Ag Irrigation Remote Control

Using Arduino-UNO and Digi-Xbee RF

Written By: Thomas G (01/2017)

Last Updated: 2017.06.27

Illustration 1: HandRemote

Illustration 2: Pump-Controller

Table of Contents

1. INTRODUCTION

The "***Ag Irrigation Remote Control***" project's goal is to provide Radio Frequency (RF) remote control and monitoring abilities to an Agriculture-Industrial 3-Phase 480VAC Irrigation Pump that is typically used in wheel-line and hand-line irrigation setups within the United States.

- ✔ The project implements
  - Digi Xbee RF radio modules (S3B, 900MHz, 1/4-Watt) which advertises a broadcast range of up to 28-miles (*Practical usage at 2-miles at ground level*)
  - Arduino micro-controller boards, LCD display shields, Zigbee shields, and various discrete devices

- ✔ The project consists of:
  - Hand-Held "***Hand Remote***"
  - Pole mounted "***Pump Controller***"
  - High-Voltage Control "***Installation Kit***" which consists of components to be added to the pumps electrical panel

- ✔ The project is Open Source and its main page Resides at: https://github.com/tgit23/AgIrrigationRemoteControl
  - Releases (All Finalized Plans, Files and Documentation) can be downloaded at: https://github.com/tgit23/AgIrrigationRemoteControl/releases
  - This document (most recent version) can be seen at: https://github.com/tgit23/AgIrrigationRemoteControl/blob/master/docs/AgIrrigationPumpRemoteControl.pdf
  - Pre-Built Purchasing options are available – See the projects main page link above

A. Signal Strength (2-Miles)

- ✔ Standard Signal distance is recommended for usage of up to 2-miles
  - Elevated or higher gain antennas can be purchased to extend this range
  - Test Scenario
    - Pump-Controller mounted 5-feet from the ground on the power pole next to the pump
    - Hand-Controller checked inside a moving vehicle (~ 3-feet off the ground)

- Test Environment
  - Houses located on square mile blocks; each mile having approximately 6 to 10 houses per mile
  - Ground level is fairly flat

- Results
  - < 1-mile - Coverage approximately 99% of spot locations (very few spots inside a house will loose signal)
  - 1-to-2 miles - Coverage approximately 90% of spot locations (a few spots inside a vehicle will loose signal)
  - 2-to-3 miles - Coverage approximately 5% of spot locations (very few spots get a signal)
  - 3-to-4 miles - A few signals were obtained at up to 3-3/4 miles away; but spot location was very very temperamental.

B. Features

- ✔ A solid rural 2-Mile range
- ✔ No cellular, FCC licensing or other requirements are needed; Just hook it up and use it
- ✔ A single Hand-Remote can control multiple irrigation pumps
- ✔ Multiple Hand-Remotes can control the same irrigation pump
- ✔ Uses the Pump Panels "AUTO" selection; so it doesn't disturb manual operation of Hand-Off & Start sequences
- ✔ Implements a "Stay-Off" after power losses that is required by power companies
- ✔ Easily expandable to offer Pressure and Water-Level monitoring

- ✔ Hand-Remote settable Alarms (buzzer) to notify user of
  - Power changes - For Example; Alarms when pump shuts down due to a power loss
  - Water Pressure boundaries (if equipped) - For Example; Alarms when pressure drops due to a hose that has fallen off or broken pipe
  - Ditch Water-Level boundaries (if equipped) - For Example; Alarms when water in the ditch is too low or too high

2. INSTALLATION INSTRUCTIONS

This section is a guide to installing the PUMP-CONTROLLER using the INSTALLATION KIT. If these products were not purchased they can be built by following the #4.BUILDING INSTRUCTIONS|outline listed in this document.

- ✔ Installation will require the following **SUPPLIES**

  - Pump-Controller *(The white box shown on the right in the picture below)*
    - Attach the Antenna to the Pump-Controller
    - Verify a the Pump-Controller has a 4-Pin male plug ending

  - Installation Kit *(Including the following items)*
    - A 480VAC 3-Phase to 24Vdc Power Supply attached to a DIN mounting rail
    - A 480VAC or higher voltage Solid State Relay (SSR) with less than 5Vdc activation attached to a DIN mounting rail
    - 12-feet of RED 600V 16-Gauge or heavier THHN or MTW wire
    - 4-feet of GREEN 600V 16-Gauge or heavier THHN or MTW wire
    - A female EXIT PLUG with 4-feet of 600V Orange(+Vdc), Brown(-Vdc), Blue(Control) leads attached
    - (2) #10-32 x 3/4" Machine Screws

- ✔ The following **TOOLS** will also be needed

- A Voltmeter that reads up to or beyond 500VAC
- A Cordless drill
- A Center tap
- #21 Drill Bit for tapping a #10-32 Machine Screw hole
- #10-32 Threading tap
- Protective insulated and flash protection personal gear
- Wire strippers and cutters
- Phillips and Small flat head screw drivers

A. Electrical Panel Introduction

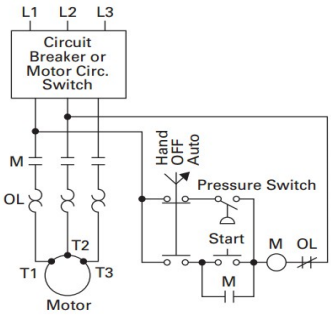
Read through the following section to gain general knowledge on the items within the pumps electrical panel.

Most 3-phase irrigation pumps that do not have variable frequency drives(VFD) or inverters are setup similarly. Below is the details of a pumps electrical 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.

Type 3R Industrial Pump Panel—C440 SSOL

Wiring Diagram

Typical Wiring Diagram



✔ Wiring Diagram Description

- **L1, L2, L3** - Are the three HOT wires of the 3-Phase Power (480VAC) going through the main breaker
- **M** - The Starter Contactor (example picture below) contains:
  - Activation Coil - Represented by the **M-Circle** in the wiring diagram just right of the Start Button (this Activates the "Hammer" Switch)
    - The Activation Coil resides "inside" the "Starter Contactor" Assembly; but can be removed by disassembling the unit.
  - Main Contacts - The **M-Contacts** (-||-) that connects the L1,L2,L3 lines → OL (Overloads) → Motor (Sometimes called the "Hammer")
    - The Main Contacts are significant part of the "Starter Contactor" as a whole unit.

  - Hold Contact - Represented by the **M-Contact** (-||-) wired in parallel and just below the Start Button
    - The Hold Contact "LOCKS" the Activation coil ON when the Start-Button press.
    - The Hold (sometimes referred to as an Auxiliary Contact) clamps onto the side of the "Starter Contactor"
    - As shown in picture above as the clamped on the right unit with a Grey square center.

- **Hand, Off, Auto** Switch
  - In the wiring diagram the Hand-Off-Auto selector switch in the "Auto" position where it is then connected to a Pressure Switch
  - The Solid State Relay that is part of the Installation kit will be attached where this "Pressure Switch" is shown in the diagram.

✔ Description of Operation

*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) Activation 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) activation 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.*

- ✔ Checking Amperage Limitations
  - Activation Coil Amperage can be gotten by looking up the "Starter Contactor" specification sheet, for example;
    - 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
    - The Solid State Relay in the Installation Kit can supply up to 10A; So it is well equipped to handle much larger Starter Contractors.

- ✔ Links to more information
  - http://www.eaton.com/ecm/idcplg?IdcService=GET_FILE&allowInterrupt=1&RevisionSelectionMethod=LatestReleased&noSaveAs=0&Rendition=Primary&dDocName=998056282226
  - http://www.eaton.com/Eaton/ProductsServices/Electrical/ProductsandServices/AutomationandControl/EnclosedControl/NEMA/PumpPanels/index.htm

B. DIN-Rail Mounting

BE SURE THE MAIN BREAKER IS TURNED OFF AND CAREFULLY CHECK THAT THERE IS NO VOLTAGE ON THE OUTPUT OF THE MAIN BREAKER WITH A PROPER METER!!! 480VAC is deadly; and can cause severe flash burn and/or death. (e.g. A meter on the wrong setting can "flash" explode causing "flash" burns on you even at a distance away). I DISCLAIM ANY AND ALL RESPONSIBLITY FOR ANY RISK OR HARM OCCURED BY FOLLOWING ANY INSTRUCTION IN THIS MANUAL.

See https://www.youtube.com/watch?v=6hpE5LYj-CY

1. Check your volt meter
   a) Before leaving to install; check that your volt meter is working by plugging it into a household outlet (Use the same setting >500 VAC)
   b) Make sure the probes are plugged into their correct spots on the meter
   c) Do not make any more changes to the volt meter after testing its correct operation.

2. Follow safe practices
   a) Never turn ON the main breaker with the panel open
   b) Never stand in front of the panel while turning on the main breaker (flash explodes will often blow the front panel clear off) - stand at the side
   c) Never work in the same side of the box where the live wires (feed wires) exist – generally at the top of the box
   d) Always stand as far away as you can

3. Prepare for Installation
   a) Switch the Main Breaker to OFF from the outside of the pumps electrical panel
   b) Open the pumps electrical panel
   c) With leather gloves on, very carefully making sure not to wiggle the probes out of place, measure the following voltages
      - From the bottom of the Main Breaker; Measure voltage between each of the 3-connectors (They should all be 0V)
      - From the bottom of the Main Breaker; Measure voltage between each of the 3-connectors and the ground metal of the electrical panel

4. Plan a location inside the Electrical Panel for the Power Supply and SSR mounted on the DIN rail
   a) Requires approximately a 6" Wide x 6" Tall x 6 1/2" Deep envelope
   b) Find a location as far away from other hot wires as possible (eliminate the risk of a loosened wires falling and touching a low-voltage wire)
   a) Preferably a location at the bottom of the box where the exit plug can quickly drop out of the box.

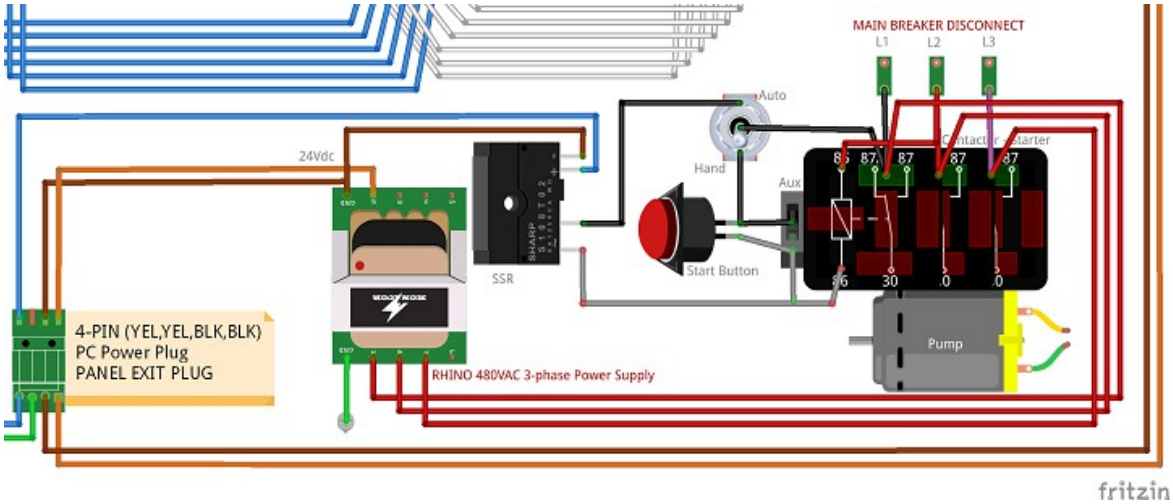
5. Mount the DIN Rail
   a) Hold the DIN Rail in the desired location and punch the metal through easily accessible mounting slots in the DIN Rail
   b) Drill two holes in the punch markings (Bit Size #21 – 5/32" - 0.1590")
   c) Tap-thread the two holes just drilled (#10-32 threading tap)
   d) Hold the DIN rail in place and fasten it with (2) #10-32 machine screws

6. Attach (If taken off) the Power Supply and Solid State Relay to the DIN Rail

Illustration 3: ***Note****; the Fulree Converter pictured IS NO LONGER USED*

C. Pump Panel wiring

NOTICE: All wiring in this section should be done with 600V rated wire

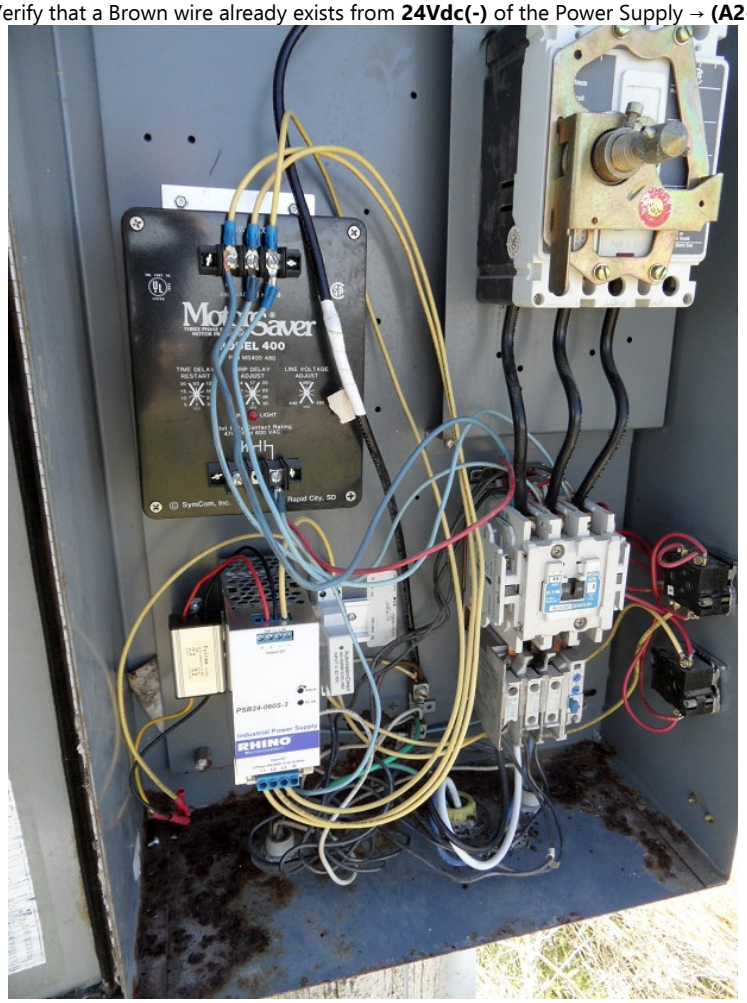


1. Wire the 480VAC 3-Phase Power Supply attached to the DIN Rail
   a) Connect Power Supply Terminals **L1, L2, L3** to the top terminals of the Contactor (i.e. Starter / Hammer) L1, L2, L3 terminals
      • Often this is easiest to do by wiring to a “Motor Saver” unit that is already wired to L1, L2, L3 if your panel has one

   b) Connect Power Supply Ground (⏚) to a bolt or terminal connected to the panels back-plane

2. Wire the Solid State Relay attached to the DIN Rail
   a) Connect Solid State Relay (15) → **AUTO** Terminal of the HAND-OFF-AUTO switch (*Typically the only terminal with no wire connected*)
   b) Connect Solid State Relay (16) → **OPPOSITE side** of the start button as the wire from the HAND-OFF-AUTO to the Start Button

3. Wire in the female **Exit Plug**
   a) With the **Female end** of the EXIT PLUG outside the box; insert the wires through a bottom hole in the Pumps Electrical Panel
   b) Connect the
      - **Orange wire** → **24Vdc(+)** of the Power Supply
      - **Brown wire** → **24Vdc(-)** of the Power Supply
      - **Blue wire** → **(A1+)** of the Solid State Relay
   c) Verify that a Brown wire already exists from **24Vdc(-)** of the Power Supply → **(A2-)** of the Solid State Relay



4. Close the Electrical Panel – Panel Control installation is now complete
   a) Wait until after installing the Pump-Controller before turning the breaker back on.

D. Pump-Controller

1. Plug the Pump-Controller into the EXIT PLUG installed in the steps above.
2. Attach the Pump-Controller box to the power pole using a wood screw at the desired height up the pole.

E. Testing

1. Turn the Electrical Panel Breaker ON.

2. Switch the Hand-Off-Auto switch to AUTO.

3. Using the HAND-REMOTE Device
   a) Find the Menu-Item “Power” for the Pump-Controller Device just installed
   b) Press (→) to 'SET->' and then UP/DOWN to set to ON
   c) Press the square 'Enter' button – The pump should start running
   d) While still on the “Power” Menu-Item; Press (→) till 'SET->' and then UP/DOWN to set to OFF
   e) Press the square 'Enter' button – The pump should shut off

3. OPERATING INSTRUCTIONS

A. Introduction

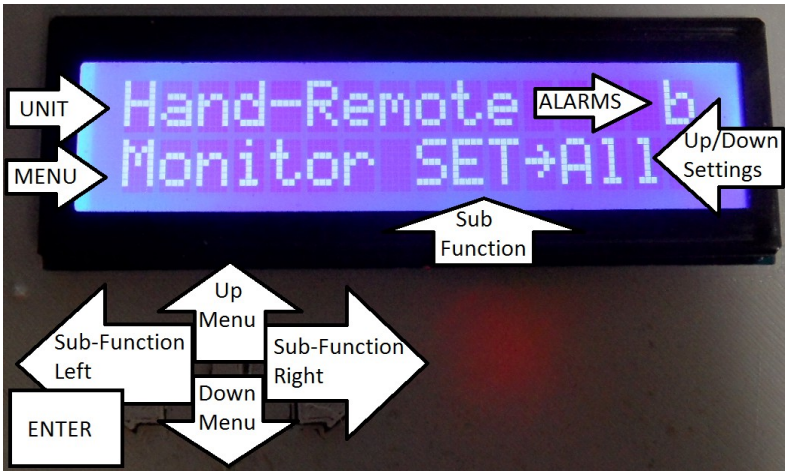
The HAND-REMOTE display has a **Top Row** and a **Bottom Row**

✔ **Top Row** (i.e. The Status Row – not changed directly)

- On the Left - The Device (Unit) that is currently under control or being monitored
- On the Right - The Alarms currently active (Alarm 'ID'entifying letters)

✔ **Bottom Row** (i.e. The Control Row – changes are made directly on this row)

- On the Left - The current Menu-Item (*Up-Down Arrows Change Item*)
- In the Middle - The current Action/Function to be applied to the Menu-Item (Right-Left Arrows Change the Function)
  - If it is (=) - Function; **Read** Value for the current Menu-Item (This can also be thought of as the "Main Menu")
  - If it is (SET→) - Function; **Set** a New Value for the current Menu-Item
  - If it is (<!) - Function; **Low Alarm** value; A "too-low" numerical value that will trigger an alarm
  - If it is (=!) - Function; **Low Alarm** option; A "is-equal-to" option that will trigger an alarm
  - If it is (>!) - Function; **High Alarm** value; A "too-high" numerical value that will trigger an alarm
  - If it is (≠!) - Function; **High Alarm** option; A "not-equal-to" option that will trigger an alarm
- On the Right - The function Value (of or for) the Device (Up-Down Arrows Change the Read, Set, or Alarm VALUE)



B. Display Table

✔ Maneuvering the display items can be represented as a table like the one shown below

- The "Location" Column is NOT part of the Display Menu; It shows where the item is **wired** to the Pump-Controller Hardware (if applicable)
- Disabled items are Skipped when maneuvering through the display table

		SUB-FUNCTIONS (←LEFT || RIGHT→)				Location
	Device	MENU ITEMS (=) Up/Down Items ENTER to refresh	SET (→) Up/Down Changes ENTER to Apply	LOW ALARM (=!) (<!) Up/Down Changes ENTER Toggles On-Off	HI ALARM (≠!) (>!) Up/Down Changes ENTER Toggles On-Off	Wire to PIN (NOT for Display)
Menu-Items (UP || DOWN)	Hand-Remote	Battery(B)	DISABLED	(<!) When less than	(>!) When greater than	A1
	Ditch Pump	Power(P)	(SET→) On (SET→) Off	(=!) Equals "On" (=!) Equals "Off"	(≠!) Not Equal "On" (≠!) Not Equal "Off"	D7
	Ditch Pump	Water (L)	DISABLED	(<!) When less than	(>!) When greater than	V64(Trig=D4,Echo=D5)
	Ditch Pump	Pressure(R)	DISABLED	(<!) When less than	(>!) When greater than	A3
	Canal Pump	Power(P)	(SET→) On (SET→) Off	(=!) Equals "On" (=!) Equals "Off"	(≠!) Not Equal "On" (≠!) Not Equal "Off"	D7
	Canal Pump	Pressure(R)	DISABLED	(<!) When less than	(>!) When greater than	A3
	Canal Pump	Pressure(S)	DISABLED	(<!) When less than	(>!) When greater than	A4

** Note: ↑V64 is Virtual Pin #64; Virtual Pins are memory spaces inside the Pump-Controller and therefore the Pump-Controller firmware actually identifies the connecting pins. This is used when Pump-Controller processing is done before a legitimate value can be determined.*

C. Alarms

✔ Alarms check read/monitored values against desired values (Low / High boundaries) and notifies the user (buzzing) if boundaries are violated

- Some menu items will have alarms while others will not – this is configured in the firmware and should be on a use-case scenario
- Anytime a button is pressed an active alarm (buzzing) will quite until "idle-monitoring" is started again (No button press for 30-seconds)
- To permanently shut off the active alarm (buzzing) the alarm itself must be turned 'OFF'

✔ Alarms can be either turned ON or OFF
- ON alarms are identified by the single character in the Top-Right of the Display
- Alarms are switched from ON → OFF or OFF → ON by
  - Press UP/DOWN until the Menu Item you'd like to turn an Alarm On/Off for - is selected
  - Press RIGHT button to select which alarm to turn On/Off; Either the Low Alarm (!=, !<) or the High Alarm (! ≠, !>)
  - Press ENTER button to Toggle the Alarm On/Off (An On-Alarm goes Off; An Off-Alarms goes On)

✔ To Set an alarm **value**
- Press UP/DOWN until the Menu Item you'd like to set an alarm boundary for is selected
- Press the RIGHT button as many times as it takes to see one of the following indicators in the Middle of the Bottom Row
  - (=!) - Sound an alarm if the value becomes EQUAL to the one indicated
  - (≠!) - Sound an alarm if the value does NOT-EQUAL equal the one indicated
  - (<!) - Sound an alarm if the value becomes LESS-THAN the one indicated
  - (>!) - Sound an alarm if the value becomes GREATER-THAN the one indicated
- Use the UP/DOWN buttons to select the comparison value or option
  - For Numerical values; Holding the UP/DOWN button for more than 5 counts will increase or decrease the value by 10
- Press ENTER button to apply the selected comparison value (Note; Applying a value will also Toggle the On/Off status of the Alarm)

✔ Notes
- When the Hand-Remote looses signal the alarms are shut-off automatically until signal is restored
- The Battery alarm will disable at any reading under 550. This prevents the battery from alarming while being powered by USB.

4. BUILDING INSTRUCTIONS

This section shows how the Hand-Remote, Pump-Controller and Installation Kit are built. It can be used to build your own setup by purchasing the individual pieces and assembling them yourself. It can also be used for troubleshooting problems or issues that may arrise.

4.1 BILL OF MATERIAL (BOM) - $242.66

Itemized Pricing

- ✔ Installation Kit (High-Voltage) = **$119.67** / Per-Pump-Controller
  - (1) RHINO 480V 3-Phase to 24VDC Power Supply = **$59.00**
  - (1) 600VAC Solid State Relay N.O. Random = **$24.50**
  - (50-feet) Outdoor data cable = $11.50
  - (25-feet) 600V Wire 16AWG+ = $ 9.18
  - (6") DIN Rail (Sold in 1-Meter / (3)) = $ 8.00
  - (Various) Solder-less Crimp Connectors = $ 6.02
  - (1) Computer Power Connector Plug = $ 0.99
  - (1) 24Vdc to 9Vdc Step-Down Buck Converter = $ 0.48

- ✔ Pump-Control = **$62.58** / Per-Pump-Controller
  - (1) Xbee 900hp Pro RF Module **= $39.00**
  - (1) High Gain Antenna = $ 6.66
  - (1) Arduino UNO R3 w/Male Headers = $ 3.80
  - (95.2) 3D PLA Plastic Filament ($0.03/cm3) = $ 3.87
  - (1) Zigbee Shield = $ 2.20
  - (15) Screw Terminal Blocks = $ 2.15
  - (1) 50x70cm Perfboard = $ 1.56
  - (1) Right-Angle 9VDC Power Plug = $ 1.76
  - (2) 4-PIN Female DuPont Pin Headers = $ 0.79
  - (2) 6-PIN Female DuPont Pin Headers = $ 0.79

- ✔ Hand-Remote = **$60.41** / Per-Hand-Remote
  - (1) Xbee 900hp Pro RF Module = **$39.00**
  - (1) High Gain Antenna = $ 6.66
  - (1) Arduino UNO R3 = $ 3.80
  - (83.8) 3D PLA Plastic Filament ($0.03/cm3) = $ 2.82
  - (1) Zigbee Shield = $ 2.20
  - (2) Keypad / LCD Shield = $ 2.19
  - (2) Resistors = $ 2.28
  - (1) ON/OFF Rocker Switch = $ 0.86
  - (1) Active Buzzer = $ 0.60

A. Pump-Panel Components ($99.50)

(1) RHINO switching Power supply
320-600 VAC input, 3-phase
24 VDC output, 60W, 2.5A
35mm DIN Rail mountable

AutomationDirect.com ($59.00/ea)

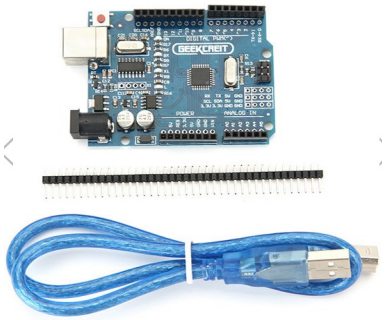
6" of DIN rail
Slotted, 35mm, 10 mm height
For mounting Power Supply and SSR of each Pump

AutomationDirect.com ($8.00/2x 1-Meter Pieces)

(1) Solid state relay
3-32 VDC input
480VAC or better SPST, N.O., SCR, 10A
35mm DIN-rail, Random type

AutomationDirect.com ($24.50/EA)

B. Electronic Control ($25.18)



(2) UNO R3 Atmega328P Development Board with Male Pin Header and USB Cable

AliExpress.com ($3.80/EA)
(1) for each HandRemote
(1) for each PumpController

(1) Keypad Shield Blue Back-light for Arduino Robot LCD 1602 Board

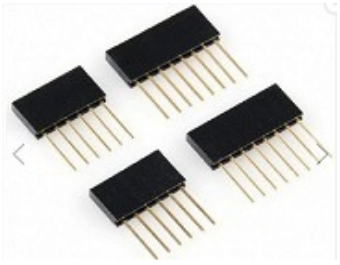
AliExpress.com ($2.19/EA)
(1) Display for each HandRemote
Wiki
Schematic
Layout

(2) Zigbee Shield RF Wireless Module Expansion Board for Arduino Xbee

Aliexpress.com ($2.20/EA)
(1) for each HandRemote
(1) for each PumpController
Schematic
Tutorial

(15) 2x Pin Plug-in Screw Terminal Block Connector 5.08mm Pitch External I/O connections

Banggood.com ($2.15 / 20pcs)
(15) for each PumpController

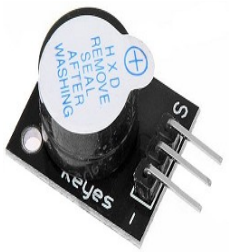


(1) 2-PIN Male – Zigbee → Perfboard POWER
(1) 4-PIN Female – Perfboard → Zigbee POWER
(2) 6-PIN Male – Zigbee I/O → Perfboard
(1) 4-PIN Female - Zigbee->Perfboard RX/TX

Aliexpress.com ($0.79 / 10pcs)
(2) Female, (2) Male for each PumpController
** Male Headers are part of Arduino Purchase **

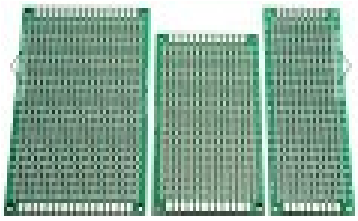
(1) Black Snap-in On/Off Rocker Switch

Aliexpress.com ($0.86 / 5pcs)
(1) for each HandRemote



(1) Active Speaker Buzzer Alarm

Aliexpress.com ($0.60 / ea)
(1) for each HandRemote



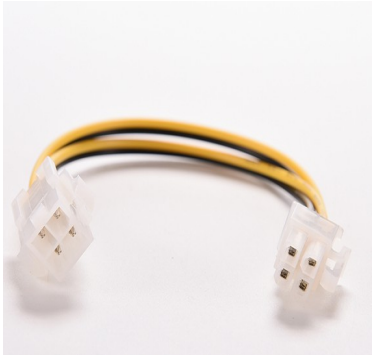
(1) FR-4 Double Side 50cm x 70cm Prototype PCB Printed Circuit Board

Aliexpress.com ($1.56/ 4pcs)
(1) for each PumpController

(2) Resistors
1 valued at 1/3 the other (10K-Ohms or higher)

Banggood.com ($2.28 / 300pcs)
(2) for each HandRemote

(1) Right-Angle 5.5x2.1mm Power Plug for Arduino Power
Aliexpress.com ($1.76 / 2pcs)
Amazon.com ($4.51 / 10pcs)



(1) ATX Computer power connector
Aliexpress.com ($0.48 / ea)

(1) LM2596 Power Supply Output 1.5V-35V DIY Part Modules
AliExpress.com ($3.27/ 5PCS)

C. RF Electronics ($91.32)

(2) Xbee-Pro 900Hp S3B Digimesh, 900Mhz, 250Mw, Rpsma Connector, 200Kbps

https://www.arrow.com/en/products/xbp9b-dmst-002/digi-international ($39.00/EA)
(1) for each HANDREMOTE and (1) for each PUMPCONTROLLER

(2) Antenna Helical 3dB Gain 900MHz

https://www.arrow.com/en/products/w1063/pulse-electronics-corporation ($6.66/ea)
(1) for each HANDREMOTE and (1) for each PUMPCONTROLLER

D. Wire & Miscellaneous ($26.70)

			
(~10-feet) 24-AWG outdoor direct burial wire Suggest Cat5e Amazon.com ($11.50 / 50ft) Enough to connect all accessory/power devices to the Pump-Controller	(25-feet) **600V** 14AWG THHN or MTW wire (~ 1-foot) 22AWG Solid Core wire Amazon.com ($9.18 / 25ft) Enough to connect RHINO P.S., SSR and Exit Panel Spec Sheet	(Various) Solder-less Terminal Connectors Banggood.com ($6.02 / 300pcs) Banggood.com ($9.69 / 100pcs) Any easy way to reliably connect wires	(2) #10-32 x 3/4" Machine Screws (5) #4 or smaller 1/4" Screws ($0.05/ea)

E. Tools Required ($NA)

- 3D-Printer (*may be optional if other casing solutions are available*)
- Soldering Iron
- Resin Core Solder
- Phillips Screw Driver

4.2 HAND-REMOTE

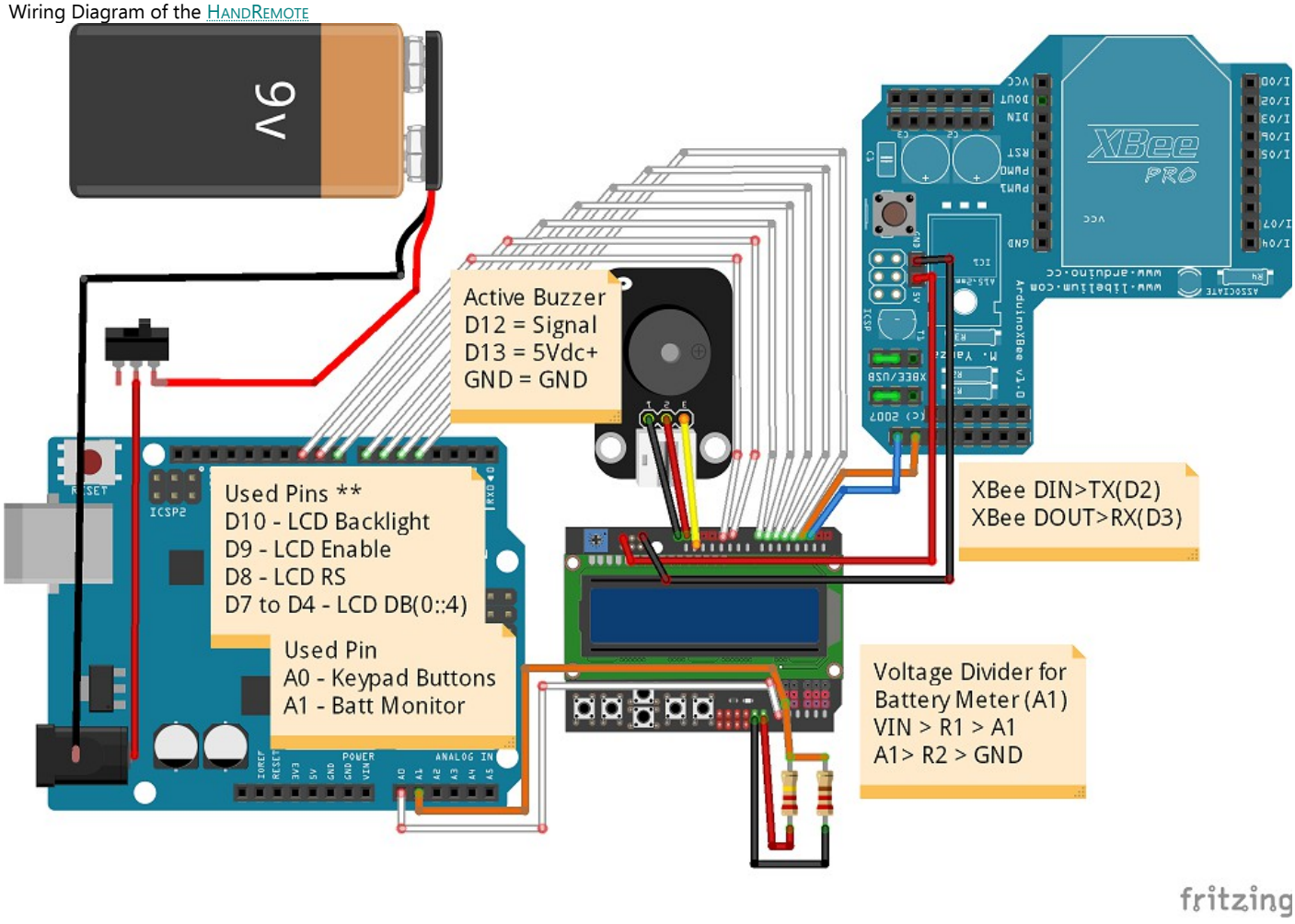
The HANDREMOTE is the portable remote control device used to control and monitor irrigation pumps remotely.

4.2.1 Electronics Assembly

- ✔ The following **Supplies** are needed to build the HANDREMOTE
  - ○ (1) Arduino UNO R3 Board
  - ○ (1) LCD Keypad Shield
  - ○ (1) Zigbee Shield
  - ○ (1) Xbee-Pro 900Hp S3B RF Module and High Gain Helical Antenna
  - ○ (1) 9Vdc Battery Connector
  - ○ (1) On/Off Rocker Switch
  - ○ (1) Active Buzzer
  - ○ (2) Resistors (One being 1/3 the value of the other; 100KOhm or larger)
  - ○ (1) 1" Black 22-Gauge Solid Core Hookup wire
  - ○ (1) 1" Red 22-Gauge Solid Core Hookup wire
  - ○ (4) #4 x 1/4" or smaller screws

- ✔ The following **Tools** are needed to build the HANDREMOTE
  - ○ Soldering Iron & Solder
  - ○ Wire Cutters / Dikes / Strippers
  - ○ Needle nose pliers

- ✔ Wiring Diagram of the HANDREMOTE



A. LCD Keypad Shield

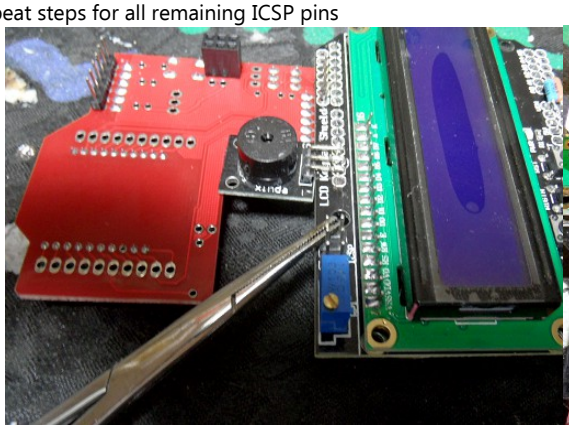
1. Wire the Battery-Voltage-Divider Resistors
   a) Select (2) Resistors (10K-ohm up to 2M-ohm values are recommended; Arduino A1 will draw 1.6nA)
      - R1 = Larger Value — Value for R1 Used = _____
      - R2 = 1/3 the Value of R1 — Value for R2 Used = _____

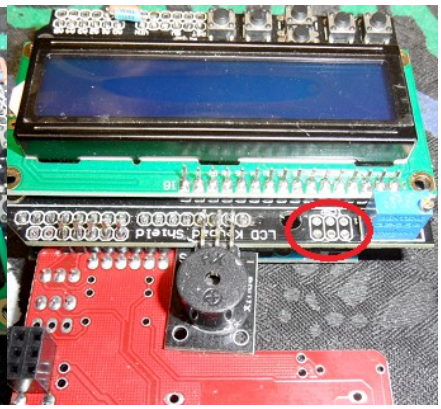
   b) Solder the Resistors on the top side of the LCD Keypad Shield
      - R1 = **VIN → A1**
      - R2 = **A1 → GND** (GND is at the Solder hole just left of VIN)

   c) 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. Remove the ICSP header pins on the LCD Keypad Shield
   a) Using Needle Nose pliers clamp onto an ICSP header pin on the front side of the Zigbee Shield
   b) Using Soldering Iron – Heat the solder side of the pin and pull the pin from the board
   c) Repeat steps for all remaining ICSP pins

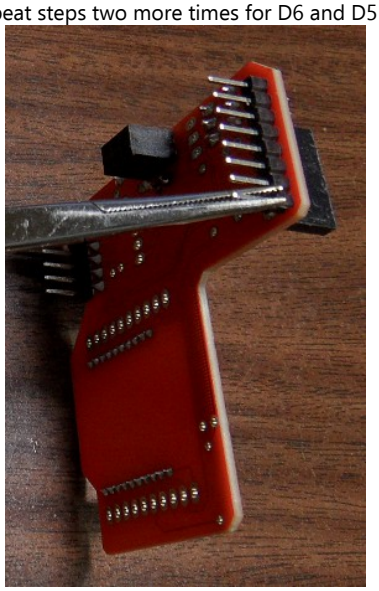




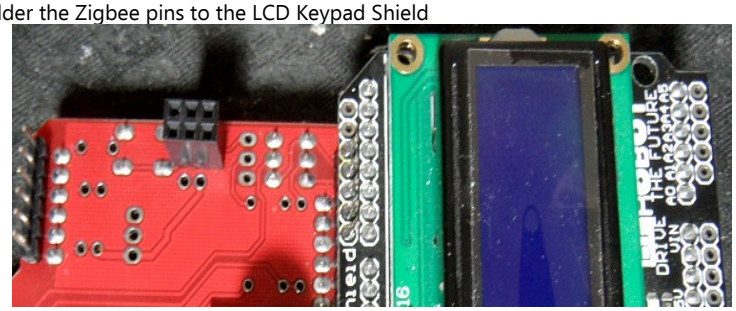
3. Plug the LCD Keypad Shield onto the Arduino UNO

B. Zigbee/Xbee Shield

1. Remove Header Pins **D7, D6, and D5** from the Zigbee Shield
   a) Using Needle Nose pliers clamp onto Pin D7 (shown in picture on 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
   c) Repeat steps two more times for D6 and D5 (moving left as D7, D6, D5)



2. Solder Zigbee Shield to the LCD Keypad (Flipped upside down & with Arduino UNO attached to the LCD Keypad Shield)
   a) Insert Zigbee Shield pins D0 → D4 (Remaining pins) into LCD Keypad Solder Pin-holes D2→D6
      - **Shifted towards center leaving 2-Pin-holes on the LCD Keypad open; as shown**
   b) Position the Zigbee Shield flat with LCD Shield and Up against the LCD Shield – Requires bending the header pins a little
   c) Solder the Zigbee pins to the LCD Keypad Shield

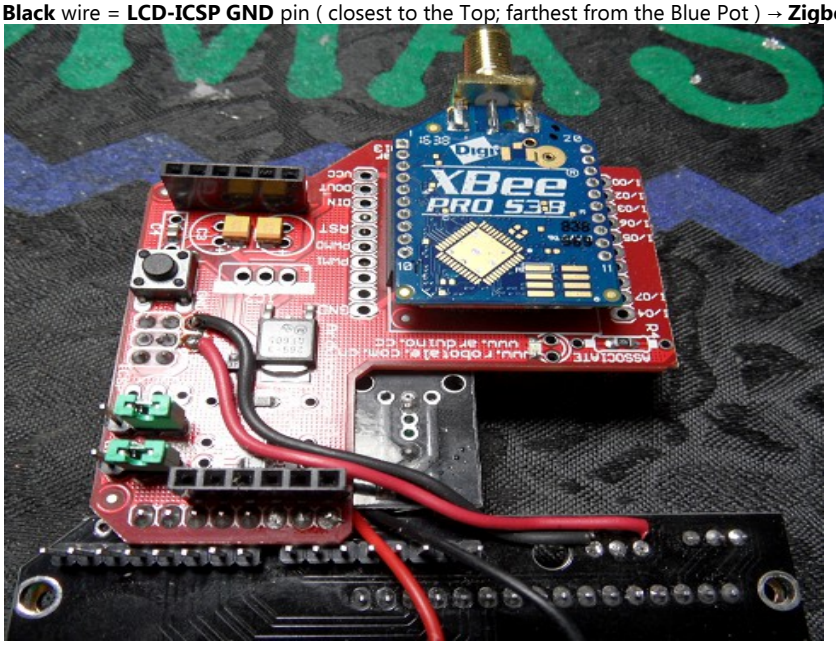


3. Wire Power to the Zigbee Shield

a) Flip over the assembly so the LCD Keypad Shield is backside and the Zigbee Shield is front side

b) Using two 22-Gauge Solid Core wire (Black & Red) Connect the LCD Keypad Shields ICSP pin header power pads to the Zigbee Shield

- **Red** wire = **LCD-ICSP 5V+** pin (closest to the Top; closest to the Blue Pot) → **Zigbee Shield (5V)** solder pad labeled as 5V
- **Black** wire = **LCD-ICSP GND** pin (closest to the Top; farthest from the Blue Pot) → **Zigbee Shield (GND)** solder pad labeled as GND



c) The green jumpers on the Zigbee Shield should be in the USB position (usually factory installed there) the picture shows these in the wrong place.

✔ Further Reading for the Curious

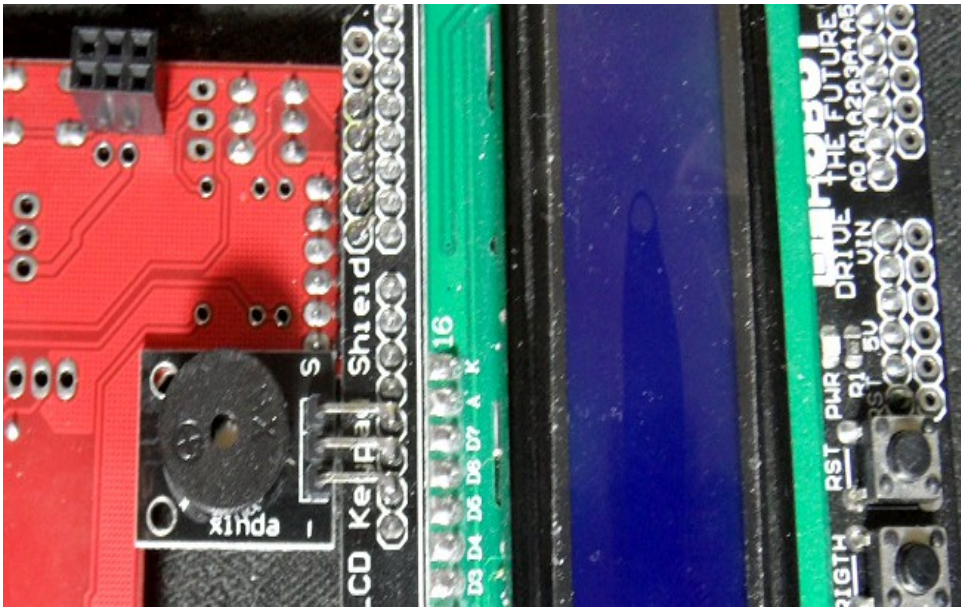
- https://www.arduino.cc/en/Main/ArduinoXbeeShield
- https://www.arduino.cc/en/Guide/ArduinoXbeeShield

C. Active Buzzer

1. Solder Active Buzzer to LCD Keypad Shield

a) Place active buzzer on edges of the Zigbee Shield as shown
b) Solder the active buzzer pins (S, +, -) to the pins as shown
c) The LCD Keypad pins (D12, D13, and GND) are determined according to the markings on the connected Arduino UNO.

- **'S' pin → D12**
- **'+' pin → D13**
- **'-' pin → GND**



2. Test the Assembly

a) Plug the Xbee Module into the Zigbee Shield

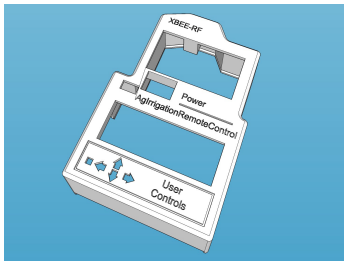
b) Plug in the USB cord to the Arduino UNO and verify
- The LCD screen lights
- The LED on the Zigbee Shield next to the word "ASSOCIATE" is blinking

c) Upload the Test.ino Firmware
- Check that the buzzer sounds after upload (a key-press will silence the buzzer)
- Check that the LCD Keys work by scrolling the menu
- Check the "Battery" meter – It should not read either 0 or 1023

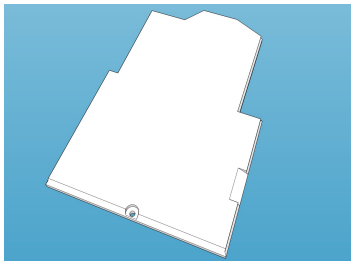
4.2.2 Hardware Assembly

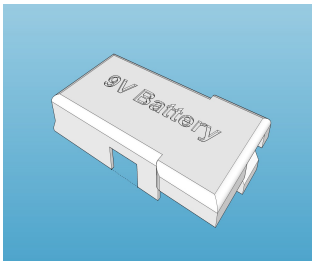
A. 3D-Prints ($2.82)

✔ Print the Following Models on a 3D-Printer
- Files Located at https://github.com/tgit23/AgIrrigationRemoteControl/tree/master/HandRemote/3D-Prints-STL
- Total Filament = 94-grams @ $0.03/gram ~ $2.82

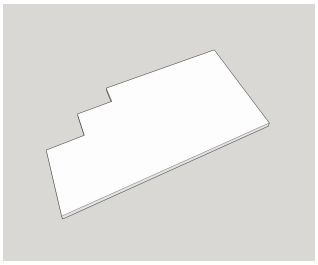


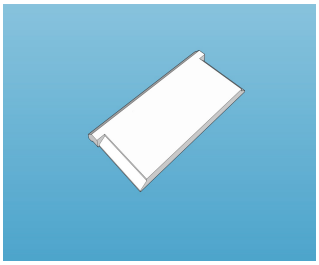
HandRemote-Case.stl (62g)

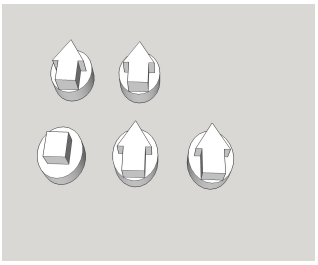

HandRemote-BackCover.stl (23g)



HandRemote-BatteryCover.stl (4g)

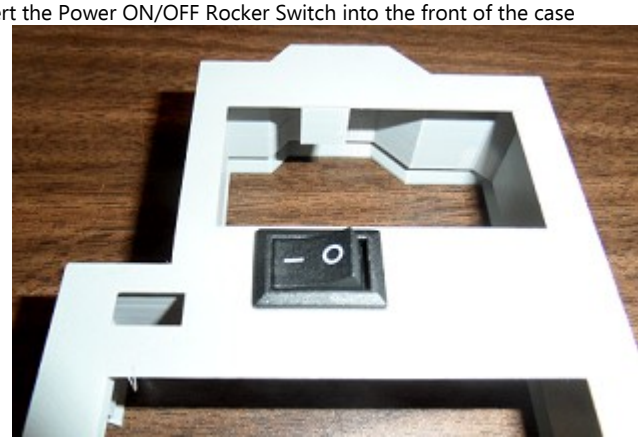

HandRemote-BatteryInsulatorPlate.stl (2g)


HandRemote-SideCover.stl (2.5g)


HandRemote-Buttons.stl (1g)

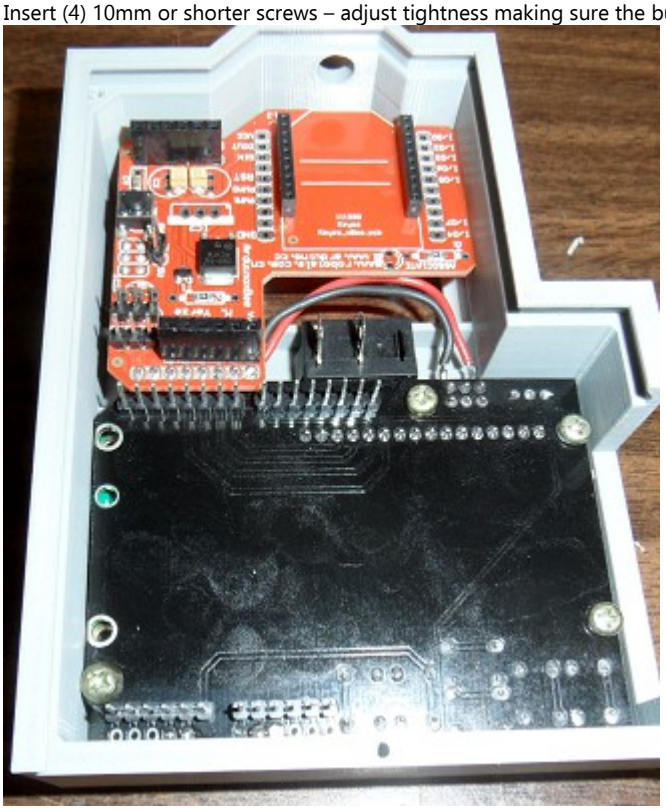
B. Install Electronics

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** 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*)



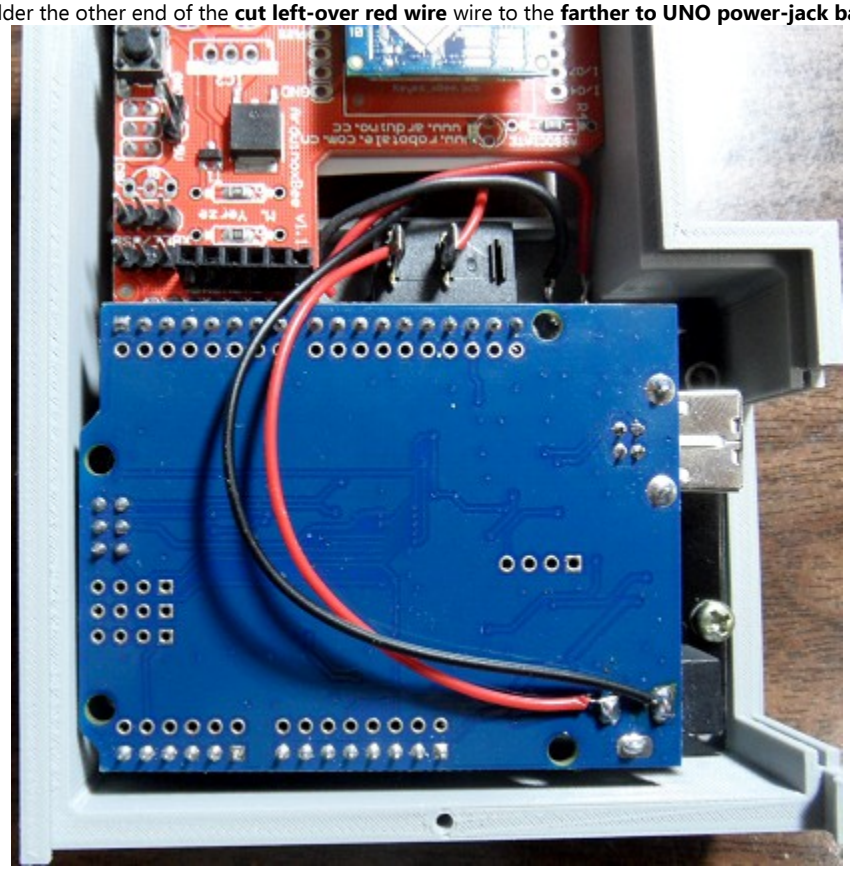
4. Plug the Arduino UNO back onto the LCD Keypad Shield

5. 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

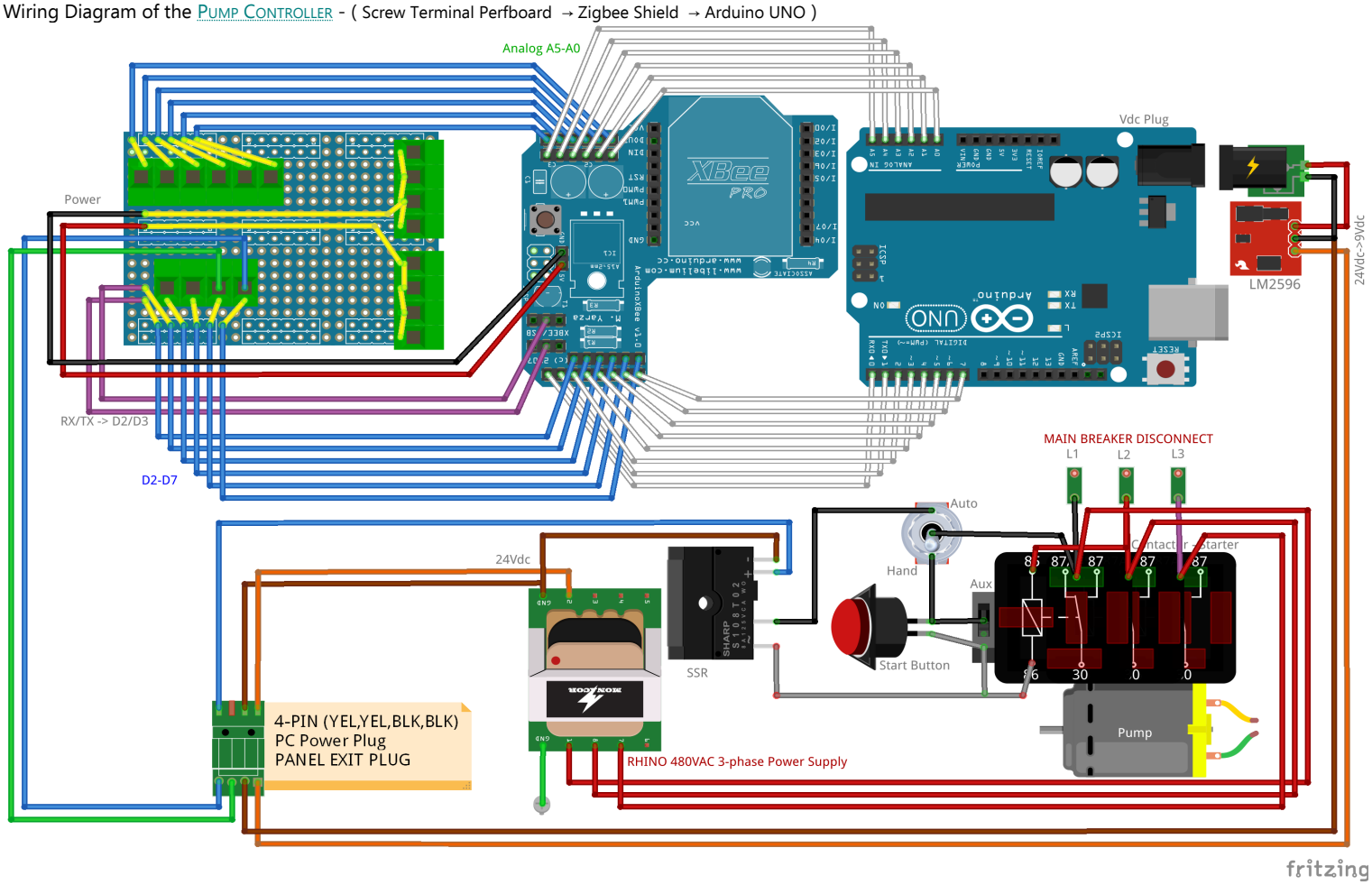
4. Attach Covers to Case
    a) Insert the HandRemote-BatteryCover.stl 3D-Print into the cases battery-hole and snap down
    b) Slide the HandRemote-BackCover.stl 3D-Print in from the bottom then insert a screw in the bottom-center
    c) Slide the HandRemote-SideCover.stl 3D-Print in the USB side until it snaps into place

4.3 PUMP-CONTROLLER

✔ The following **Supplies** are needed to build the PUMPCONTROLLER
- (1) Arduino UNO R3 Board
- (1) Zigbee Shield
- (1) Xbee-Pro 900Hp S3B RF Module and High Gain Helical Antenna
- (1) 50x70cm Perfboard
  - (9) 5.08mm Pitch Screw Terminal Blocks
  - (1) 2-PIN to 4-PIN Female DuPont Header
  - (1) 3-PIN or 4-PIN Female DuPont Header
  - (2) 6-PIN Male DuPont Headers
  - (1) 2-PIN Male DuPont Header
  - (18") 22 Gauge Solid Core Hook-up wire
- (1) Right Angle DC Power Connector
- (1) LM2596 DC-DC Buck Power Converter
- (10') Outdoor Burial data cable
- (1) ATX CPU Computer Power Cable
- (1) #4 or smaller screw
- (~12) Heat Shrinking tubes

✔ The following **Tools** are needed to build the PUMPCONTROLLER
- 24Vdc Desktop Power Supply
- Soldering Iron & Solder
- Wire Cutters / Dikes / Strippers
- Small flat screw driver

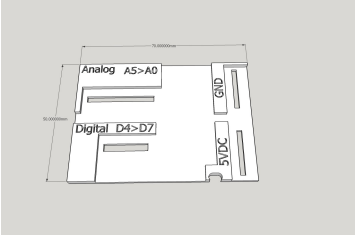
✔ Wiring Diagram of the PUMP CONTROLLER - (Screw Terminal Perfboard → Zigbee Shield → Arduino UNO)



4.3.1 Electronics Assembly

A. 3D-Prints ($0.20)

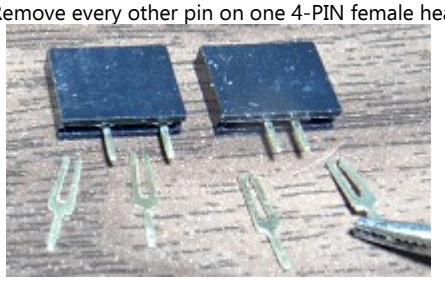
- ✔ Print the Following Model on a 3D-Printer
  - Files Located at https://github.com/tgit23/AgIrrigationRemoteControl/tree/master/PumpController/3D-Prints-STL
  - Total Filament = 6.3 cm^3 @ $0.03/$cm^3$ ~ $0.20



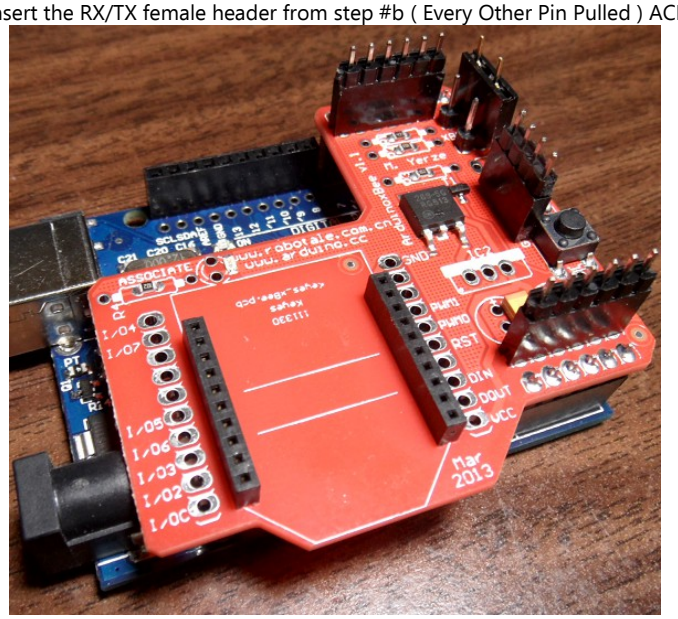
PumpController-PerfboardCover.stl (6.3 cm^3)

B. Perfboard Assembly

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)

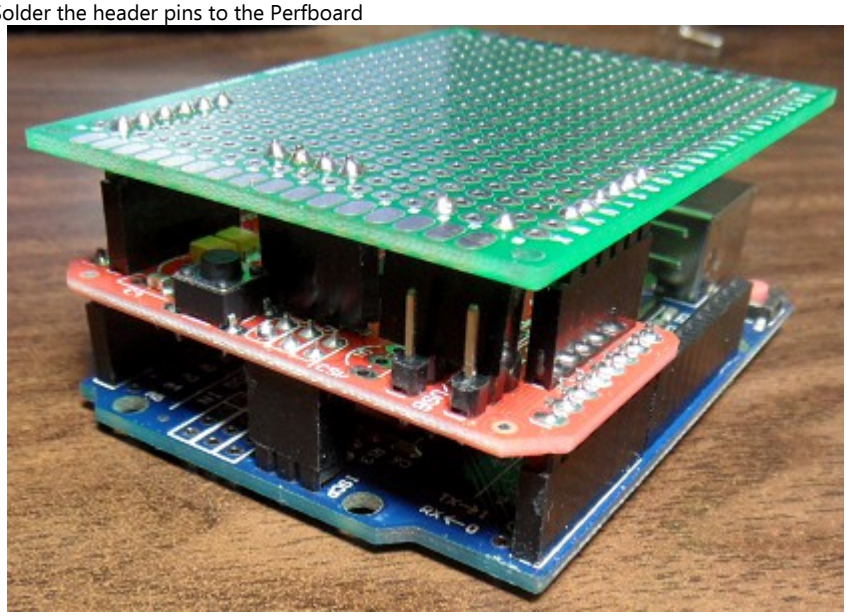


   c) Remove the (2) Green Jumpers on the Zigbee Shield
   d) Plug the Zigbee Shield onto the Arduino UNO
   e) Plug a 6-Pin Male Header into the Analog Header of the Zigbee Shield (A5 → A0)
   f) Plug a 6-Pin Male Header into the Digital Header of the Zigbee Shield (D2 → D7)
   g) Insert the POWER female header from step #a (Outer Pins Pulled) into the Zigbee Shield holes labeled 5V & GND (next to the reset button)
   h) Plug a 4-Pin Male Header into the top of the POWER female header
   i) Insert the RX/TX female header from step #b (Every Other Pin Pulled) ACROSS the two center male header (XBEE/USB) pins of the Zigbee Shield

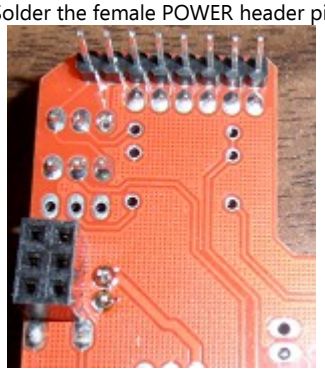


2. Solder the header pins to a 5cmx7cm 2.54mm pitch Perfboard
   a) Align and place the Perfboard onto the header pins

b) Solder the header pins to the Perfboard



3. Solder the Zigbee female POWER header pins
   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. Add wires to the header pins
   a) Cut (10) wires approximately 1" long and strip one end.
   b) Cut (2) wires approximately 3" long and strip one end.
   c) Gather (2) Pulled male pin headers (from ICSP Hand-Remote) or two more 1" wires

d) Attach (2) Pin Headers or Small wires from the Top-side of the Perfboard across RX/TX female pin headers and solder to D2, D3 Digital pins
e) Unplug the Perfboard from the Zigbee Shield
f) Insert (6) 1" wires from the Zigbee-side of the Perfboard into the holes just **below** the analog male pin headers, bend and solder
g) 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
h) 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

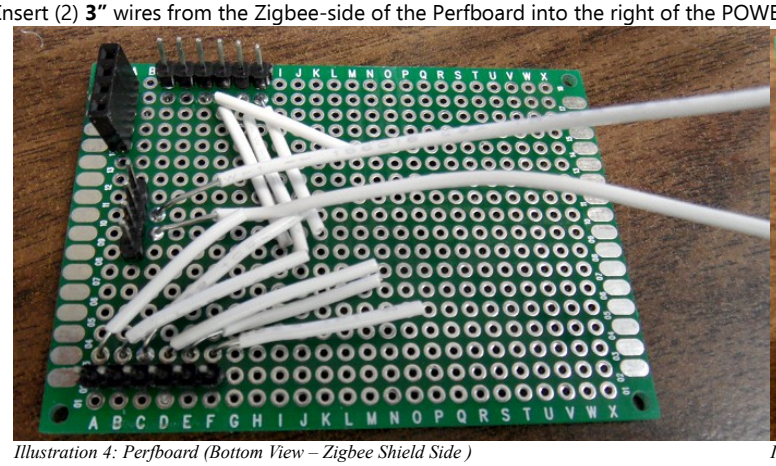


Illustration 4: Perfboard (Bottom View – Zigbee Shield Side)

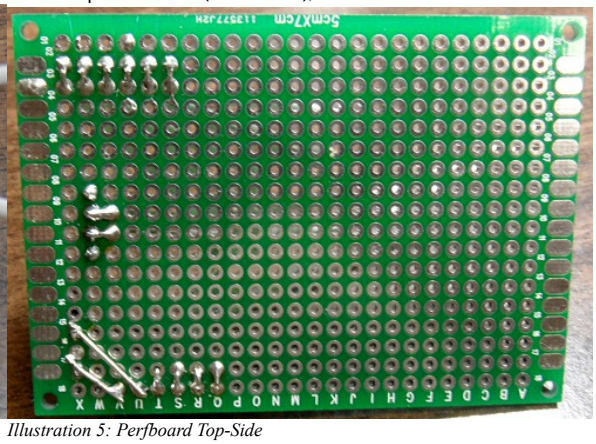
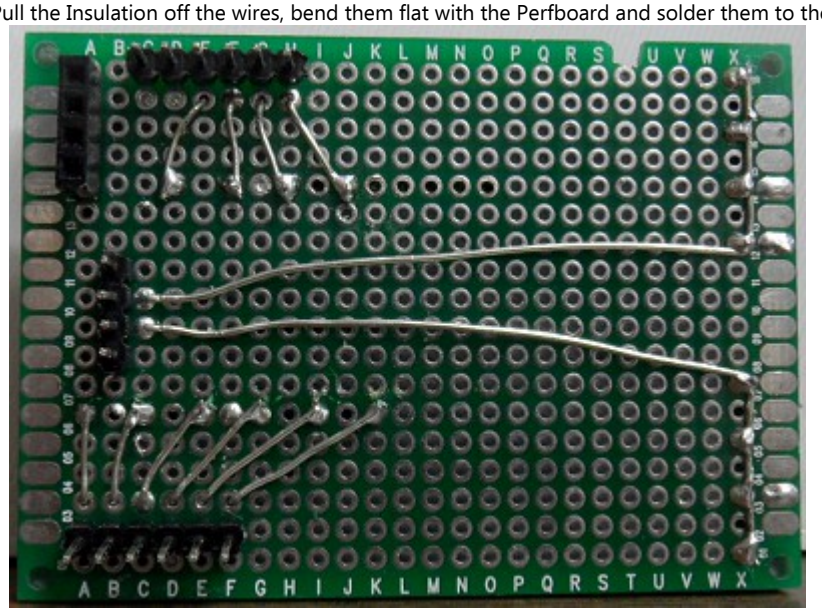


Illustration 5: Perfboard Top-Side

5. Install the Screw Terminals
   a) Place the HandRemote-PerfboardCover.stl 3D-Print on the top of the Perfboard
   b) Insert (3) 2x Screw Terminals 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)

   Note: If drilling out the holes is required; drill from the soldering side as to not dismount the solder-to pads

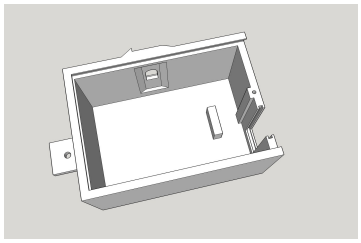
6. Solder the Screw Terminals to their corresponding header wires
   a) Turn the Perfboard over holding the screw terminals into place
   b) Sold each screw terminal pin to the Perfboard
   c) Pull the Insulation off the wires, bend them flat with the Perfboard and solder them to their corresponding pin (See picture or Wiring Diagram)

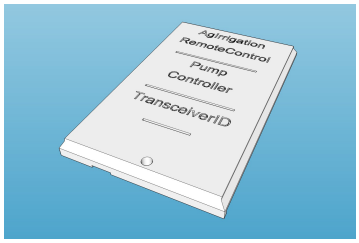


4.3.2 Hardware Assembly

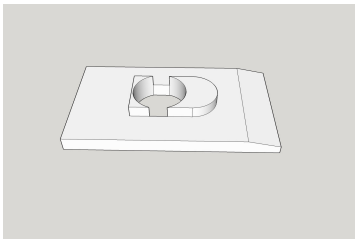
A. 3D-Prints ($3.87)

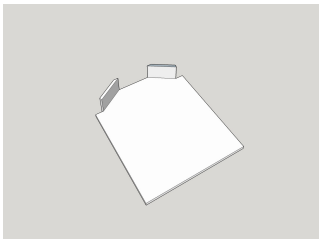
✔ Print the Following Models on a 3D-Printer
- Files Located at https://github.com/tgit23/AgIrrigationRemoteControl/tree/master/PumpController/3D-Prints-STL
- Total Filament = 129-grams @ $0.03/gram ~ $3.87

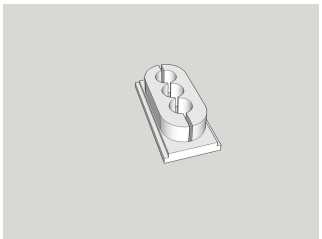

PumpController-Case.stl (98g)

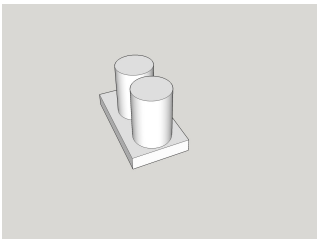


PumpController-Cover.stl (25g)


PumpController-AntennaCover.stl (1g)

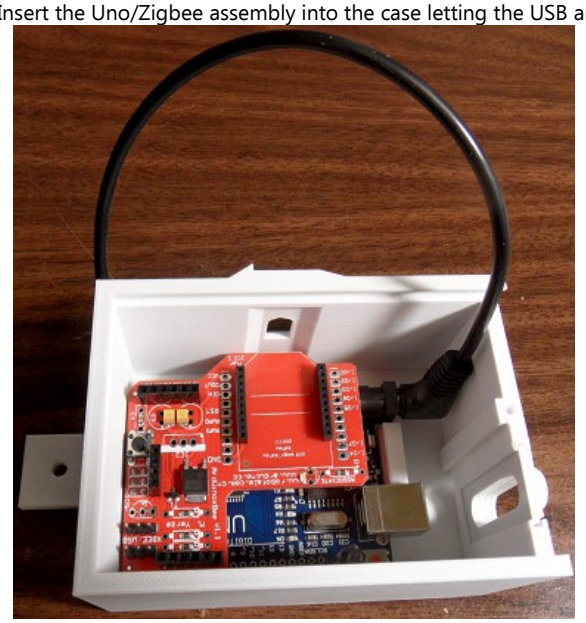

PumpController-XbeeInsulatorPlate.stl (1g)


PumpController-CableClampCover.stl (3g)

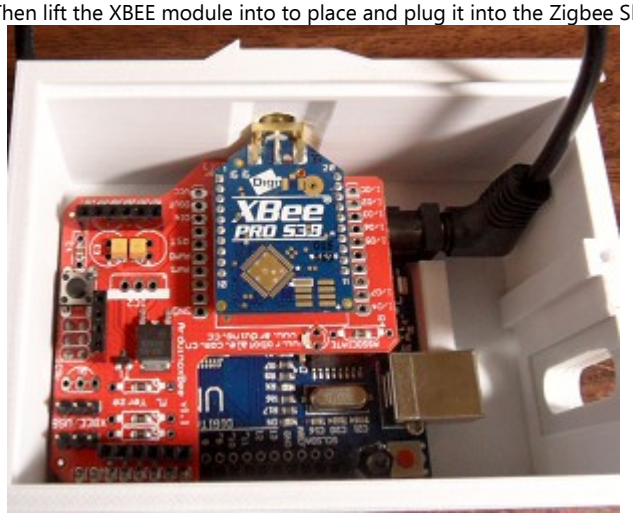

PumpController-CableClampCover-2HoleInsert.stl (1g)

B. Install Electronics

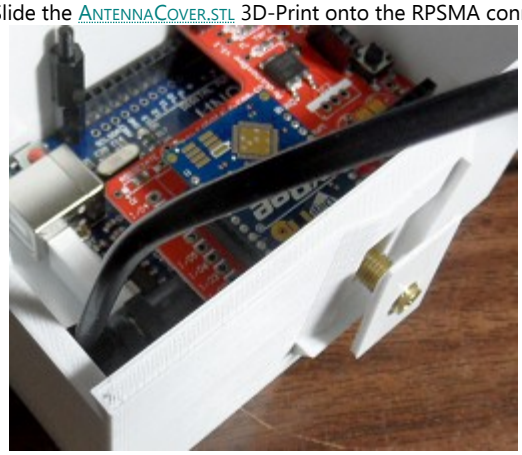
1. Install UNO and Zigbee into the Case
   a) Remove the Screw Terminal Perfboard if attached and connect the Arduino UNO & Zigbee Shield
   b) Insert the Uno/Zigbee assembly into the case letting the USB and power Jack sit on each side of the case stub.



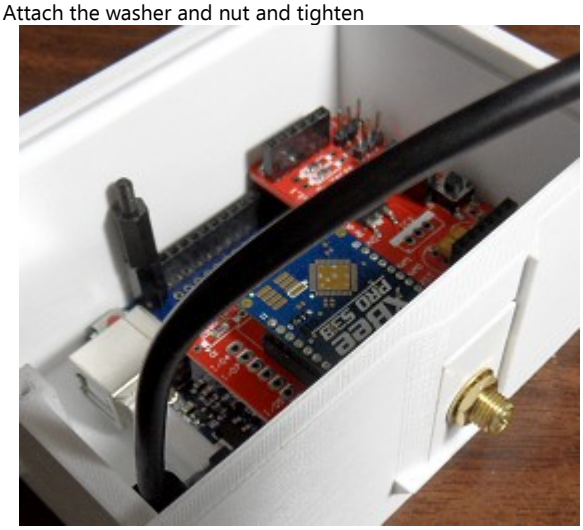
2. Install the XBEE module
   a) First slide the XBEE RPSMA Connector into the case slot
   b) Then lift the XBEE module into to place and plug it into the Zigbee Shields Xbee plug-in



3. Install the Xbee Antenna Cover
   a) Slide the ANTENNACOVER.STL 3D-Print onto the RPSMA connector from the outside of the case



b) Attach the washer and nut and tighten



C. Install Cable

1. Wire the LM2596-DC-Converter Module to one end of the cable

a) Cut a 10' length of Outdoor data cable with 3 or more leads
b) Strip off the outer insulation of one end

c) Cut a 10' length of Outdoor data cable
d) Strip off the outer insulation of one end approximately 4" back
e) Strip off the ends of the Orange and Brown twisted pairs just enough to fit in the LM2596 solder holes

- Solder the **Orange** twisted pair to the LM2596 Module pad labeled **IN+**
- Solder the **Brown** twisted pair to the LM2596 Module pad labeled **IN-**

2. Wire the Right-Angle 5.5x2.1mm DC Power Plug to the LM2596-DC-Converter Module

a) Cut the Right-Angle DC Power plug cords length to approximately 3"
b) Strip off the outer insulation approximately 1" back
c) Strip off the ends of the Red and Black wire just enough to fit in the LM2596 solder holes

- Solder the **Red Wire** to the LM2596 Module pad labeled **OUT+**
- Solder the **Black Wire** to the LM2596 Module pad labeled **OUT-**

3. Adjust the LM2596 Module to produce 9Vdc from 24Vdc

a) Clip on a **24Vdc+** desktop power supply to **IN+ and IN-**
b) Attach a Voltmeter to OUT+ and OUT-
c) Adjust the Blue Pot till the measured voltage is **9Vdc on OUT+ and OUT-**

4. Attach the ATX CPU Computer Power Plug to the other end of the cable

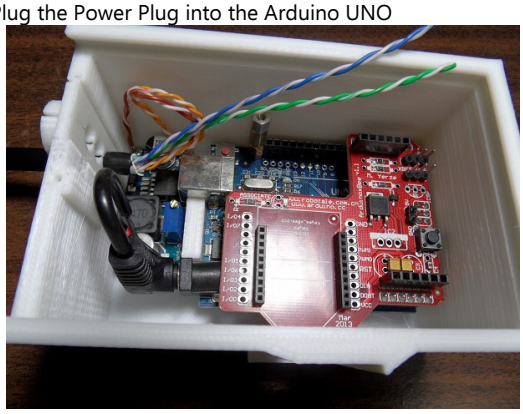
a) Cut the Male end of the ATX CPU Computer Power Plug leaving about 1" of Wire from the plug
b) Strip all the wires back about 1/4"
c) Slide on heat shrinking tube for each wire and the bundle of wires

d) With the Lock Tab on the plug On-Top and the Plug facing forward (wires facing back):

- Solder the **Left Yellow** to the **Orange** twisted pair
- Solder the **Left Black** to the **Brown** twisted pair
- Solder the **Right Yellow** to the **Blue** twisted pair
- Solder the **Right Black** to the **Green** twisted pair

5. Install the Cable into the PUMP-CONTROLLER Box

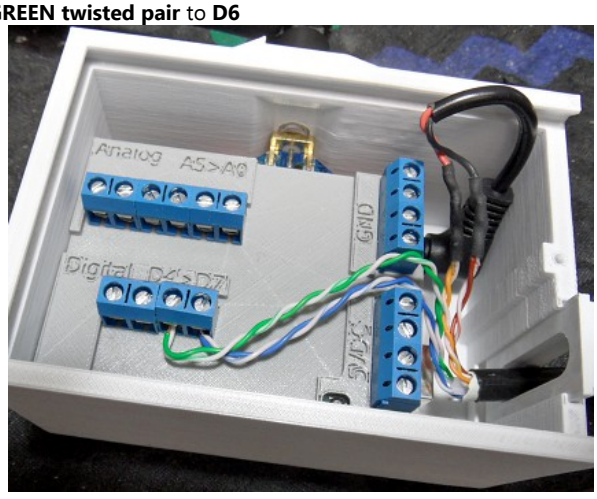
a) Insert the cable into the CABLECLAMPCOVER.STL 3D-Print and insert it with the LM2596 Module into the Case as shown
b) Plug the Power Plug into the Arduino UNO



6. Attach Perfboard and Wire to Cable

a) Place Xbee Shield
b) Carefully plug the Perfboard back onto the Zigbee Shield
c) Strip the ends of the BLUE/GREEN twisted pairs of the Cable and connect them to the Screw Terminals

- **BLUE twisted pair** to **D7**
- **GREEN twisted pair** to **D6**



4.4 INSTALLATION KIT

A. DIN Rail & Components

1. Prepare Din Rail

a) Cut a section of DIN Rail 6" long
b) Attach the Power Supply Unit
c) Attach the Solid State Relay (SSR)
d) Run a 600VAC 14-Gauge BROWN wire from **24VDC(-)** on top of the Power Supply to **A1** of the Solid State Relay

B. Pump-Panel Exit Cable

1. Create the Pump-Panel Exit Cable that will exit the Pumps Electrical Panel and connect to the PUMP-CONTROLLER

   a) With the **Female end** of the ATX CPU Computer Power Plug (Male side used in #4.3.2.3.Install Cable|outline)
   - Cut the wires approximately 1" from the plug
   - Install Heat Shrinking tube to the wires that will be soldered below
   - With the plugs notch (for male clip) On-Top and plug facing forward (wires facing back)
     - Strip and Solder the Plugs **Right Yellow** wire to 4-feet of **Orange** 600V 14AWG wire
     - Strip and Solder the Plugs **Right Black** wire to 4-feet of **Brown** 600V 14AWG wire
     - Strip and Solder the Plugs **Left Yellow** wire to 4-feet of **Blue** 600V 14AWG wire
     - Leave the Left Black wire
     - (Picture Coming Soon)

5. SYSTEM SETUP

✔ This section is a guide for
- ○ Configuring the Xbee RF Module, which for correct usage, requires a unique Network-ID
- ○ Uploading the proper Firmware onto the Hand-Remote and Pump-Controller's Arduino UNO
- ○ Details about changing the firmware to match users needs (Such as adding a pressure meter or water-level meter)

✔ This section only needs to be read if:
- ○ The project is built from scratch - In which case the firmware must be uploaded to both the Hand-Remote and Pump-Controller
- ○ The current setup has changed - If a digital pressure meter or ultrasonic meter (water-level) is added; firmware must represent those additions
- ○ Special operation is required - Experienced programmers can program the units to operate however they wish

5.1 Adding Expansions

1. Follow the steps in #5.2.Uploading Firmware|outline to Step #3 for the Hand-Remote firmware; Hold off on Step #4; "Compile & Upload"

2. Once the file is open in Arduino Sketch IDE; locate the following code-block which should be right at the top

```
//=============================================================================================================
//---------------------------- SIMPLE USER CONFIGURATION SETTINGS ----------------------------------------------
//=============================================================================================================
#define BUILD_VERSION                    20170622  // Release Version used to Build the Unit ( without the dots )
#define TRANSCEIVER_ID                   1         // Unique numeric (ID)entity for this Unit(1-15)
#define XBEECONFIG                       0         // Configure the XBEE using XCTU Digi Software by setting this to 1
#define ULTRASONIC_WATER_LEVEL_INSTALLED 0         // 0 = NO, 1 = YES ( Wire TRIG -> D4, ECHO -> D5 )
#define WATER_PRESSURE_INSTALLED         0         // 0 = NO, 1 = YES ( Wire SENSE -> A3 )

//^^^[ END - SIMPLE USER CONFIGURATION SETTINGS ]^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

3. As described above with the text following '//'
   a) Change ULTRASONIC_WATER_LEVEL_INSTALLED to '1' if a water-level sensor has been attached to the Pump-Controller.
   b) Change WATER_PRESSURE_INSTALLED to '1' if a digital pressure sensor has been attached to the Pump-Controller.

4. Now proceed to Step #4 in #5.2.Uploading Firmware|outline and Compile & Upload the changes just made

5.2 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) Plug in a USB cable from the Computer to the unit
   b) 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
   c) Select the Board; Sketch menu Tools → Board → Arduino/Genuino UNO
   d) 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

5.3 Xbee RF Configuration

The XBEE RF Module is setup from the factory to work without any additional changes. However, often it is very wise to assign non-factory setting in order to keep your RF communications from being interfered with by other XBEE devices in the neighborhood. Changes to the ID NETWORK ID (described below) will need to be preformed on all HAND-REMOTES and all PUMP-CONTROLLERS

5. Upload Firmware with XBEE Configuration Set
   a) To change the XBEE settings – Follow Steps #1 to #3 in #4.2.A SYSTEM OPERATIONS|Firmware|Uploading

      With the Firmware file open find the following line of code; which should be very close to the top

```
#define XBEECONFIG 0            // 1 to enter XBEE Configuration Mode
```

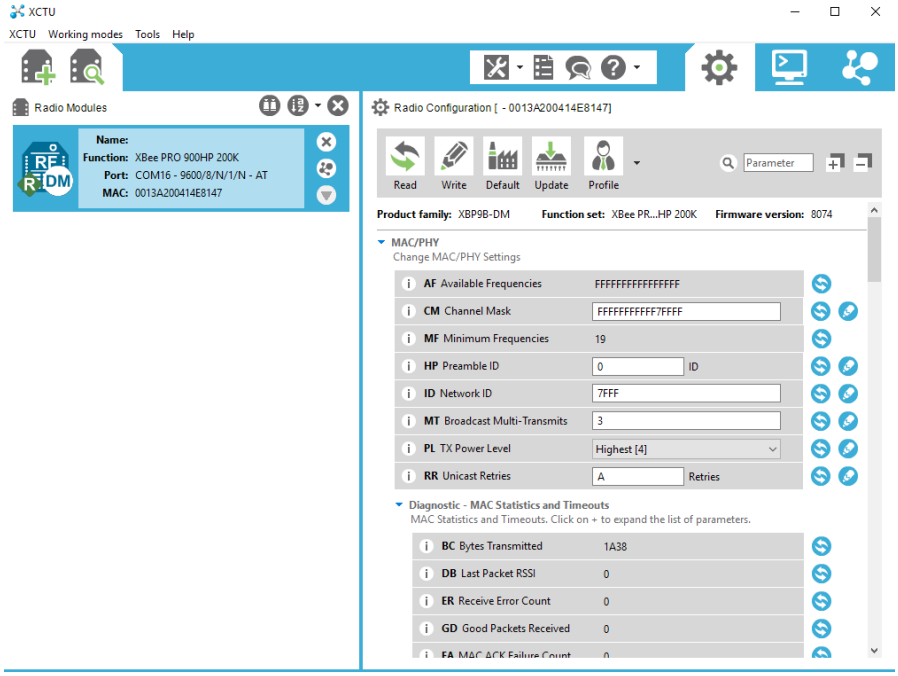
      Change it to

```
#define XBEECONFIG 1            // 1 to enter XBEE Configuration Mode
```

   b) Then proceed to upload the firmware to the Unit as outline in Step #4 in #4.2.A SYSTEM OPERATIONS|Firmware|Uploading

6. Install Digi-XCTU
   a) The Digi XCTU software can be gotten from https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu

7. Run Digi-XCTU



   a) Using the + (Add Radio Modules) mark in the top left corner enter the unit's COM port (The Arduino COM Port)
   b) The main values that may be of interest are:
      - ID Network ID - This sets a Unique "Channel" to keep your units communications separate from any others out there (Must be same on all units)
      - PL TX Power Level - If devices are close; a lower power setting can preserve battery life

8. Save Settings to the Xbee and Reload Firmware
   a) Be sure to click the "Write" button in XCTU to save any settings that were made
   b) Close XCTU
   c) In Arduino Sketch IDE restore the following line to its original

```
#define XBEECONFIG 0            // 1 to enter XBEE Configuration Mode
```

   d) Upload the Firmware to the Unit again as outline in Step #4 in #4.2.A SYSTEM OPERATIONS|Firmware|Uploading

5.4 Customizing Firmware

5.4.1 Pump-Controller

- ✔ The Pump-Controller Firmware preforms the following duties
  - ○ Wait for Hand-Remote Requests and generate responses
  - ○ Calculate values (Virtual-Pin Values) for special devices that require a process to obtain a value (like ping/echo for Ultrasonic level measurements)

- Retain a Power-Off status after a power failure has occurred at the pump (Arduino is designed to keep digital pins low after a power failure)

NOTE: Wiring, Timing, and Almost all system functions are implemented on the HAND-REMOTE Firmware

A. Virtual Pins

Virtual pins are non-hardware related value identifiers that are created and assigned by the PUMP-CONTROLLER firmware and then accessed by the HAND-REMOTE as a "Pin" value. An example case is the ultrasonic water level; The pump-controller firmware pings and times an echo of the ultrasonic distance meter using its D4 (Trigger) and D5 (Echo) pins and stores the measured value on "virtual pin 64". When the Hand-Remote queries for a value on pin-64 it actually gets the measured value which is a combination of D4 & D5 already calculated by the PUMP-CONTROLLER.

✔ An Example of a Virtual Pin setup on the PUMP-CONTROLLER firmware

```
void loop(){

  XBee.Available();


#if US_PRESENT>0
  // Read UltraSonic water level
  int ulCurrentTime = millis();
  if ( ulCurrentTime > ulLastPing + 1000 ) {
    XBee.VirtualPin(64, sonar.ping_in() );
    ulLastPing = ulCurrentTime;
  }
#endif

}
```

5.4.2 Hand-Remote

✔ The Hand-Remote Firmware preforms the following duties

- Storing the alarm values in non-volatile memory (EEPROM) so user values and settings aren't lost every-time the power is turned off
- Updating the current status (MAIN Values) of the Menu-Items and checking for Alarm boundaries.

- Processes User Input
  - UP/DOWN user buttons will increments or decrements the Menu-Items
  - RIGHT/LEFT user buttons will increments or decrements the Sub-Function of each Menu-Item
  - ENTER (square) user button will preform a value **SET**; or update MAIN values when they are being displayed

- Menu-Items are associated/connected to a **Device** and its **Hardware Pin**
  - Device - Example; The Hand-Remote running this firmware, or a Pump-Controller device controlling a pump
  - Pin - Example; 7(for D7) or A2 as identified inside the Pump-Controller Unit or by the Arduino UNO board

- Menu-Items have **Sub[?]** Values, such as
  - The actual/read/**MAIN** value of the Menu-Items - Example; Power is currently either ON/OFF (current status)
  - **SET**ting a Value for the Menu-Item - Example; SET the Power to either ON/OFF
  - Setting a Too **LO**w **ALARM** Value - Example; When pressure is below 10 sound the Low Alarm
  - Setting a Too **HI**gh **ALARM** Value - Example; When pressure is above 100; sound the High Alarm

- Menu-Items can have Option Values (*An Optional way of handling Menu-Item value assignments*)
  - Options associates a common MAIN-Value to a Text identifier; for example ON/OFF.
  - A Menu-Items value must be a select-able set of Options or a Number but cannot be both.
  - The maximum number of options is limited by program line `#define MAXOPTIONS 2 // Maximum number of Menu Item Options allowed`

- Menu-Item Limits
  - Program line `#define MAX_MENU_ITEMS 15 // Maximum number of Menu Items allowed ( using 71% dynamic memory )`
  - Determines the maximum number of menu-items that can exist; it can be increased slightly but may produce memory warnings.

A. Adding & Deleting Devices

✔ The Hand-Remote Firmware tracks ALL the associated devices it can control and monitor.

- To add another Device to the control and monitoring system
  - Add any Unique Name to the Comma separated list of devices on the Program Line shown below
    - `uDevices HandRemote, CanalPump, DitchPump; // Name and Define all controllable devices in the System`
  - At the top of the *SetupMenu()* function; Assign the Unique Named device a *.Text* Identifier (Used in the Menu)
  - At the top of the *SetupMenu()* function; Assign the Unique Named unit Unit's *.Transceiver_ID* number assigned in the units firmware
    - Example Program Line `CanalPump.Text = "Canal Pump"; CanalPump.TransceiverID = 10;`

- Delete devices by removing the items in the adding devices listed above.
  - Note: If there is no conflict of Transceiver_ID's or naming there really isn't a substantial need to remove a "missing" device from the system.

B. Creating a Menu-Item

1. Find the **SetupMenu()** function in the Hand-Remote firmware file; This is where all the Menu-Items are defined

2. Indexing - Determine where in the list of Menu-Items you'd like the new item to appear
   a) All Items MUST BEGIN with line `MenuItemsIdx++;` except the very first item (Usually the Hand-Remote battery status)
   b) The line above increments the Indexing (line count) to allow the new menu-item to exist
   c) Items that do not have [MenuItemsIdx] indexing are used in other places of the firmware and should ***NEVER BE CHANGED!***

3. Device - Assign the device this Menu-Item is attached to
   a) This is the Device's Unique Name given in Step #A|Adding & Deleting Devices
   b) The unit is generally a named Pump-Controller device with an assigned Transceiver_ID.

4. Pin – Assign the Hardware Pin of the device that the Menu-Item will control or monitor
   a) The hardware pin is the terminal inside the controlling device box (pump-controller) that the control/monitor equipment is wired to
   b) NOTE: Digital Pins (D4 → D7) are identified by only their number (e.g. 4 = D4) whereas Analog pins require both letter and number (e.g. A2).
      - Digital Example; `Menu[MenuItemsIdx].Pin=7;`
      - Analog Example; `Menu[MenuItemsIdx].Pin=A3;`

5. Text - Assign the Menu-Item a defining Text
   a) The text is what will be displayed on the Hand-Remote display
      - Example; `Menu[MenuItemsIdx].Text="Power";`

6. Options - Determine IF the Menu-Item will have textual "Options" or is just a numerical value
   a) If the Menu-Item will report numerical values such as Voltage, Water Level, Pressure etc... Skip to step #5
   b) Determine the Menu-Items Options and assign each a;
      - Text Example; `Menu[MenuItemsIdx].Option[0].Text = "On";`
      - Value Example; `Menu[MenuItemsIdx].Option[0].Value = HIGH;`

7. Sub[SET] – Determine IF the user should be allowed to set a value on the Location (Output Pins like turning Power ON/OFF)

   a) If the Menu-Item is for monitoring status and the user will not be setting its value (INPUT)... Skip to step #7
   b) If the Menu-Item needs to allow the user the ability to set the value (OUTPUT)
      - Set the "State" to SETTABLE to allow the user to SET the value
      - Example; `Menu[5].Sub[SET].State = SETTABLE;`

8. Sub[??ALARM] – Determine if the value should be monitored with an alarm
   a) LOALARM
      - If the Menu-items value is numeric and the value should be checked for getting too small
      - OR If the Menu-item value is an "option" and the value should be checked to see if it is EQUAL
        - Assign an LOALARM identifier Example; `Menu[5].Sub[LOALARM].ID = 'c';`
   b) HIALARM
      - If the Menu-items value is numeric and the value should be checked for getting too large
      - OR If the Menu-Item value is an "option" and the value should be checked to see if it is NOT-EQUAL
        - Assign a HIALARM identifier Example; `Menu[5].Sub[HIALARM].ID = 'C';`

9. Proceed to #4.3.2.3.Updating Changes|outline

C. Deleting a Menu-Item

1. Find the menu item you'd like to delete by identifying it by its' `Menu[MenuItemIdx].Text` setting
2. Select ALL the items from `MenuItemsIdx++;` up to but no including the next `MenuItemsIdx++;`
3. Press delete

D. Menu-Item Setup (Example)

The menu items are defined in the **SetupMenu()** *Function identified by line 'void SetupMenu() {'. Each item in the Menu has a numeric index (i.e. Menu[index-#] below the constant 'MONITOR' is assigned index-0 and 'PUMPIDX' is assigned index-1).*

```
//================================================================================================================
//------------------------------ MENU STRUCTURE ( ADVANCED CONFIGURATION ) ---------------------------------------
//================================================================================================================
/*****************************************************************************************************************//**
 * @brief  Setup the LCD menu
```

```
 * @remarks
 * - Allows a single spot customization to the user interface
 * - Display will show the items in the same order as they are defined here
 * @code
 *   exmaple code
 * @endcode
*****************************************************************************************************************/
uDevices HandRemote, CanalPump, DitchPump;                  // Name and Define all controllable devices in the System
void SetupMenu() {

  HandRemote.Text = "Hand Remote"; HandRemote.TransceiverID = TRANSCEIVER_ID;
  DitchPump.Text = "Ditch Pump"; DitchPump.TransceiverID = 10;
  CanalPump.Text = "Canal Pump"; CanalPump.TransceiverID = 11;

  //BATT (idx-0) ----------------------------
  Menu[BATT].Device = HandRemote;
  Menu[BATT].Pin = A1;                                      // Battery level is gotten from the Hand-Remote pin A1
  Menu[BATT].Text = "Battery(B)";                           // Create a menu item for monitoring the Battery
  Menu[BATT].ValueModifier = BATTVOLTS;                     // Modify raw value to show voltage
  Menu[BATT].Sub[LOALARM].ID = 'b';                         // A Low Alarm is identified by a lower-case 'b'

  //----------------------------------------
  MenuItemsIdx++;
  idx = MenuItemsIdx;                                       // !!!-- Set where the Menu will start --!!!
  Menu[MenuItemsIdx].Device = DitchPump;
  Menu[MenuItemsIdx].Pin = 7;                               // Power is set/got on all Pump-Controller's on pin [D7]
  Menu[MenuItemsIdx].Text = "Power(P)";                     // Create a menu item for Power Control
  Menu[MenuItemsIdx].Sub[SET].State = SETTABLE;             // Allow this Value to be 'SET' by the user
  Menu[MenuItemsIdx].Sub[LOALARM].ID = 'p';                 // A Low Alarm is identified by a lower-case 'p'
  Menu[MenuItemsIdx].Sub[HIALARM].ID = 'P';                 // A High Alarm is identified by an upper-case 'P'
  Menu[MenuItemsIdx].Option[0].Text = "Off";                // Power can be "Off"             - Option #0 = Off
  Menu[MenuItemsIdx].Option[0].Value = LOW;                 // "Off" will be the value 'LOW"  - Off = LOW
  Menu[MenuItemsIdx].Option[1].Text = "On";                 // Power can be "On"              - Option #1 = On
  Menu[MenuItemsIdx].Option[1].Value = HIGH;                // "On" will be the value 'HIGH'  - On = HIGH
  Menu[MenuItemsIdx].LastOptionIdx = 1;                     // Last Option Index defined      - Number of Options - 1

  //----------------------------------------
#if ULTRASONIC_WATER_LEVEL_INSTALLED>0
  MenuItemsIdx++;
  Menu[MenuItemsIdx].Device = DitchPump;
  Menu[MenuItemsIdx].Pin = 64;                              // Water Level is read from VIRTUAL (Pump-Controllers firmware) pin 64
  Menu[MenuItemsIdx].Text = "Water (L)";                    // Create a menu item for Water Level Transducer
  Menu[MenuItemsIdx].Sub[LOALARM].ID = 'l';                 // A Low Alarm is identified by a lower-case 'l'
  Menu[MenuItemsIdx].Sub[HIALARM].ID = 'L';                 // A High Alarm is identified by an upper-cse 'L'
#endif
  //----------------------------------------
#if WATER_PRESSURE_INSTALLED>0
  MenuItemsIdx++;
  Menu[MenuItemsIdx].Device = DitchPump;
  Menu[MenuItemsIdx].Pin = A3;                              // The 'signal' is gotten on all Pump-Controllers on pin [A3]
  Menu[MenuItemsIdx].Text = "Pressure(R)";                  // Create a menu item for the Primary Pressure Transducer
  Menu[MenuItemsIdx].ValueModifier = PRESSURE;              // Modify value to display PSI instead of MPa
  Menu[MenuItemsIdx].Sub[LOALARM].ID='r';                   // A Low Pressure alarm is identified by a lower-case 'r'
  Menu[MenuItemsIdx].Sub[HIALARM].ID='R';                   // A High Pressure alarm is identified by an upper-case 'R'
#endif
/*// UN-COMMENT BELOW FOR A SECOND PUMP-CONTROLLER ( NAMED "Canal Pump" ) WITH TRANSCEIVER_ID = 11
  //----------------------------------------
  MenuItemsIdx++;
  Menu[MenuItemsIdx].Text = "Power(C)";                     // Create a menu item for Power Control
  Menu[MenuItemsIdx].Device = CanalPump;
  Menu[MenuItemsIdx].Pin = 7;                               // Power is set/got on all Pump-Controller's on pin [D7]
  Menu[MenuItemsIdx].Sub[SET].State = SETTABLE;             // Allow this Value to be 'SET' by the user
  Menu[MenuItemsIdx].Sub[LOALARM].ID = 'c';                 // A Low Alarm is identified by a lower-case 'p'
  Menu[MenuItemsIdx].Sub[HIALARM].ID = 'C';                 // A High Alarm is identified by an upper-case 'P'
  Menu[MenuItemsIdx].Option[0].Text = "Off";                // Power can be "Off"             - Option #0 = Off
  Menu[MenuItemsIdx].Option[0].Value = LOW;                 // "Off" will be the value 'LOW"  - Off = LOW
  Menu[MenuItemsIdx].Option[1].Text = "On";                 // Power can be "On"              - Option #1 = On
  Menu[MenuItemsIdx].Option[1].Value = HIGH;                // "On" will be the value 'HIGH'  - On = HIGH
  Menu[MenuItemsIdx].LastOptionIdx = 1;                     // Last Option Index defined      - Number of Options - 1

  //----------------------------------------
  MenuItemsIdx++;
  Menu[MenuItemsIdx].Text = "Pressure(F)";                  // Create a menu item for the Primary Pressure Transducer
  Menu[MenuItemsIdx].Device = CanalPump;
  Menu[MenuItemsIdx].Pin = A3;                              // The 'signal' is gotten on all Pump-Controllers on pin [A3]
  Menu[MenuItemsIdx].ValueModifier = PRESSURE;              // Modify value to display PSI instead of MPa
  Menu[MenuItemsIdx].Sub[LOALARM].ID='f';                   // A Low Pressure alarm is identified by a lower-case 'r'
  Menu[MenuItemsIdx].Sub[HIALARM].ID='F';                   // A High Pressure alarm is identified by an upper-case 'R'

  //----------------------------------------
  MenuItemsIdx++;
  Menu[MenuItemsIdx].Text = "Pressure(S)";                  // Create a menu item for the Secondary Pressure Transducer
  Menu[MenuItemsIdx].Device = CanalPump;
  Menu[MenuItemsIdx].Pin = A4;                              // Menu-item is for Pump-Option #0 (Canal) on Pin (A4)
  Menu[MenuItemsIdx].ValueModifier = PRESSURE;              // Modify value to display PSI instead of MPa
  Menu[MenuItemsIdx].Sub[LOALARM].ID='s';                   // A Low Pressure alarm is identified by a lower-case 's'
  Menu[MenuItemsIdx].Sub[HIALARM].ID='S';                   // A High Pressure alarm is identified by an upper-case 'S'
*/
  //------------[ Start-Up the Display ( DO NOT CHANGE! )]-------------
  for ( int i = 0; i <= MenuItemsIdx; i++ ) {
```

```
    if ( Menu[i].Sub[LOALARM].ID != NULL || Menu[i].Sub[HIALARM].ID != NULL ) {
      EEPROMGet(i);                                          // Load Alarm values from EEPROM
    }
  }
  GetItem();                                                 // Get starting Menu item
  LCD_display();                                             // Update the display
}
```

6. OPTIONAL ACCESSORIES

6.1 Hand-Remote

6.1.1 9V Rechargeable Batteries ($18.90)

It is highly recommended to purchase high output 300mAH or more rechargeable 9V Batteries; as the Hand-Remote will eat threw batteries pretty quickly.

At time of writing EBL 4x 600mAHh 9V Li-ion Rechargeable Batteries with Charger could be purchased for $18.90
http://www.ebay.com/itm/300974624904

NOTE: Even if a USB power is plugged in – If the power switch is turned ON it will still draw power from the battery. If you plug in USB for power be sure the power switch on the Hand-Remote unit is turned OFF.

6.2 Pump-Controller

6.2.1 Pressure

- ✔ Features
  - Offers a way to remotely monitor the water pressure
  - Default programming on pin A3

A. Materials ($9.59)

5V 0-1.2 MPa Pressure Transducer Sensor Oil Fuel Diesel Gas Water Air Sensor
Banggood.com ($9.59/EA)

(~30-feet) 4-lead outdoor wire
required to reach from the pipe to the Pump-Controller
(See Outdoor Data Wire in BOM)

B. Installation

1. Remove the current Pressure Gauge from its location

2. Screw in the Pressure Transducer (Typically does fit the same as the pressure gauge)

3. Wire up the Transducer as follows
   a) Transducer RED wire (+5V) → Cat5e ORANGE
   b) Transducer BLACK wire (GND) → Cat5e BROWN
   c) Transducer YELLOW wire (SIGNAL) → Cat5e BLUE

4. Wire the other end of the Cat5e Wire into the Pump-Controller
   a) Cat5e ORANGE → 5VDC
   b) Cat5e BROWN → GND
   c) Cat5e BLUE → A3

C. Firmware Adjustments

The current Firmware supports Pressure measured in PSI.

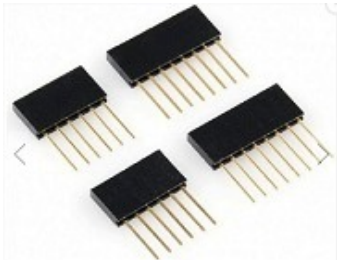
http://forum.arduino.cc/index.php?topic=376384.0

6.2.2 Water Level

✔ Features
- Offers a way to remotely monitor the water level in a ditch
- Default programming on pins D4 (TRIG) and D5 (ECHO) – *On Pump-Controllers Firmware*

A. Materials ($10.77/ea)

(1) DC 5V Waterproof Ultrasonic Module
Distance Measuring Transducer Sensor
banggood.com ($ 10.77/EA)

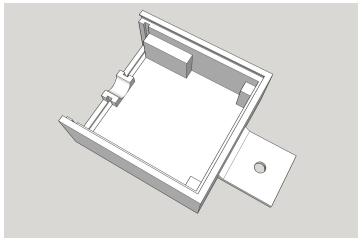


(1) 4-PIN Female Dupont Header
to connect to Circuit Board Pins
(See Dupont Headers in BOM)

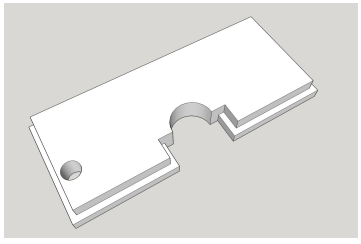
(~30-feet) 4-lead outdoor wire
required to reach from the ditch to the Pump-Controller
(See Outdoor Data Wire in BOM)

B. 3D-Prints ($1.30)

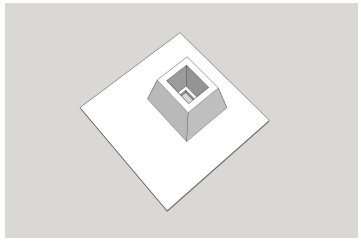
✔ Print the Following Models on a 3D-Printer
- Files Located at https://github.com/tgit23/AgIrrigationRemoteControl/tree/master/Accessories/Sonic%20Water%20Level%20Meter/3D-Prints-STL
- Total Filament = 43.7 cm^3 @ $0.03/$cm^3$ ~ $1.30



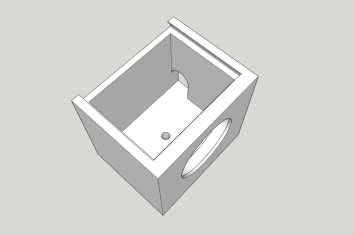
SonicCircuitBoardCase.stl (14.3 cm^3)



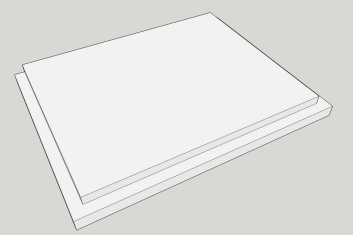
SonicCableCover.stl (2.5 cm^3)



SonicCircuitBoardCover.stl (6.7 cm^3)



SonicHeadCase.stl (17.6 cm^3)



SonicHeadCover.stl (2.6 cm^3)

C. Firmware Adjustments

Current Pump-Controller and Hand-Remote Firmware is setup to support Ultrasonic Water Level Meter

6.2.3 Auxiliary Contact ($43.75)

- ✔ Auxiliary contact, side mounted, 1 N.O. Contact (NOTE: This must match/fit the pump panels Contactor Unit)
  - https://www.automationdirect.com/adc/Shopping/Catalog/Motor_Controls/Eaton_Cutler-Hammer_Contactors_-z-_Starters_-z-_Overloads/Auxiliary_Contacts/C320KGS1 - $43.75/ea

- ✔ Notes
  - Auxiliary Contactor must be purchased according to the contactor they will FIT on.
    - The above suggested purchase will ONLY work with the NEMA-2 Model # AN16GNO Series B1 Contactor
  - The Auxiliary Contactor typically already used by the Start-lock CANNOT be used for this purpose as it has 480V wires already attached to it.
  - Often a used aux contactor can be found much cheaper on ebay

- ✔ Benefits
  - Offers a way to monitor the actual (i.e. directly) the pump power status

6.2.4 Flow Meter

Did not implement due to high costs and limited benefits

http://www.banggood.com/TUF-2000M-TS-2-Digital-Ultrasonic-Flowmeter-Flow-Meter-Ultrasinic-Flow-ModuleRTU-p-1087832.html ($186.77)

7. OTHER RESOURCES / LINKS

- ✔ Support
  - You can email me at tgit28@gmail.com with any questions, comments and or requests.

- ✔ RF Controllers

  - Commercial
    - http://www.forbixindia.com/electronics/remote-motor-control/
    - http://www.remotecontroleverything.com/long-range-wireless-remote-control-3-phase-power-water-pump/
    - http://www.sprinklerwarehouse.com/remote-control-guide-s/6282.htm
    - http://www.sprinklerwarehouse.com/Hunter-Remote-Controllers-Timers-s/109.htm
    - http://rayshobby.net/opensprinkler/
    - Rain Bird Controller wiring diagram http://www.lawnh2o.com/rainbird/PDF/Wiring-Manual.pdf

  - Home-made Projects
    - https://www.instructables.com/id/COMPACT-AND-ROBUST-AUTOMATED-AGRICULTURE-REMOTE/

- ✔ Cellular Controllers

  - Commercial
    - http://www.myfieldnet.com/
    - https://www.ag-rite.com/system-description

- 380V 50Hz 500W (TOO SMALL) https://www.alibaba.com/product-detail/GSM-industrial-three-phase-power-switch_60270579733.html ($100-$200)
- Looks cheap http://www.bieneelectronics.com/products/br_application.htm

○ Home-made Projects
- https://www.instructables.com/id/SMS-controlled-Wireless-Irrigation-System/
- https://www.stavros.io/posts/arduino-powered-irrigation-system/

✔ Component considerations
- ○ Atmel328 with RF boards https://www.digitalsmarties.net/products/jeenode
- ○ ALL about Solid State Relays http://www.phidgets.com/docs/Solid_State_Relay_Primer
- ○ Remote Relay https://www.controlanything.com/Relay/Relay/XSC_PROXR
- ○ RF Remote http://www.remotecontroleverything.com/long-range-wireless-remote-control-3-phase-power-water-pump/ ($83 – A serious alternative @ 2000-meter?? range)

✔ Digi Xbee Links
- ○ Sparkfun Xbee User Guide https://learn.adafruit.com/xbee-radios/
- ○ Sparkfun Xbee Buying Guide https://www.sparkfun.com/pages/xbee_guide
- ○ Xbee 1-WATT (up to 40-miles) https://www.digi.com/products/xbee-rf-solutions/embedded-rf-modules-modems/xtend-module#overview (S3B is .25 Watts)
- ○ Xbee Homepage https://www.digi.com/products/xbee-rf-solutions