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

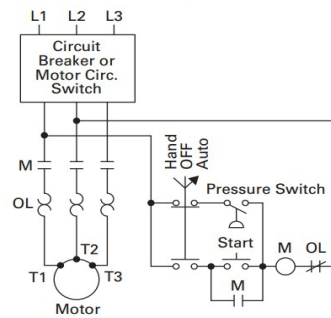 ○ Box   - Eaton Irrigation Pump Control Type 3R Enclosure Rainproof
    ▪ http://www.eaton.com/Eaton/ProductsServices/Electrical/ProductsandServices/AutomationandControl/EnclosedControl/NEMA/PumpPanels/index.htm

○ 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:
    - ◦ Activate Coil   - Shown as a circle in the wiring diagram just right of the Start Button
    - ◦ Main Contacts   - The Contacts that connect the L1,L2,L3 lines → OL → Motor
    - ◦ 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/FreedomSeries/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. OPTIONAL ACCESSORIES

4.1 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

4.2 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

4.3 Water Meter

5. 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

5.1 MATERIALS (BOM) - $190.25

5.1.1 Pump-Panel Components ($87.75)

Illustration 3: Power Supply

Illustration 4: Solid State Relay

Illustration 5: DIN Rail

○ RHINO switching power supply, 24 VDC (adjustable) output, 2.5A, 60W, 320-600 VAC input, 3-phase, aluminum housing, 35mm DIN rail mount. ($59.00)
  ▪ https://www.automationdirect.com/adc/Shopping/Catalog/Power_Products_(Electrical)/DC_Power_Supplies/24VDC,_3-Phase_Input/PSB24-060S-3
  ▪ Alternative
    ▫ If purchasing a 12VDC power supply like this one www.mouser.com/ProductDetail/TDK-Lambda/DPP120-12-3/ ($104.04) the DC-to-DC Converter ($7.99)
      won't be necessary.

○ Solid state relay, 35mm DIN-rail mount, 3-32 VDC input voltage, SPST, normally open SCR, 10A contact rating, 48-480 VAC load voltage, zero cross. ($20.75)
  ▪ 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

○ DIN rail, slotted, 35mm, 10 mm height, 1m length, aluminum. Package of 2 ($8.00)
  ▪ 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

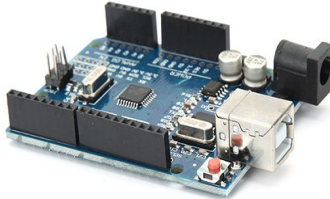### 5.1.2 Electronic Control ($54.63)

Illustration 6: (2) Arduino UNO R3
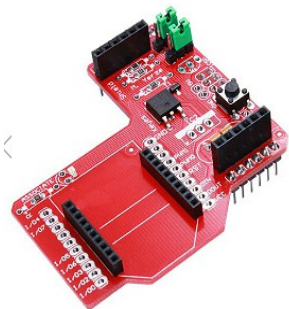

Illustration 7: LCD Keypad Shield

Illustration 8: (2) Zigbee Shield

○ (2) Geekcreit UNO R3 Atmega328P Development Board for Arduino ($3.34/ea)
  ▪ http://us.banggood.com/Wholesale-Warehouse-UNO-R3-ATmega328P-Development-Board-For-Arduino-No-Cable-wp-Usa-964163.html

○ Keypad Shield Blue Backlight for Arduino Robot LCD 1602 Board ($4.11/ea)
  ▪ http://www.banggood.com/Keypad-Shield-Blue-Backlight-For-Arduino-Robot-LCD-1602-Board-p-79326.html
  ▪ Resources
    ▫ 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 ($4.83/ea)
  ▪ http://www.banggood.com/Zigbee-Shield-RF-Wireless-Module-Expansion-Board-For-Arduino-XBee-p-916226.html
  ▪ Resources
    ▫ This Schematic is the right one https://www.arduino.cc/en/uploads/Main/XbeeShieldSchematic.pdf
    ▫ https://www.cooking-hacks.com/documentation/tutorials/xbee-arduino-raspberry-pi-tutorial/
    ▫ http://www.cooking-hacks.com/skin/frontend/default/cooking/pdf/arduino_XBee.pdf

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

○ 20pcs 2 Pin Plug-in Screw Terminal Block Connector 5.08mm Pitch ($2.15)
  ▪ http://www.banggood.com/20pcs-2-Pin-Plug-in-Screw-Terminal-Block-Connector-5_08mm-Pitch-p-993197.html

○ Fulree DC 8-35V to 1.5-24V Adjustable Buck Converter Power Supply Voltage Regulator For Car Truck Boat ($7.99)
  ▪ 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
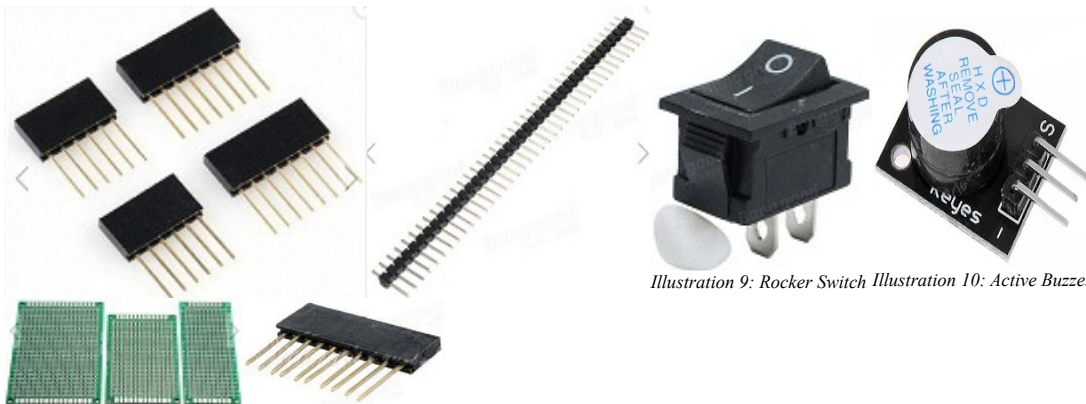
Illustration 9: Rocker Switch Illustration 10: Active Buzzer

○ 10P 2.54MM Stack-able Long Connector Female Pin Header ($1.19)
  ▪ http://www.banggood.com/Stackable-Long-Pins-Header-Connector-Female-Kit-p-88504.html

○ 30 Pcs 40 Pin 2.54mm Single Row Male Pin Header Strip For Arduino Prototype Shield DIY ($3.99)
  ▪ http://www.banggood.com/30-Pcs-40-Pin-2_54mm-Single-Row-Male-Pin-Header-Strip-For-Arduino-Prototype-Shield-DIY-p-1033757.html

○ Black Snap-in On/Off Rocker Switch ($3.34)
  ▪ http://www.banggood.com/20Pcs-Black-Push-Button-Mini-Switch-6A-10A-110V-250V-KCD1-101-2Pin-Snap-in-OnOff-Rocker-Switch-p-1096193.html

○ Active Speaker Buzzer Alarm ($1.17)
  ▪ http://www.banggood.com/Arduino-Compatible-Active-Speaker-Buzzer-Alarm-Module-For-PC-Printer-p-76358.html?rmmds=search

○ 40Pcs FR-4 Double Side Prototype PCB Printed Circuit Board ($9.69)
  ▪ http://www.banggood.com/40Pcs-FR-4-Double-Side-Prototype-PCB-Printed-Circuit-Board-p-995732.html

5.1.3 RF Electronics ($80.88)

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

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

5.1.4 Miscellaneous Components

○ (7) Small Screws
○ (4) Resistors of Value 10K or higher
○ (1) Right-Angle 9Vdc Power Plug for Arduino Power Jack
○ (2) Diodes
○

5.2 HAND REMOTE

The HANDREMOTE is the remote controller unit used to control the pump remotely, display the monitored values and notify the user of any malfunctions using alarm settings.
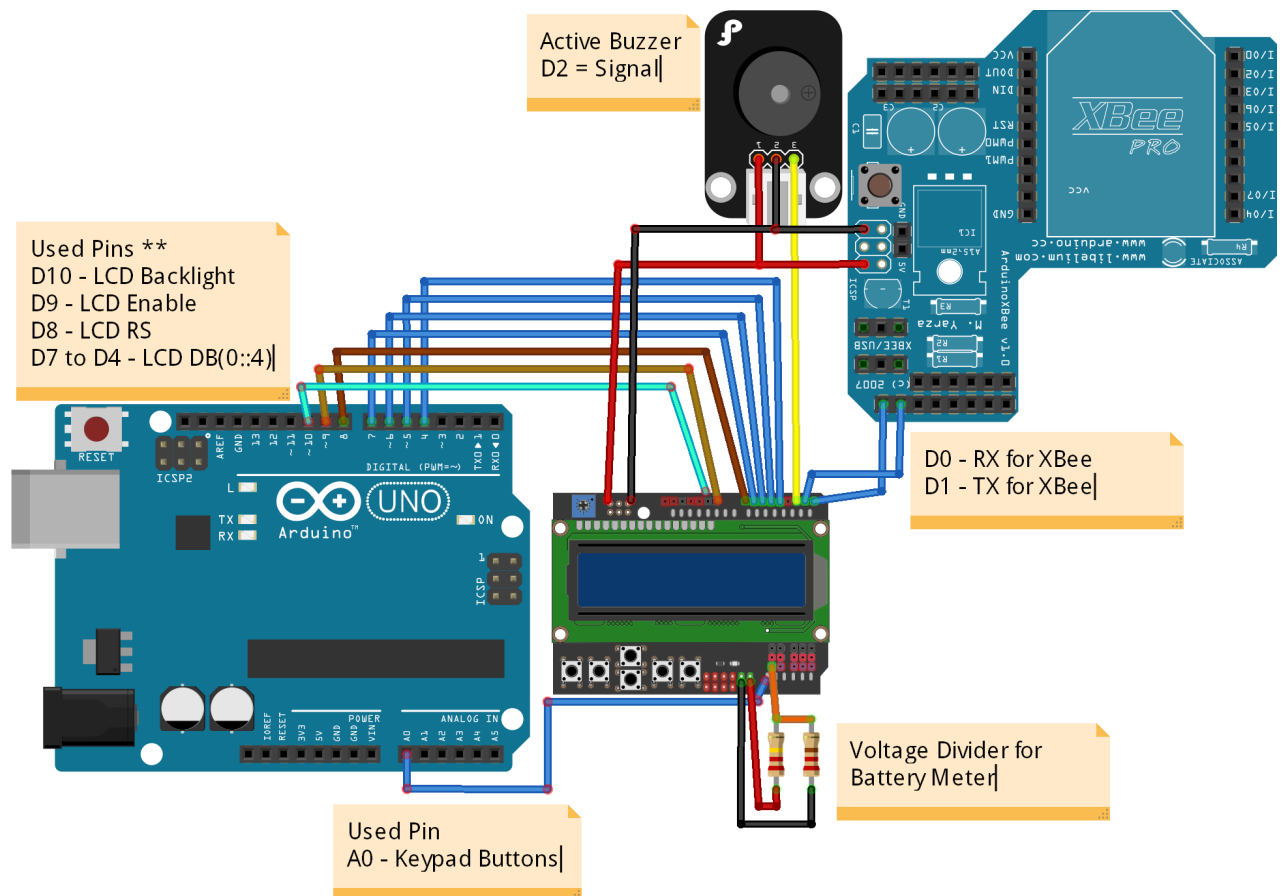
- ✔ Features
  - ○ Buzzer alarm settings for monitoring values
  - ○ Battery level indicator
  - ○ Programmable Interface through USB

5.2.1 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
  - ○ 1- 9Vdc Battery Connector
  - ○ 1- Active Buzzer
  - ○ 1- Helical Antenna

Active Buzzer
D2 = Signal

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

D0 - RX for XBee
D1 - TX for XBee

Voltage Divider for
Battery Meter

Used Pin
A0 - Keypad Buttons

fritzing

A. LCD Keypad Shield

1. Wire the Battery-Meter Voltage-Divider
       a) Select (2) Same-Value Resistors (10K-ohm up to 1M-ohm is recommended)
       b) Solder First Resistor on the top of the LCD Shield from **VIN** → **A1**
       c) Solder Other Resistor 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 1/2 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.
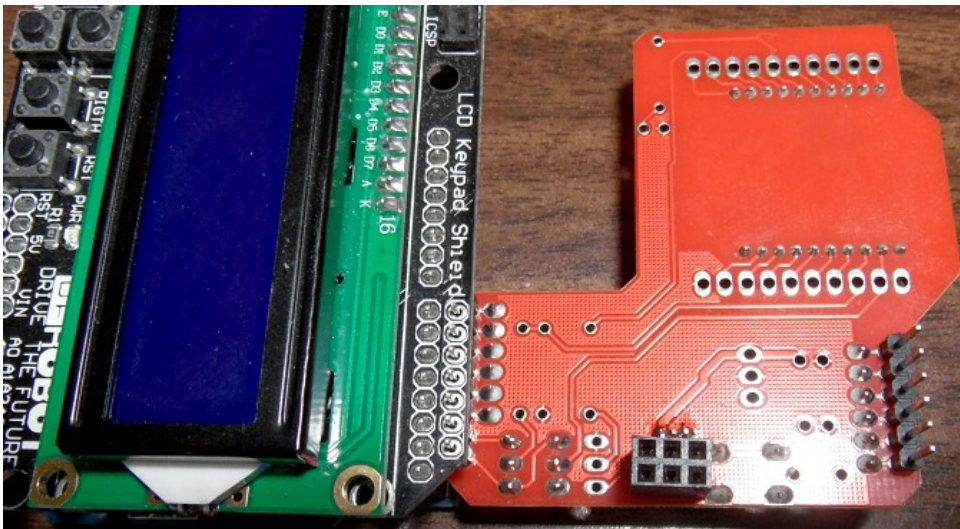
2. Plug the LCD Keypad Shield onto the Arduino UNO

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
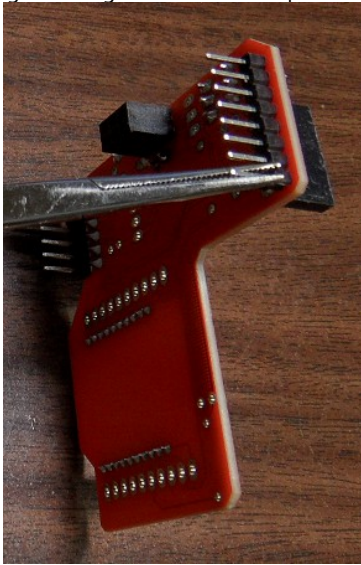
2. Solder Zigbee Shield to the LCD Keypad (Flipped upside down)
    a) Insert pins D0 → D6 (D7 removed in Step #1) into the Top Solder holes of the LCD Keypad Shield
    b) Position the Zigbee Shield flat with LCD Shield and Up against the LCD Shield – Requires bending the header pins a little (with 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.
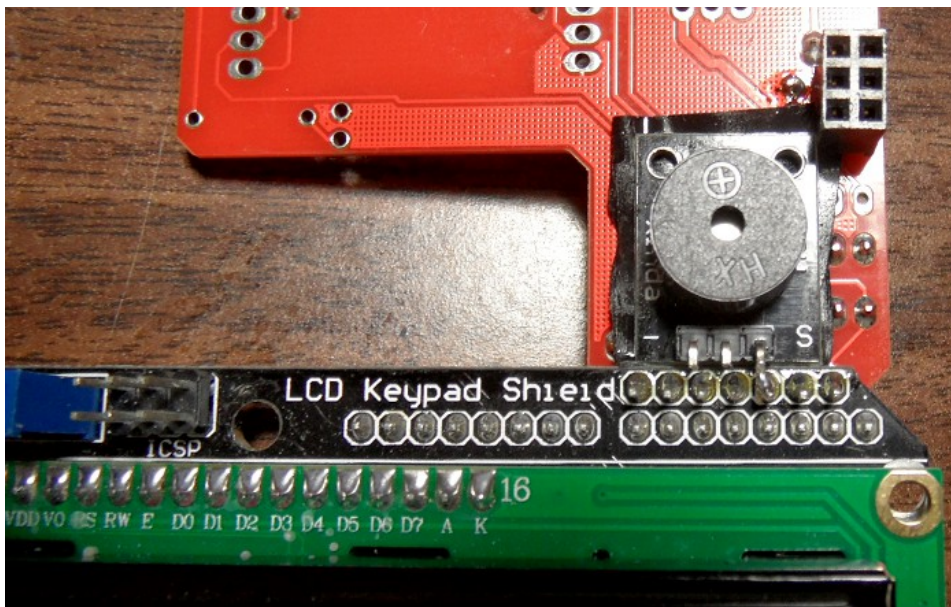
2. Wire Power to the Active Buzzer and Zigbee Shield

    a) 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

    b) 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.
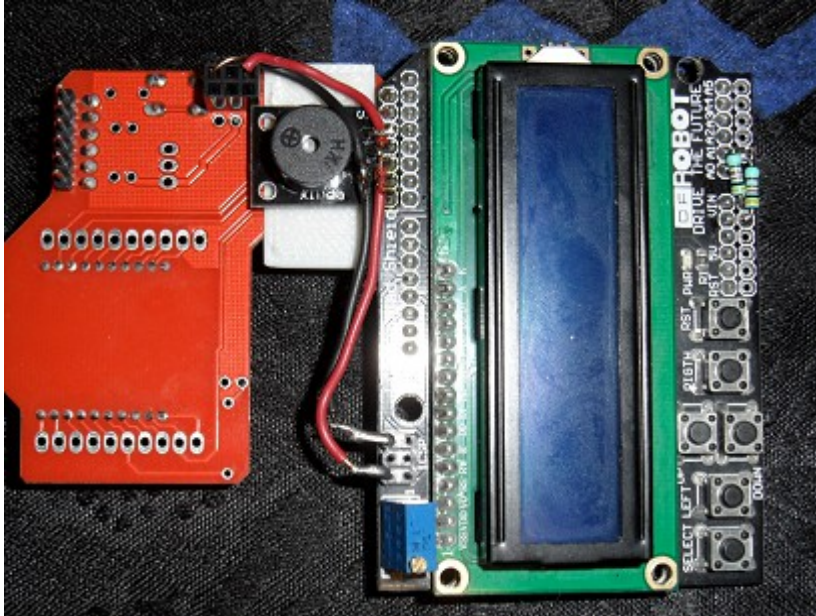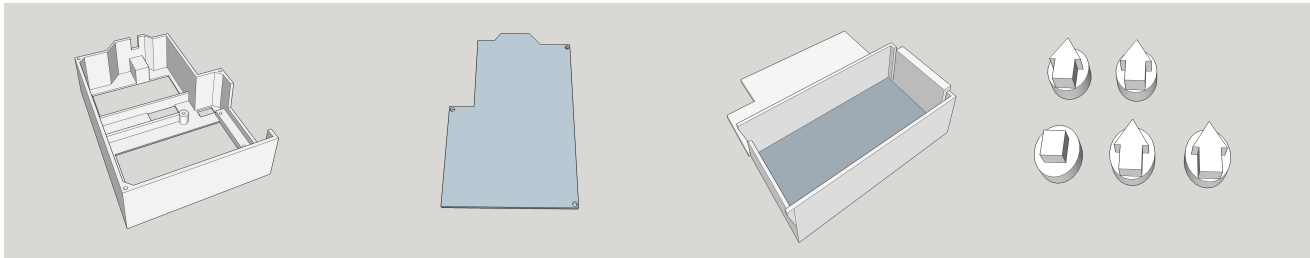
5.2.2 Hardware Assembly

A. 3D-Prints

Illustration 11: Case.stl	*Illustration 12: BackCover.stl*	*Illustration 13: Battery Cover*	*Illustration 14: Buttons.stl*

✔ Print the Following Models on a 3D-Printer
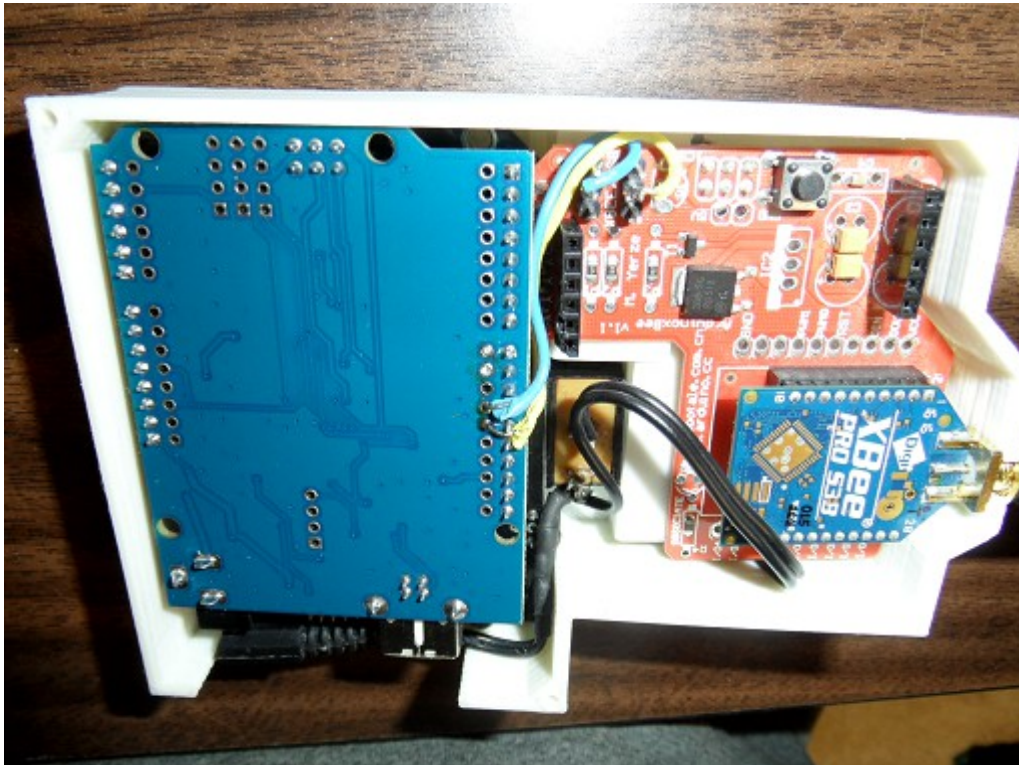- ○ Case   WL-HandRemote-Case.STL
- ○ BackCover   WL-HandRemote-BackCover.STL
- ○ Buttons   WL-HandRemote-Buttons.STL
- ○ BatteryCover   WL-HandRemote-BatteryCover.STL

B. Install Electronics

1. Insert the Buttons in their proper slots inside the Case

2. Install the LCD, Buzzer, Zigbee Electronics Assembly (from Step #2.1) into the Case

3. Slide the Xbee into the Top-Hole and then plug into Zigbee Shield

C. Install Battery Power

1. Snap the ON/OFF Toggle switch into the Case from the front-side
2. Route the 9Vdc Battery Power Plug from the front, around Xbee shield, to the back
3. Wire (Red) wire to one side of the switch
4. Wire (Black) wire to the 9Vdc Power Plug (Jack)
5. Wire (Red) wire from other side of switch to the 9Vdc Power Plug (Jack)

5.2.3 Firmware & Debug

✔ Run / Upload / Debug Configurations – The HANDREMOTE requires different hardware configurations for special operations.

   ○ **Run-Mode**   = When **Jumpers exist** on Zigbee Shield in the XBEE Position

   ○ **Upload-Mode**   = When **NO jumpers** are on the Zigbee Shield.
      ▪ Allows Uploading Firmware to the Arduino.

   ○ **Debug-Mode**   = When **NO jumpers** are on on the Zigbee Shield and two **Jumper-Wires** exist from the Arduino to the Zigbee Shield.
      ▪ Jumper-Wires go from Arduino **D8 & D9** → **Center Pins** on Zigbee Shield
      ▪ Allows Uploading Firmware to the Arduino
      ▪ Enables the Arduino Serial Monitor while still being Operational

✔ 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. (Debug Mode)

   ○ 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 → D8 & D9   - 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 DIN/DOUT are aligned
         ▫ On the shield (DIN is the closest jumper to Digital Pins Header on the Shield)

A. Uploading Firmware

1. Install Arduino Sketch IDE   https://www.arduino.cc/en/Main/Software

2. Install the PEERIOSERIALCONTROL Library
      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. Install HANDREMOTE 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 the File HANDREMOTE.INO within the folder unzipped in step 'd' above
      f) Select the Port the HANDREMOTE USB is connected to; in Sketch menu TOOLS → PORT
      g) Select the Board; Sketch menu TOOLS → BOARD → ARDUINO/GENUINO UNO
      h) Setup the HANDREMOTE to UPLOAD-MODE by removing both Zigbee Shield Jumpers
      i) Press the Up-Arrow in the Top-Left Corner to upload the firmware onto the HANDREMOTE

The LCD should start showing readable characters. Use the Operation Instructions to verify its correct operation

B. 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


5.3 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

5.3.1 Build Electronics

Wiring Diagram of the PUMP CONTROLLER - (Arduino Base → Zigbee Shield → Screw Terminal Perfboard)
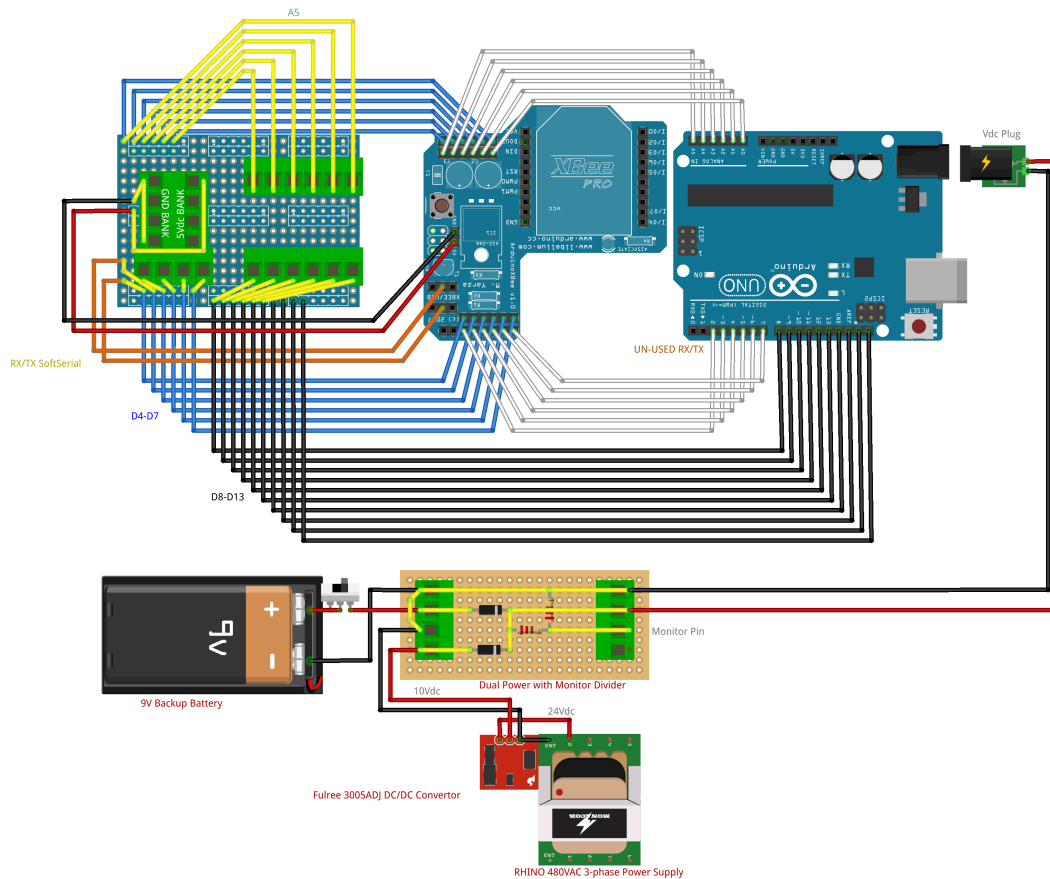
A. Install Zigbee Shield

1. Remove the Jumpers from the (XBEE/USB) Headers
2. Solder a New 2-Pin Male Header into the Holes for (GND / 5V) as shown below
3. Plug the Zigbee Shield onto the Arduino UNO
4. Plug the Xbee PRO S3B Module into the Zigbee Shield

Solder New Header Here

Remove Jumpers
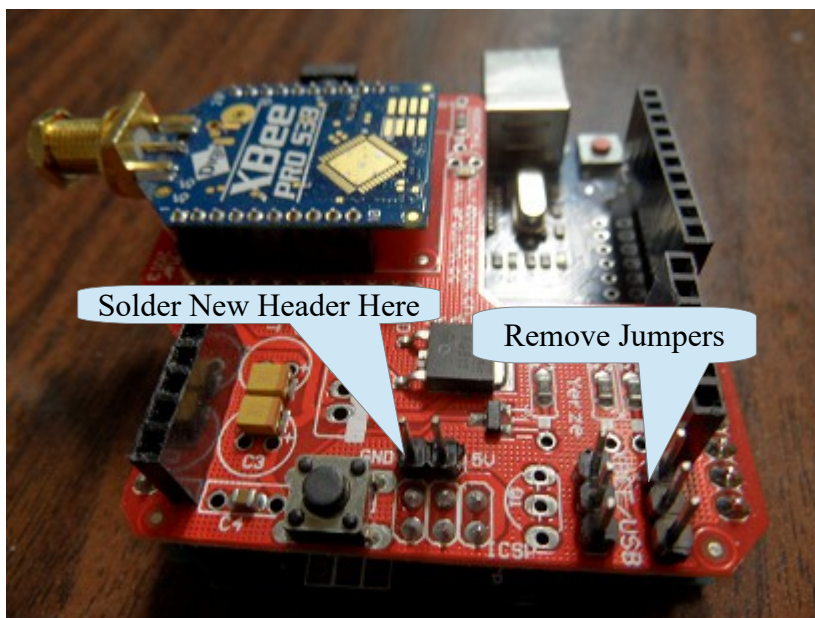
B. Build Screw-Terminal Perfboard

Note: Its important to plug in all the headers BEFORE soldering them in, to get a proper alignment

1. Plug a 2, 4, or 6P Female header onto the (GND/5V) male header you soldered onto the Zigbee Shield in Section 5.2.1
2. Plug a 2, 4, or 6P Female header across the CENTER pins of the (XBEE/USB) male header the jumpers were removed from in Section 5.2.1
3. Plug 6P Male headers onto the two 6P – Female headers which are part of the Zigbee Shield
4. Plug a Long 10P Female Header into the Arduino to extend D8-D13 up to the same level as the Zigbee Female Headers
5. Plug a 10P Male header into the Long 10P Female Header just attached in Step #4 above.
6. Drop the 5cm x 7cm Perf board onto the top of the Zigbee Shield lining up the headers plugged in above to their nearest Perf Board hole.
7. Solder the header pins to the top of the Perf board sitting on the Zigbee Shield
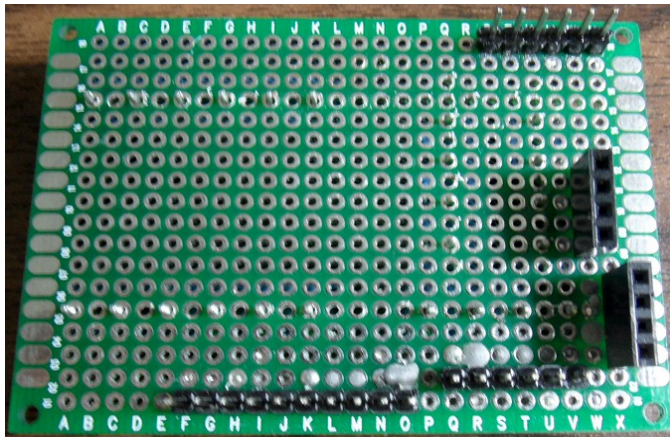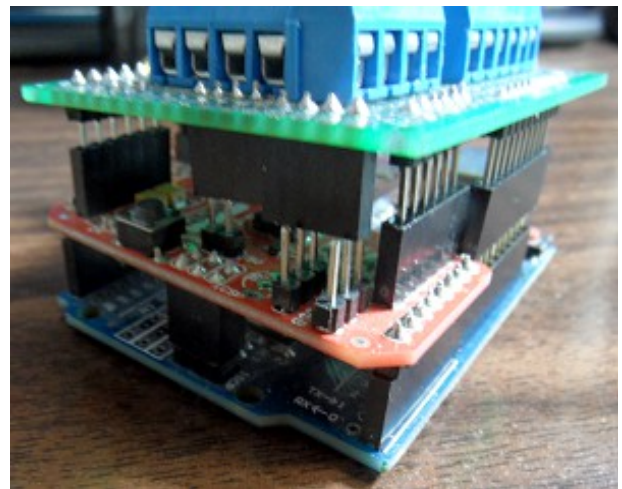

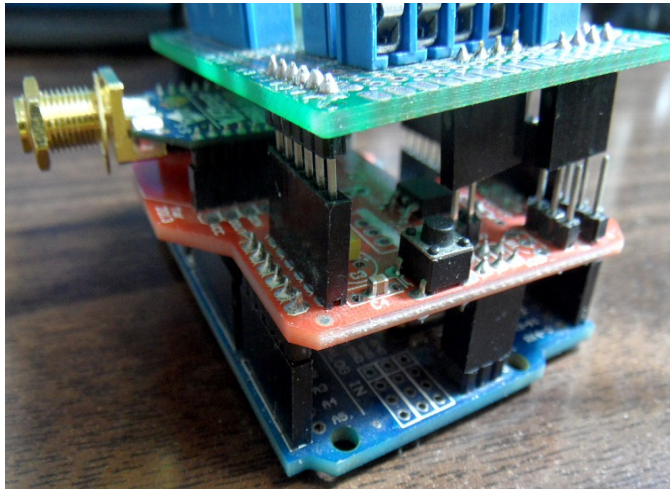*Illustration 15: Under-side with headers facing right*
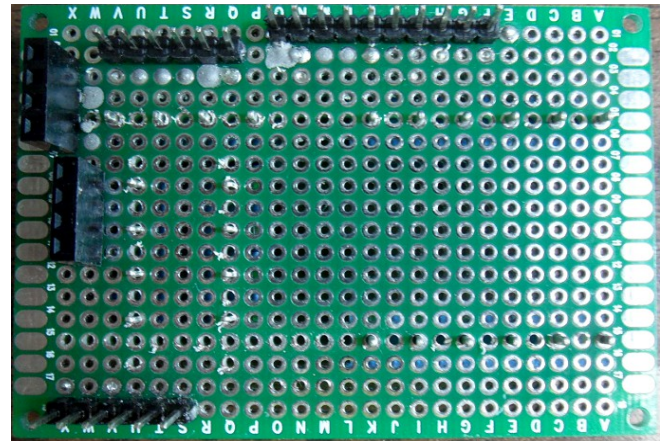

Illustration 16: Under-side with other facing left

8. Organize and Prep the Screw Terminals

    a) Slide together the Screw-Terminals into **(2) 6-Screw** and **(3) 4-Screw** sets (*Notice: Blocks have slide together seams on their edges*)

    b) **Temporarily** insert them into the Perfboard as shown below
- EITHER pinch the pins with pliers until they fit inside the holes
- OR the Perfboard holes can be drilled out to 1.1mm in order to get a proper fit.
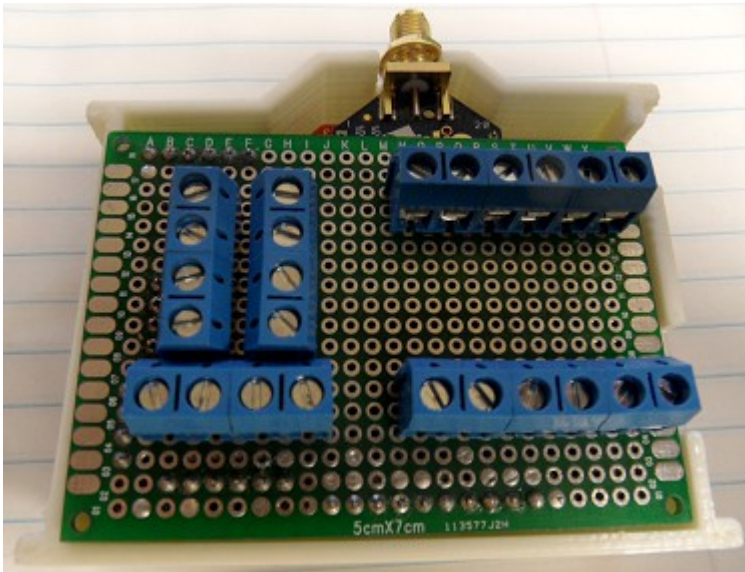
9. Solder in the jumper wires

    a) Use copper hard-core breadboard jumper wire

    b) 1st - Fill the holes the wires will be inserted into with solder

    c) 2nd - Strip one end about 1/2" from the end

    d) 3rd - Heat the solder and insert the wire down to the insulation. The cooling solder will stabilize the wire

    e) Repeat for all holes shown below next to the Male Headers and the GND/5V Female Header
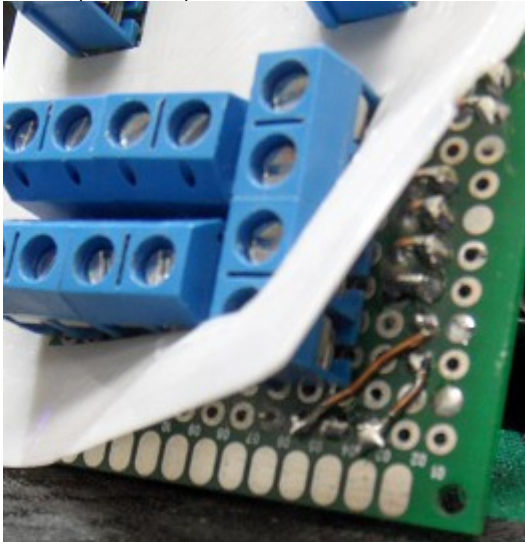- Refer to Wiring Diagram @ Section 5.2
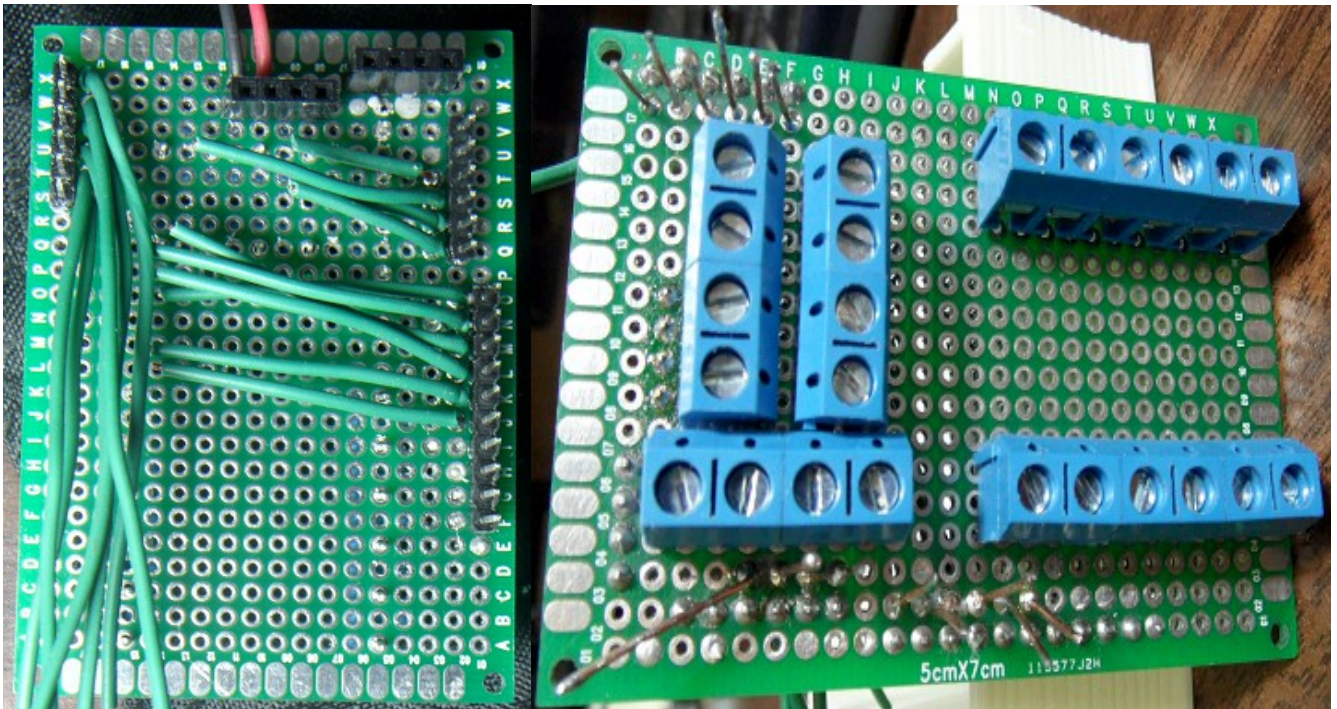
10. Solder the Top-Side jumper wires to the Pin Headers
    a) Cut to length, bend-over, and Solder jumper wire ends to Header Pins
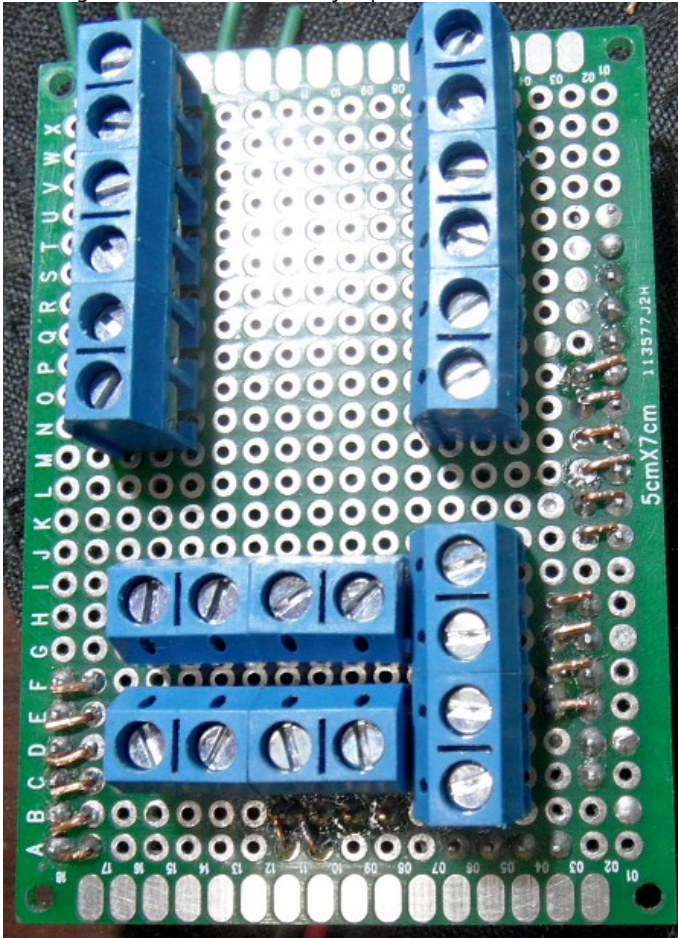

11. Attach two Top-Side Jumpers from A2 → XBEE DIN and A3 → XBEE DOUT

12. Finally Install the Screw Terminal Blocks

    a) Remove the Screw Terminal Blocks from the Perfboard
    b) Place the Plastic Lining (3D-Prints) as shown above over the Perfboard aligned so the Terminal Blocks can be placed in the openings
    c) Turn the Perfboard over and Cut, Strip and Solder the Jumper Wires to their Destination
        • Again Refer to Wiring Diagram @ Section 5.2
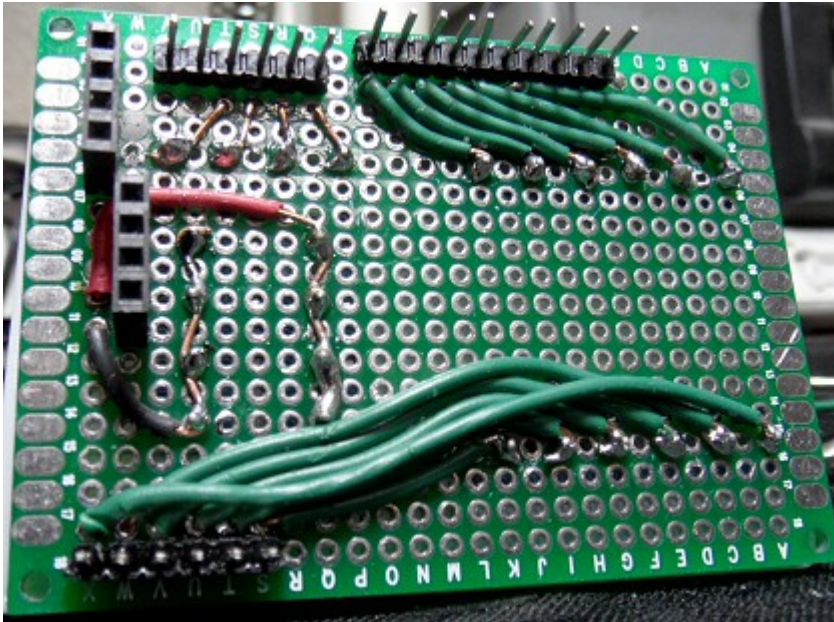

13.

C. Build 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
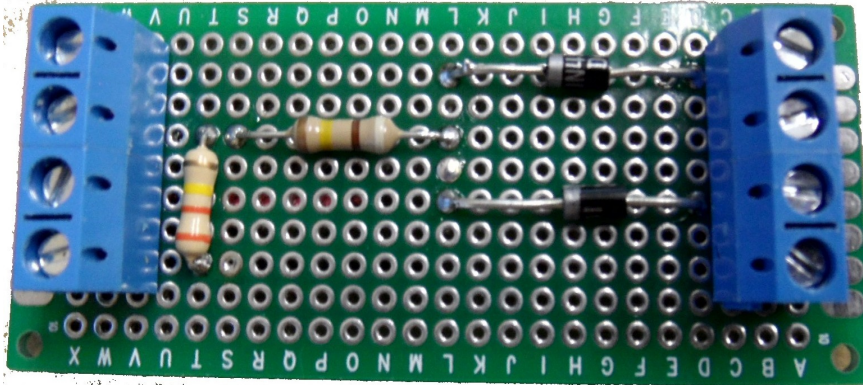
    3. Solder Connections
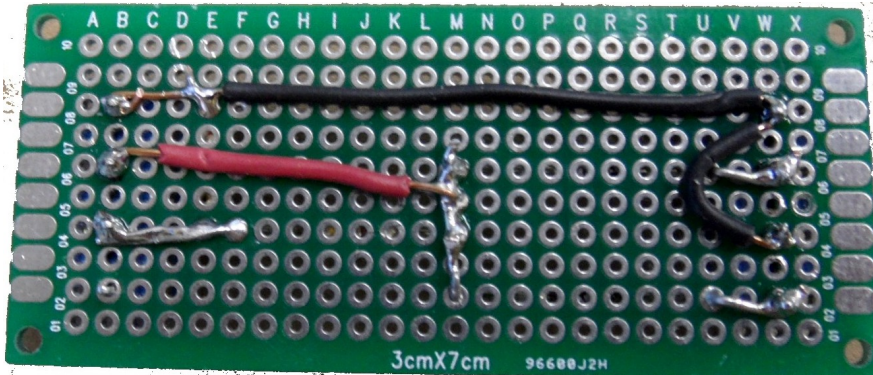
5.3.2 Hardware Assembly
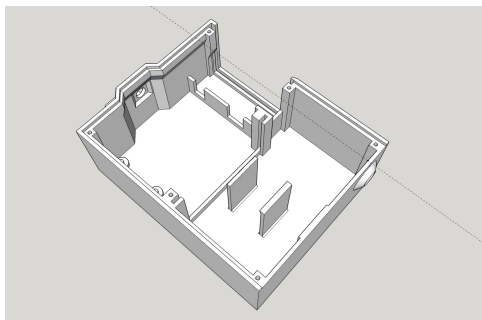
A. Print Models (3D-Prints)
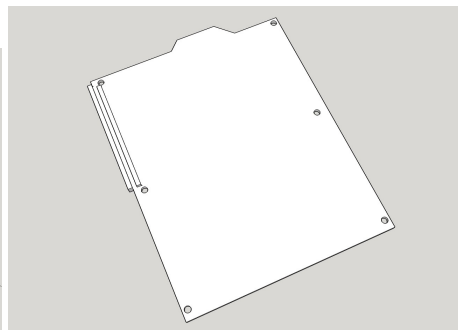

Illustration 17: Case.stl
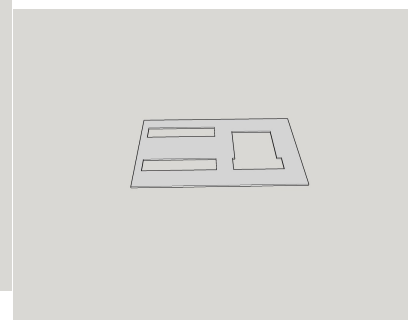

Illustration 18: Cover.stl

Illustration 19: ScrewTerminal-PCB-Cover

✔ Path to 3D-Print Models @ https://github.com/tgit23/AgIrrigationRemoteControl/tree/master/PumpController/3D-Prints
    ○ Files
        ▪ .skp   - Google SketchUp Make 2017 File
        ▪ .stl   - STereoLithography File
        ▪ .gcode   - G-Code for Case created using Slic3r and Marlin Firmware for a 1.75mm Filament (Plastic Scribbler 1100 AD)

B. Install Components

    1. Remove the Screw Terminal and Xbee Module from the Arduino Stack

    2. Insert the Arduino UNO & Zigbee Shield into the Left-Top cubby of the Case; aligning the Power Jack and USB Jack with Case fitting on the right

    3. Insert the Xbee RPSMA Antenna plug through the Top hole of the case and plug it into the Zigbee Shield

    4. Plug the Screw Terminal into the Zigbee Shield

    5. 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

    6. Slide the SideCover into the Right-Top-Side of the Case to Cover the DC Power & USB plugs

    7. Roll and apply Plumbers putty around the RPSMA Antenna Plug and Install Nut and Washer

    8. Roll and apply Plumbers putty around the Cover and attach with screws to the top of the Case

5.3.3 Install Firmware