIN). Analog inputs on the Arduino should never exceed 5Vdc. The voltage divider will split the battery voltage to 1/3 its actual voltage. This will allow the hand-remote to monitor the battery level. The larger the resistor values the less drain on the battery; but too-large a value resistors may affect accuracy.
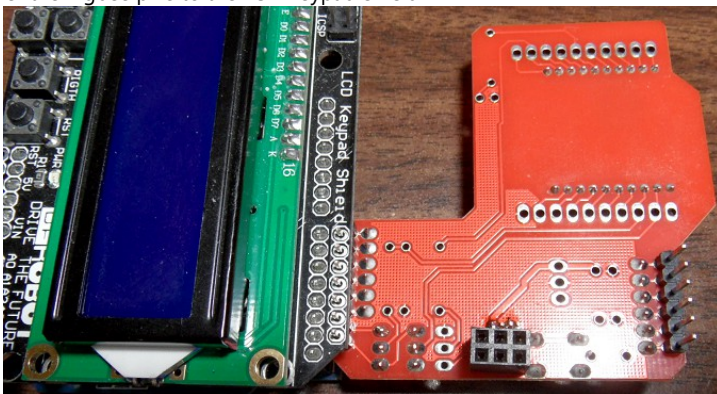
    2. Plug the LCD Keypad Shield onto the Arduino UNO

B. Zigbee/Xbee Shield

    1. Remove Header Pin **D7** From Zigbee Shield
        a) Using Needle Nose pliers clamp onto Pin D7 of the back of the Zigbee Shield
        b) Using Soldering Iron – Heat the top of the pins solder pad and pull the pin from the board
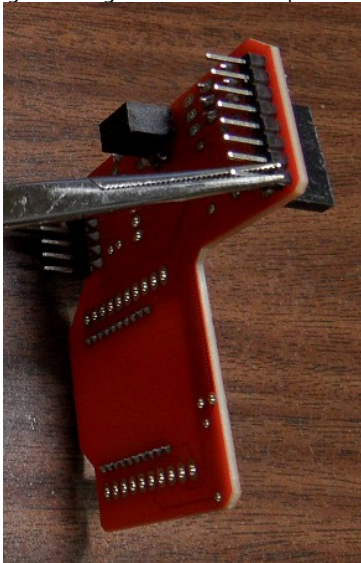
    2. Solder Zigbee Shield to the LCD Keypad (Flipped upside down)
        a) Insert pins D0 → D6 (D7 removed in Step #1) into the Top Solder holes of the LCD Keypad Shield
        b) Position the Zigbee Shield flat with LCD Shield and Up against the LCD Shield – Requires bending the header pins a little (**with UNO Attached**)
        c) Solder the Zigbee pins to the LCD Keypad Shield

✔ Further Reading for the Curious
- ▪ https://www.arduino.cc/en/Main/ArduinoXbeeShield
- ▪ https://www.arduino.cc/en/Guide/ArduinoXbeeShield

C. Active Buzzer

1. Solder Active Buzzer to LCD Keypad Shield

    a) Either print a small piece of plastic or add Black tape to the Back-Side of the Active Buzzer to prevent shorting against the Zigbee Shield.

    b) Solder the **'S' pin** on the Active Buzzer to **D2** (3rd Left Pin – LCD Face Up) on the LCD Shield

    c) Cut the (-) and center pins of the Active Buzzer leaving about 1/8" for connection but short enough not to touch D3 & D4.

2. Wire Power to the Active Buzzer and Zigbee Shield

    a) Bend the first and last pins closest to the edge of the LCD Keypad Shields ICSP pin header

    b) Solder one end of a **Red** wire → Left-Most Pin of the LCD Keypad Shields ICSP pin header (**5V+**)
        • Pull wire to **Active Buzzer (+)** Pin (Center Pin), strip insulation and solder
        • Pull wire to Lower-Right of the **Zigbee ICSP (5V+)** plug, strip insulation, cut to length and insert into plug

    c) Solder one end of a **Black** wire → Right-Most Pin of the LCD Keypad Shields ICSP pin header (**5V+**)
        • Pull wire to Active **Buzzer Left-Pin (-)**, strip insulation and solder
        • Pull wire to Top-Right of the **Zigbee ICSP (GND)** plug, strip insulation, cut to length and insert into plug
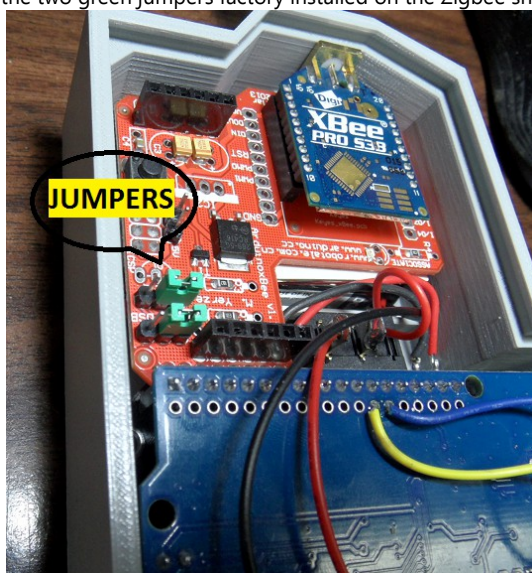
    Red/Black wires were not cut but the insulation was split using automatic wire strippers. The Center and Left Pins of the Buzzer were cut just long enough to solder wires to them.
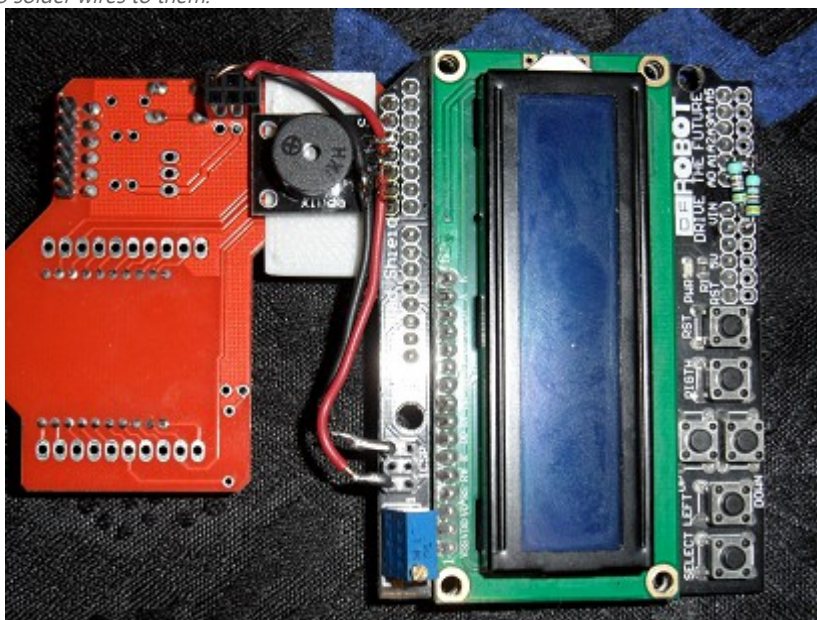
3. Cut the remaining ICSP pins as short as possible to allow fit inside case

D. Communications

1. Remove the two green jumpers factory installed on the Zigbee shield

2. Connect Arduino UNO pins D11 & D12 to the Zigbee Communication Header
    a) From an upside-down view; left-top of Arduino UNO; Count the pin holes to the right D0,D1,D2.... to D11
    b) Solder a wire from Arduino UNO pin D11 to the top-jumper-header center-pin of the Zigbee shield
    c) Solder a second wire from Arduino UNO pin D12 to the bottom-jumper-header center-pin of the Zigbee shield
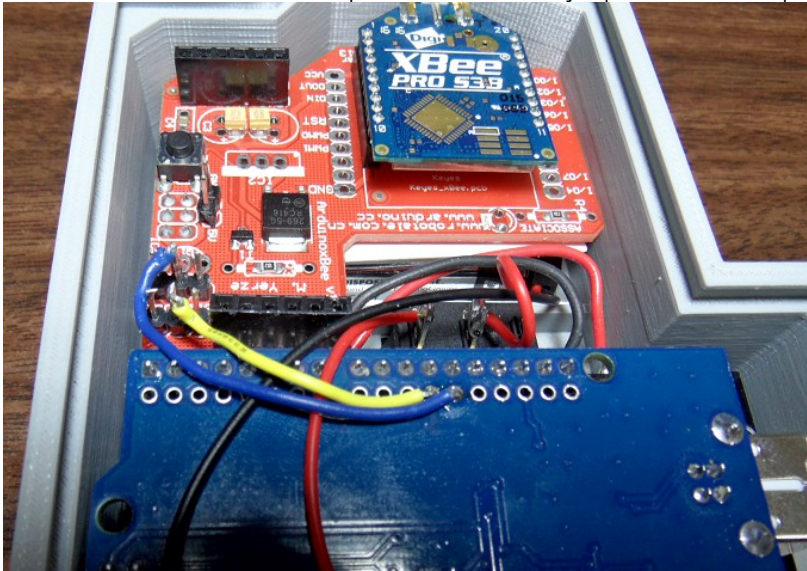
✔ Further Details for the Curious

    ○ Xbee and Arduino both use Serial Communications –- **but** they both cannot use the same Serial Port at the same time.
        ▪ In order to Upload Firmware to the Arduino the Xbee must be disconnected from the lines (No Jumpers)
        ▪ In order to Connect to the Xbee through USB Serial the Arduino Microcontroller must be disconnected (*UN-usable for our Arduino UNO*)
        ▪ Thus; We add Jumper-wires to put the Xbee on a different "Serial Port" when we need access to both Serial Devices.

    ○ The Factory Jumpers on the Zigbee Shield come with two modes of operations
        ▪ Xbee Position   = Xbee UART Serial connects to Arduino Serial
        ▪ USB Position   = Xbee UART Serial connects to USB Serial (Computer); Requiring the removal of the micro-controller.

    ○ Jumper-Wires for Debug
        ▪ Computer USB Serial → Arduino   - When NO jumpers exist; the Xbee is disconnected from RX/TX Pins and thus allows clean USB Serial Monitor.
        ▪ Routes Xbee UART → D11 & D12   - The SOFTWARESERIAL Library allows UART communications on D8 & D9 which is wire-jumped to the Xbee RX/TX.

    ○ Pin Orientation
        ▪ The Jumpers on the Zigbee Shield are aligned Top->Bottom Exactly the same as the Xbee Plug labeled DIN/DOUT are aligned
          ▫ On the shield (DIN is the closest jumper to Digital Pins Header on the Shield)

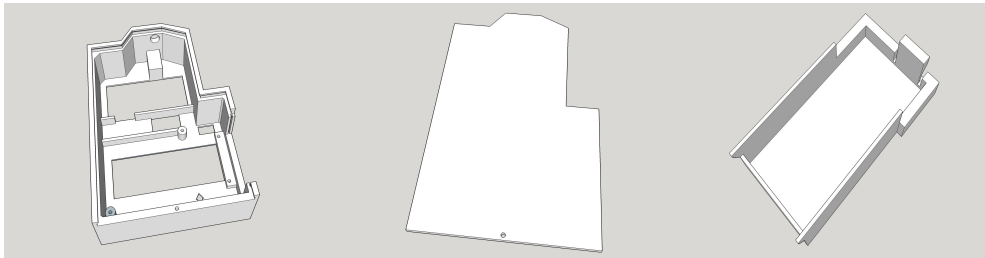## 6.2.2 Hardware Assembly

A. 3D-Prints

Illustration 16: Case.stl   Illustration 17: BackCover.stl   Illustration 18: Battery Cover
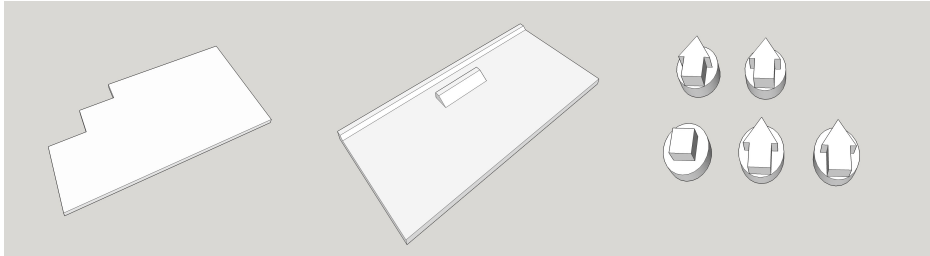
Illustration 19: BatteryInsulatorPlate.stl Illustration 20: SideCover.stl   Illustration 21: Buttons.stl

✔ Print the Following Models on a 3D-Printer
- ○ /HandRemote/3D-Prints-STL/HandRemote-Case.stl
- ○ /HandRemote/3D-Prints-STL/HandRemote-Buttons.stl
- ○ /HandRemote/3D-Prints-STL/HandRemote-BackCover.stl
- ○ /HandRemote/3D-Prints-STL/HandRemote-BatteryCover.stl
- ○ /HandRemote/3D-Prints-STL/HandRemote-BatteryInsulatorPlate.stl
- ○ /HandRemote/3D-Prints-STL/HandRemote-SideCover.stl

B. Install Electronics

1. Insert the Power ON/OFF Rocker Switch into the front of the case
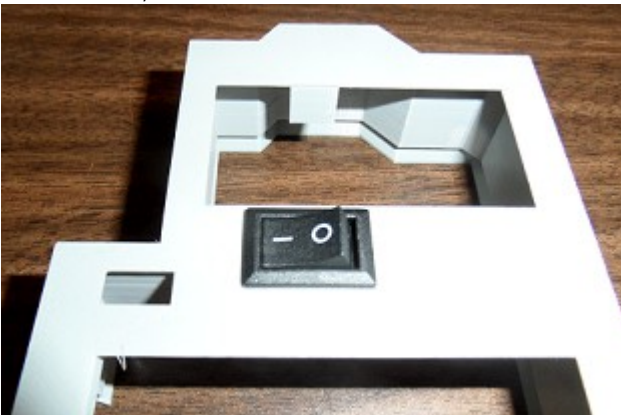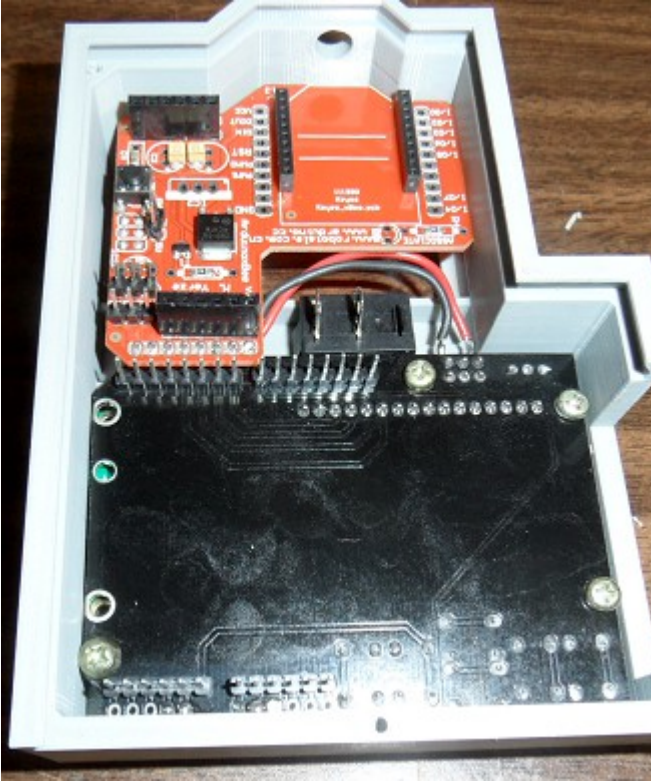

2. Place the buttons in their proper place inside the case; using masking tape on the front-side to hold the buttons in their place

3. Install the LCD, Buzzer, Zigbee **Electronics Assembly** (from Step #2.1) into the Case (Arduino UNO Removed)
   a) Insert (4) 10mm or shorter screws – adjust tightness making sure the buttons still click (*too tight will pinch buttons down permanently*)

4. Plug the Arduino UNO back onto the LCD Keypad Shield
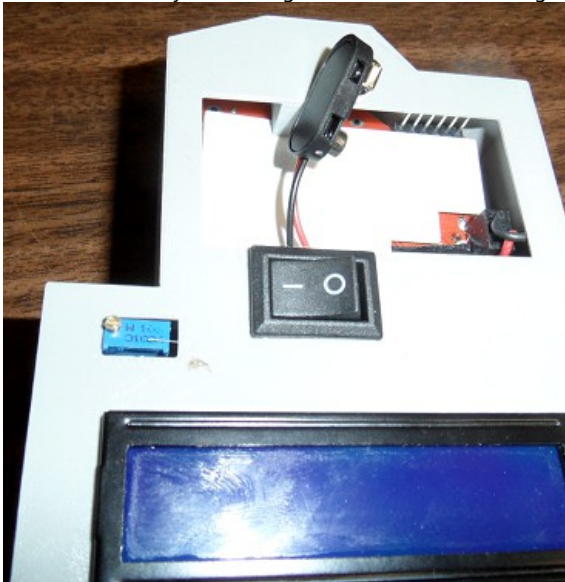5. Slide the Xbee into the Top-Hole and then plug into the Zigbee Shield
6. Attach the Antenna to the Xbee RPSMA Connector

C. Wire in Battery Power

1. Install Plugs
   a) Insert the HandRemote-BatteryInsulatorPlate.stl 3D-Print inside the battery hole
   b) Route the 9Vdc Battery Power Plug from the front, around Zigbee shield, to the back

2. Wire in the Switch and Power to the Arduino UNO
   a) Solder the **Black wire** to the **closest to UNO power-jack backside pin** (as pictured)
   b) Leaving an inch-or-so slack on the 9VDC plug; measure and cut the **Red wire** on the **closest rocker-switch connector**
   c) Strip the end and solder the wire to the rocker switch connector
   d) Strip the end of the **cut left-over red wire** and solder it into the **"other" connector on the rocker-switch**
   e) Solder the other end of the **cut left-over red wire** wire to the **farther to UNO power-jack backside pin** (as pictured)
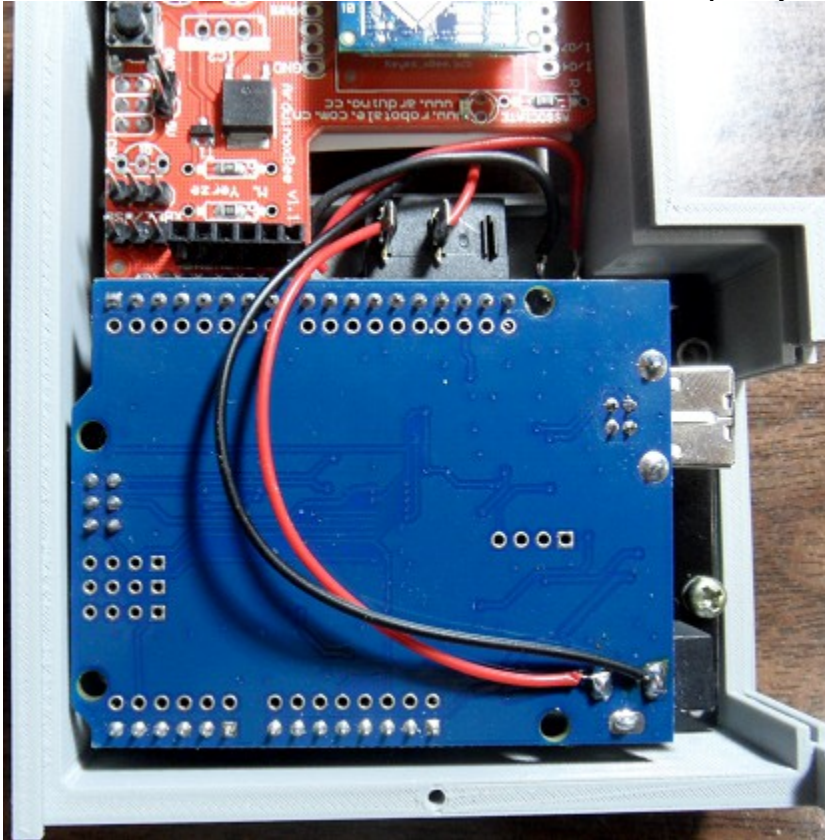
3. Check for Power
   a) Plug in a 9V-Battery into the 9V plug
   b) Flip the rocker-switch to power-on (marked with a "-") and verify that the LCD display lights

4. Attach Covers to Case
   a) Insert the HANDREMOTE-BATTERYCOVER.STL 3D-Print into the cases battery-hole and snap down
   b) Slide the HANDREMOTE-BACKCOVER.STL 3D-Print in from the bottom then insert a screw in the bottom-center
   c) Slide the HANDREMOTE-SIDECOVER.STL 3D-Print in the USB side until it snaps into place

6.3 PUMP CONTROLLER

✔ Features
   ○ Back-up Arduino power source for when power is lost (Track time off and turn-back-on automatically)
      ▪ Back-up battery level indicator and possible charge circuit
   ○ Timer and Logger of events

✔ Automation
   ○ Automatically turn off the pump if no water exists
   ○ Automatically re-start pump if a power failure has occurred
   ○ Set to Automatically turn on pump when water height is okay

6.3.1 Electronics Assembly

Wiring Diagram of the PUMP CONTROLLER - (Screw Terminal Perfboard → Zigbee Shield → Arduino UNO)
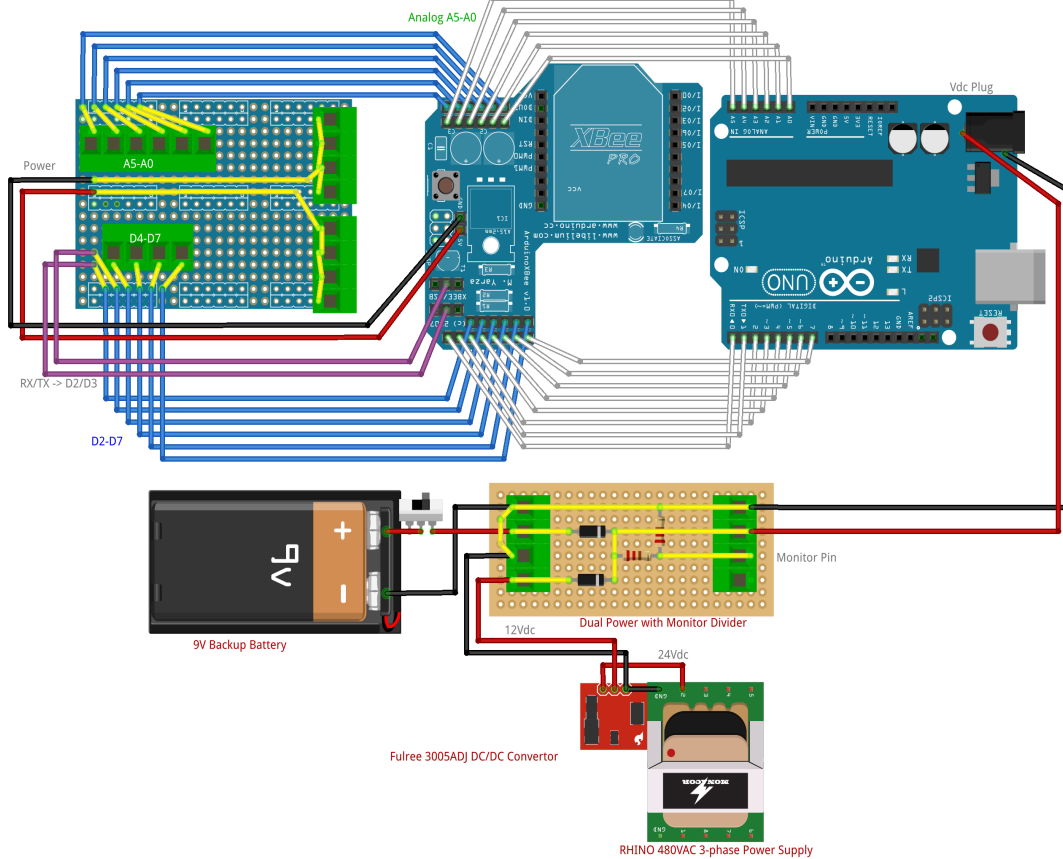

A. Screw-Terminal Perfboard

1. Install Headers
   a) Remove outer pins on one 4-PIN female header (POWER)
   b) Remove every other pin on one 4-PIN female header (RX/TX)
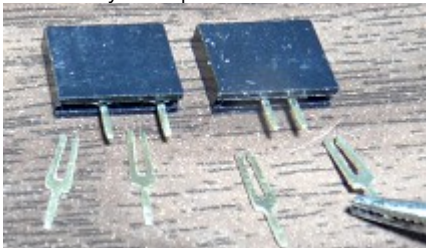
c) Remove the (2) Green Jumpers on the Zigbee Shield and plug it onto the Arduino UNO
d) Plug a 6-Pin Male Header into the Analog Header of the Zigbee Shield (A5 → A0)
e) Plug a 6-Pin Male Header into the Digital Header of the Zigbee Shield (D2 → D7)
f) Insert the POWER female header from step #a into the Zigbee Shield holes labeled 5V & GND (next to the reset button)
g) Plug a 4-Pin Male Header into the top of the POWER female header
h) Insert the RX/TX female header from step #b ACROSS the two center male header (XBEE/USB) pins of the Zigbee Shield
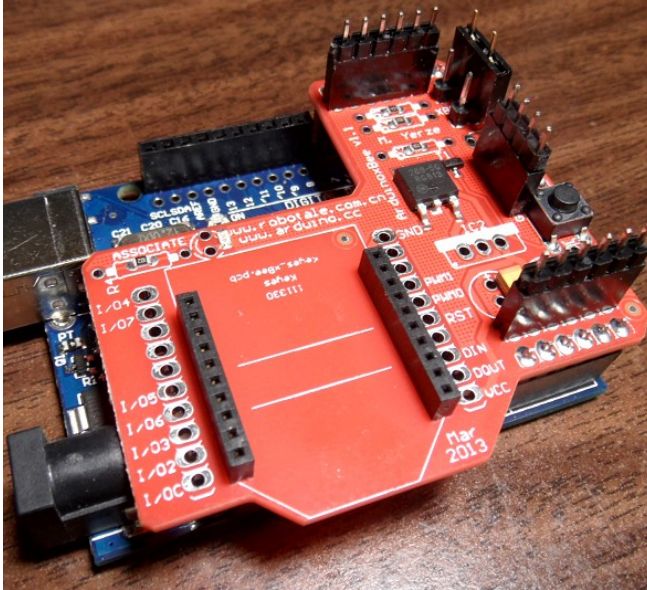
2. Install a 5cmx7cm 2.54mm pitch Perfboard and Solder the header pins
    a) Align and place the Perfboard onto the header pins
    b) Solder the header pins to the Perfboard
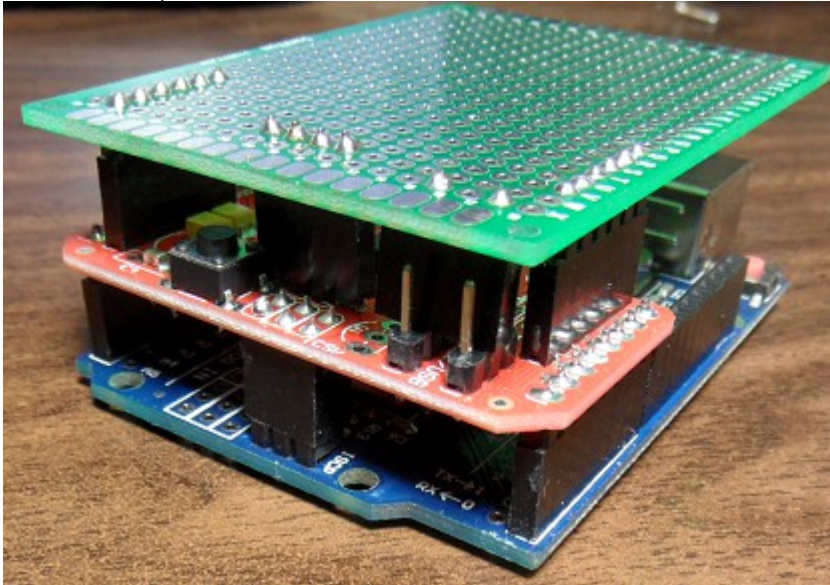
3. Solder on the Zigbee female POWER header
    a) Unplug the Arduino UNO from the Zigbee Shield leaving the Perfboard attached
    b) Solder the female POWER header pins to the holes of the Zigbee shield
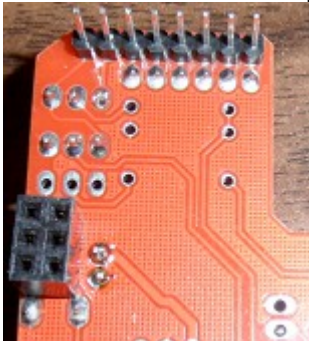
4. Solder on header pin wires
    a) Cut (12) wires approximately 1" long and strip one end.
    b) Cut (2) wires approximately 3" long and strip one end.

    c) Unplug the Perfboard from the Zigbee Shield
    d) Insert (6) 1" wires from the Zigbee-side of the Perfboard into the holes just **below** the analog male pin headers, bend and solder
    e) Insert (4) 1" wires from the Zigbee-side of the Perfboard into the 4-right (D4-D7) holes just **above** the digital male pin headers, bend and solder
    f) Insert (2) **3"** wires from the Zigbee-side of the Perfboard into the right of the POWER male pin headers (Center-2), bend and solder
    g) Attach (2) 1" wires from the Top-side of the Perfboard across RX/TX female pin headers and solder to D3, D4 Digital pins as shown and trim.
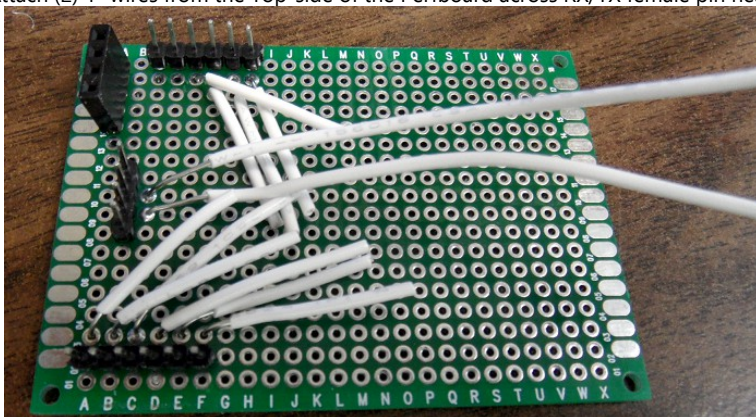

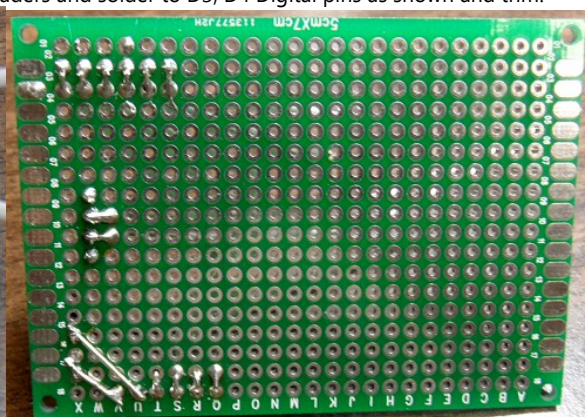*Illustration 22: Perfboard Zigbee-Side (Bottom)*   *Illustration 23: Perfboard Top-Side*

5. Place Screw Terminals
    a) Place the HANDREMOTE-PERFBOARDCOVER.STL 3D-Print on the top of the Perfboard lining up grooves with header pin solder joints
    b) Insert (3) 2x Screw Terminals along the Analog section starting at the Right and working left to allow sliding the blocks together at their edges
    c) Insert (2) 2x Screw Terminals along the Digital section with the same method above
    d) Insert (2) 2x Screw Terminals along the 5VDC section (same as above)
    e) Insert (2) 2x Screw Terminals along the GND section (same as above)
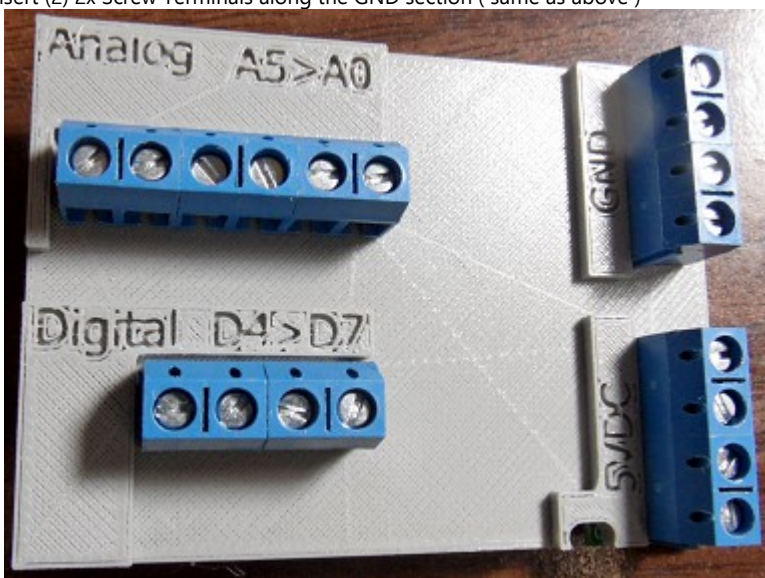
6. Solder Screw Terminals and Wires
    a) Turn the Perfboard over holding the screw terminals into place
    b) Sold each screw terminal pin to the Perfboard
    c) Pull the Insulation off the wires, bend them flat with the Perfboard and solder them to their corresponding pin (See picture or Wiring Diagram)
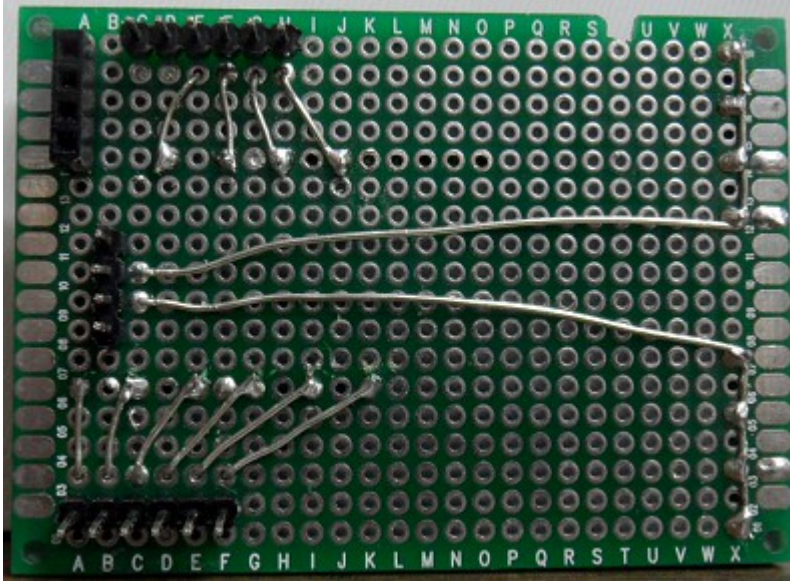

B. Dual-Power Perfboard

1. Using a 3cm x 7cm Perfboard

2. Install Components
    a) (2) Diodes – Any 1/4 Watt Diode Will do
    b) (2) Resistors – Diagram shows a (910K Diodes → 330K → GND), However any values that produce a voltage divider less than 1/2 will work.
        • This sets up a Voltage Divider; The maximum input voltage on the Arduino Analog Pins is 5Vdc so we have to divide the voltage
        • See https://www.abelectronics.co.uk/tools/resistor-voltage-divider.aspx
    c) (2) 4-Screw Terminal Blocks Assemblies
    d) Solder the leaded components onto the board from the top-side for stability
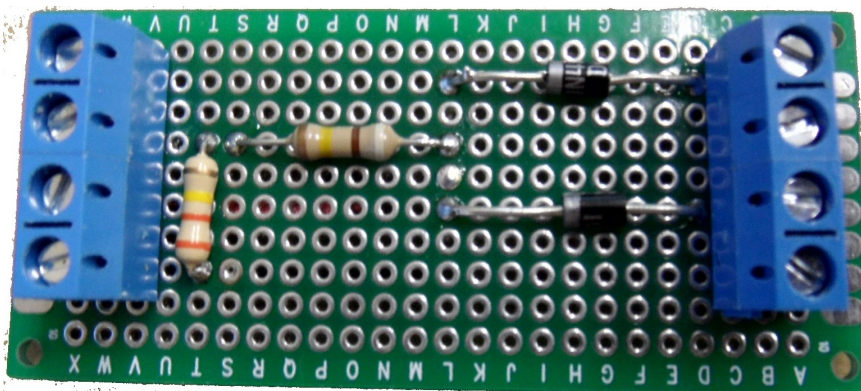
3. Solder Connections
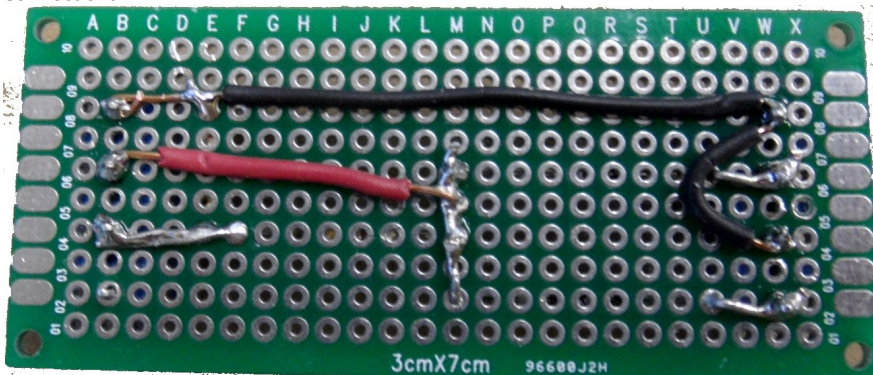
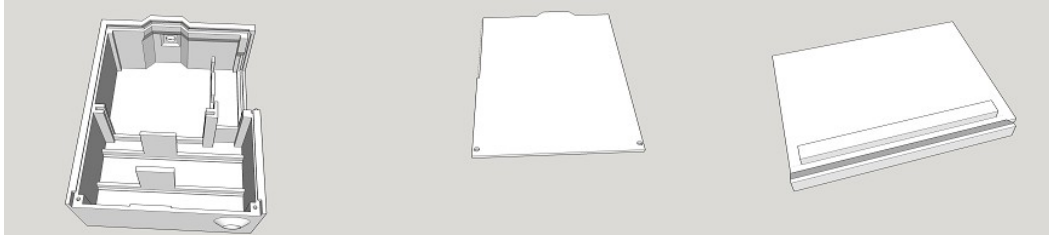## 6.3.2 Hardware Assembly

A. Print Models (3D-Prints)

*Illustration 24: Case.stl*   *Illustration 25: Cover.stl*   *Illustration 26: SideCover.stl*

1. 3D-Print the Following Models @ https://github.com/tgit23/AgIrrigationRemoteControl/tree/master/PumpController/3D-Prints-STL
   ○ Case.STL
   ○ Cover.STL
   ○ SideCover.STL

B. Install Electronics

1. Install UNO, Zigbee & Xbee Module into the Case
   a) Remove the Screw Terminal Perfboard if attached and connect the Arduino UNO & Zigbee Shield
   b) Insert the Xbee Module into the Zigbee Shield and remove the nut and washer
   c) Angle the UNO/Zigbee/Xbee combo in the Case and move forward; inserting the Xbee RPSMA Connector through the case top hole
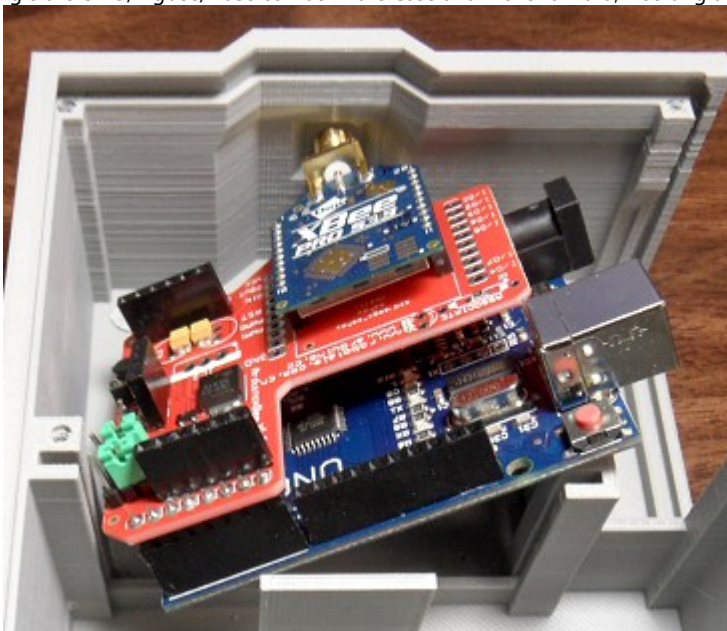
2. Install the Screw Terminal Perfboard
    a) Seat the UNO/Zigbee/Xbee combo in the Case
    b) Install the 6-Long Pin Female Header into the Arduino
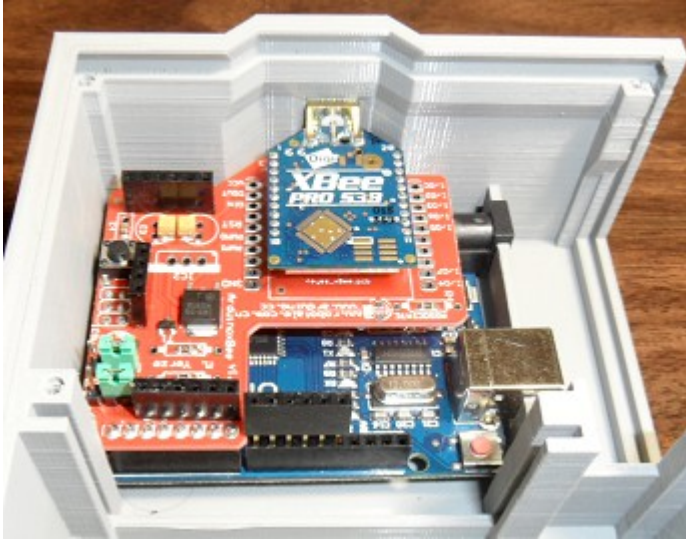
    c) Place a little bit of black tape on the Screw Terminal Perfboard back-side where it hangs over the Xbee Module to prevent any shorts
    d) Plug the Screw Terminal Perfboard into the Zigbee/UNO

3. Install the Xbee Antenna
    a) Put the nut and washer on the Xbee RPSMA Connector on the outside of the case
    b) Screw the high gain antenna onto the RPSMA Connector

C. Wire in Power

1. Install Power
    a) Wire the Power Units to the Dual-Power Perfboard (9Vdc Storage Box & Fulree DC/DC Converter)
    b) Wire the Arduino DC Power Jack to the Dual-Power Perfboard
    c) Slide the Dual-Power Perfboard into the Center-of-Case Groove just under the Arduino UNO
    d) Slide the 9Vdc Battery Storage Case and Fulree DC/DC Converter into the Tabbed Slots just below that.
    e) Plug the DC Power Jack into the Arduino UNO

2. Slide the SideCover into the Right-Top-Side of the Case to Cover the DC Power & USB plugs
3. Roll and apply Plumbers putty around the RPSMA Antenna Plug and Install Nut and Washer
4. Roll and apply Plumbers putty around the Cover and attach with screws to the top of the Case