Ag Irrigation Remote Control

Using Arduino-UNO and Digi-Xbee RF

Written By: Thomas G (01/2017)
Feel free to use: no strings attached (text content only / images respectfully referenced)
Last Updated: 2017.06.25

Illustration 1: HandRemote

Illustration 2: Pump-Controller

Table of Contents

1. INTRODUCTION

The "***Ag Irrigation Remote Control***" project's goal is to provide RF remote control and monitoring features for an Agriculture-Industry 3-Phase 480VAC Irrigation Pump; typically used in wheel-line and hand-line irrigation setups within the United States.

This project implements Digi-Xbee RF radio modules which advertises a broadcast range of up to 28-miles (More practically a mile or two at ground level) along with Arduino micro-controller boards and an LCD display. The project consists of a Hand-Held "***Hand Remote***" and a pole mounted "***Pump Controller***" as well as a few high voltage control components added to the pumps electrical panel.

- ✔ The project is Open Source and Resides at: https://github.com/tgit23/AgIrrigationRemoteControl
  - ○ Contains a review of the releases and the changes made per release edition.
  - ○ Contains information on purchasing a kit that has already been constructed and tested.
  - ○ Contains the most recent version of this document at: https://github.com/tgit23/AgIrrigationRemoteControl/docs/AgIrrigationPumpRemoteControl.pdf

A. Signal Strength

- ✔ Signal Distance (**Recommended usage for up to 2-miles**; Unless a higher antenna gain or elevation is implemented)
  - ○ Claimed Distance
    - ▪ Digi XBEE S3B RF radio modules are advertised to have a range of up to 26-miles
  - ○ 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. Multiple Device Abilities

✔ Any number of Hand-Remotes can control any number of Pump-Controllers
- Hand-Remotes can be configured to control any number of Pump-Controllers
- Multiple Hand-Remotes can be configured to control the same Pump-Controllers or other Pump-Controllers

2. INSTALLATION INSTRUCTIONS

This section is a guide to installing the PUMP-CONTROLLER to the mechanisms inside the irrigation pumps electrical panel.

✔ The following **SUPPLIES** will be needed

- Pump-Controller
  - The Controller Box
  - Antenna
  - Cable with 4-Pin Plug ending

- Installation Kit Including the following items
  - 480VAC 3-Phase to 24Vdc Power Supply attached to a DIN mounting rail
  - 480VAC or higher Output Voltage Solid State Relay (SSR) 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
  - (2) #10-32 x 3/4" Machine Screws

✔ The following **TOOLS** will be needed
- Voltmeter that reads up to or beyond 500VAC
- Cordless drill
- Center tap
- Bit (#21 for #10-32 Machine Screws)
- Threading tap (#10-32 for #10-32 Machine Screw)
- Protective insulated and flash protection personal gear
- Wire strippers and cutters

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

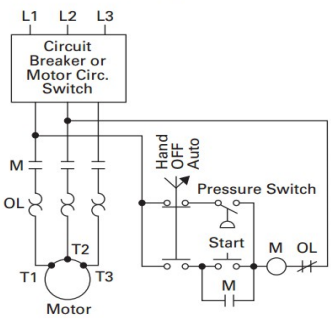
A. Electrical Panel Introduction

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 (see Panel picture; white block with a blue center) contains:
  - Activation Coil - The **M-Circle** in the wiring diagram just right of the Start Button (Power to 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 → Motor (Sometimes called the "Hammer")
    - The Main Contacts are significant part of the "Starter Contactor" as a whole unit.

  - Hold Contact - The **M-Contact** (-||-) wired in parallel and just below the Start Button "LOCKS" the Activation coil ON at 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
  - Wiring diagram above shows this 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 Contactors.

✔ 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

1. 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.

2. Drill two holes in the desired location, punch, tap-thread, and mount the DIN rail horizontally

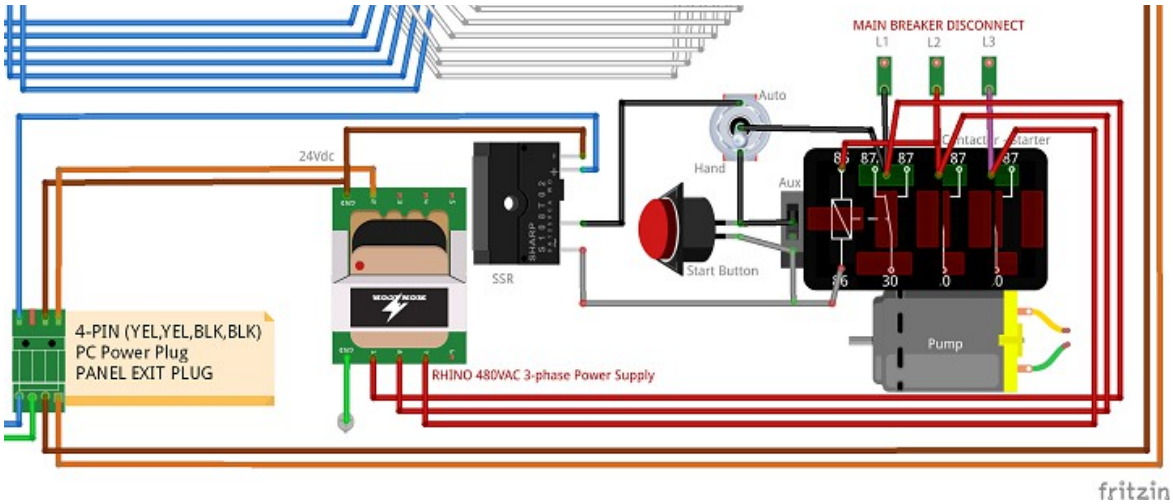
a) Recommend using a #10-32 x 1/2" Machine Screw
b) Drill bit size for #10-32 is #21 – 5/32" - 0.1590"

3. Mount the DIN rail with the Power Supply and Solid State Relay attached

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

C. Pump Panel wiring

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

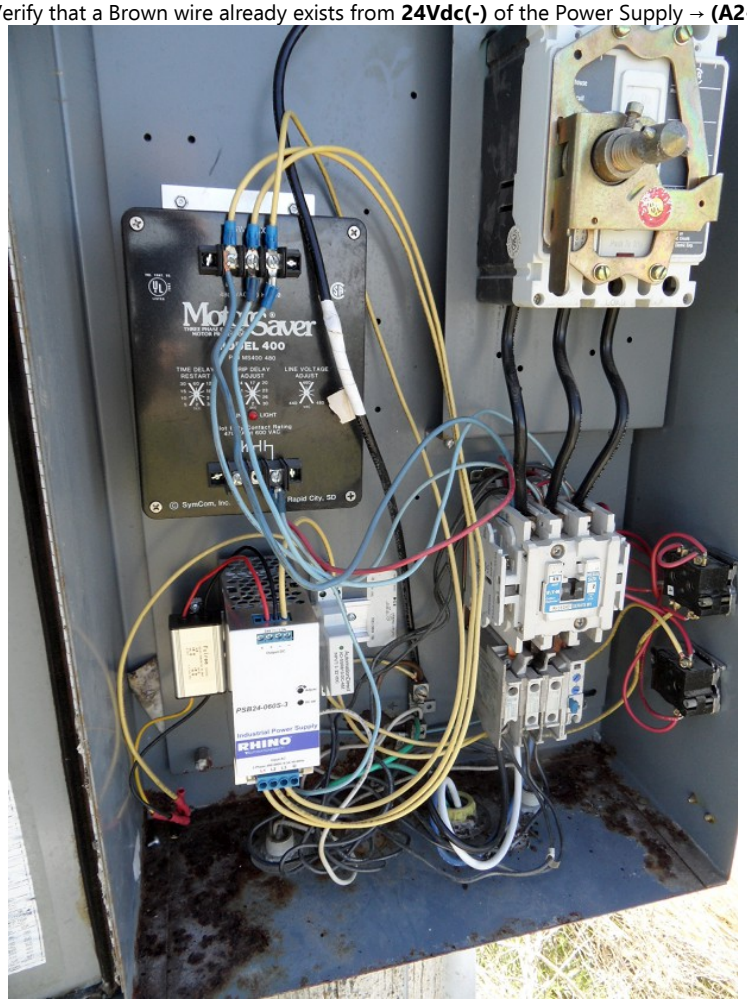


1. Wire the 480VAC 3-Phase Power Supply
   a) Connect Power Supply Terminals L1, L2, L3 to the top terminals of the Contactor (i.e. Starter / Hammer) L1, L2, L3 terminals
      • Often this is easiest to do by wiring to a "Motor Saver" unit that is already wired to L1, L2, L3 if your panel has one

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

2. Wire the Solid State Relay
   a) Connect Solid State Relay (15) → **AUTO** Terminal of the HAND-OFF-AUTO switch
   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 Computer power plug from 3.3.1.A **Exit Plug** insert 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. Precautions
   a) Never stand in front of the panel door while activating the breaker (stand to the side or behind the panel)
   b) Be sure to stand far away from the pump panel while testing the unit – if the SSR wasn't wired correctly; it WILL EXPLODE.

2. Turn the Electrical Panel Breaker ON.
3. Switch the Hand-Off-Auto switch to AUTO.

4. Using the Hand-Remote Device
   a) Find the Menu-Item “Power” for the Pump-Controller Device just installed
   b) Press (→) till 'SET->' and then UP/DOWN to switch to ON

c) Press the square 'Enter' button – The pump should start running
d) Press (→) till 'SET->' and then UP/DOWN to switch to OFF
e) Press the square 'Enter' button – The pump should shut off

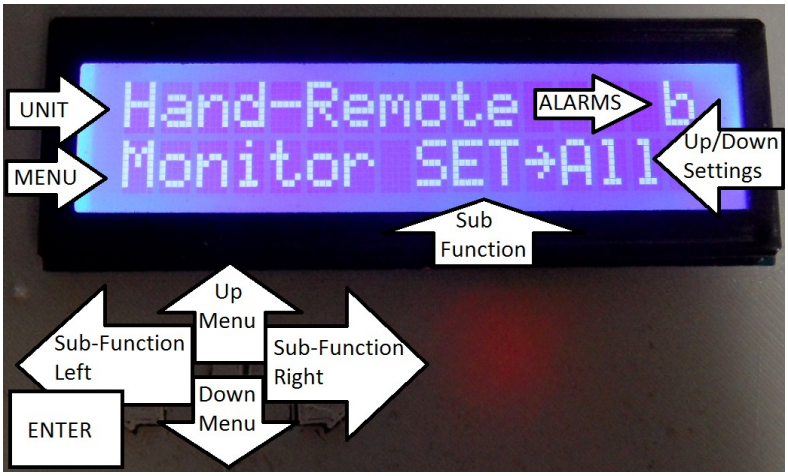
3. OPERATING INSTRUCTIONS

Notice: Firmware must be uploaded to a newly assembled unit before it can operate; visit #4.1.A SYSTEM OPERATIONS| Firmware | Uploading for details

A. Introduction

✔ The display has a Top Row and a Bottom Row that are adjusted by using (Arrow Buttons as shown below in the parenthesis)

- Top Row (Status Row)
  - Left – The Device (Unit) that is currently under control or being monitored
  - Right – The Alarms for the current Unit's → Menu-Item that are 'ON' (Alarm 'ID'entifying letters)
- Bottom Row
  - Left (*Up-Down Arrows*) – The current Menu-Item
  - Middle (Right-Left Arrows) – The Current Sub-Function of the Menu-Item
    - (=) – **Read** Value for the current Menu-Item (This can also be thought of as the "Main Menu")
    - SET→ – **Set** a Value for the current Menu-Item
    - (=!) or (<!) – **Low Alarm** boundary; Sets an alarm value that is either equal-to an option or is less-than a number
    - (≠!) or (>!) – **High Alarm** boundary; Sets an alarm when value is either not-equal to an option or is greater-than a number
  - Right (Up-Down Arrows) – The Value (of or for) the Unit's → Menu-Item's → Sub-Function



✔ Maneuvering the display items can be represented as in the table 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.

B. Alarms

- ✔ Alarms check 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 15-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 (! ǂ, !>)
    - ▪ 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
    - ▪ (ǂ!) - 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 individual pieces and assembling or used for a greater understanding of the devices themselves for debugging and troubleshooting.

4.1 BILL OF MATERIAL (BOM) - $242.66

Pricing for multiple units may be a little less than itemized pricing due to re-usage of excess material.

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 = **$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

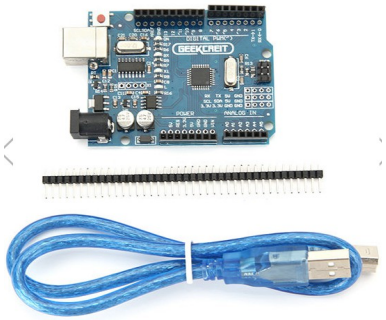
A. Pump-Panel Components ($99.50)

(1) RHINO switching Power supply, 24 VDC output,
2.5A, 60W, 320-600 VAC input, 3-phase, 35mm DIN
AutomationDirect.com ($59.00/ea)
--- OR ---
(1) Delta 3-phase to 24 VDC Power Supply 2.5A
Mouser.com ($65.93/EA)
--- OR ---
TDK Lamda Rail Power Supplies 120W 12V 10A 380-480VAC
Mouser.com ($104.04/EA)

6" of DIN rail, slotted, 35mm, 10 mm height
For mounting Power Supply and Solid State Relay of each Pump
AutomationDirect.com ($8.00/2x 1-Meter Pieces)

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

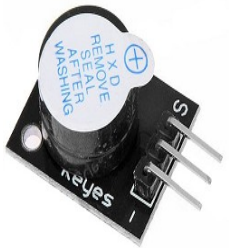
(1) Keypad Shield Blue Back-light for Arduino Robot LCD 1602 Board

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

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

(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

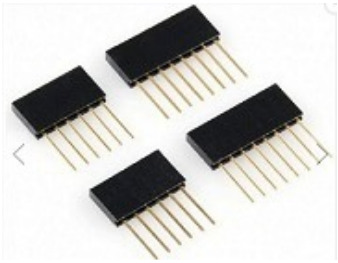


(1) Active Speaker Buzzer Alarm

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

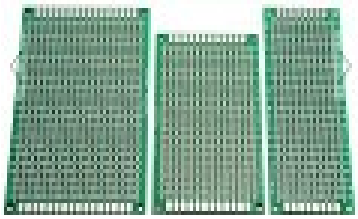
(1) Right-Angle 9Vdc Power Plug for Arduino Power Jack
Aliexpress.com ($1.76 / 2pcs)
Amazon.com ($4.51 / 10pcs)

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



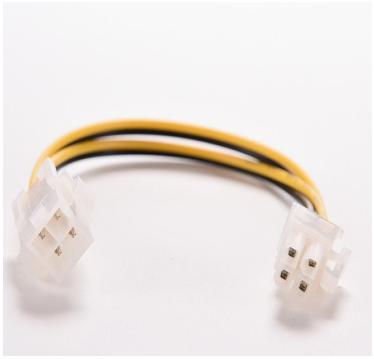
(1) 2-PIN Male – Zigbee → Perfboard POWER
(1) 4-PIN Female – Perfboard → Zigbee POWER
(2) 6-PIN Male – Zigbee I/O → Perfboard
(1) 4-PIN Female - Zigbee->Perfboard RX/TX

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



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

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



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

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

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

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

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

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

(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 ($26.70)

		
(~30-feet) 24-AWG outdoor direct burial wire Amazon.com ($11.50 / 50ft) Enough to connect all accessory/power devices to the Pump-Controller	(~10-feet) **600V** 12-AWG to 16-AWG, 75-deg-C 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

E. Miscellaneous Materials

- ✔ (2) #10-32 x 3/4" Long Machine Screws
- ✔ (5) #4 or smaller x 1/4" Long Screws

F. Tools Required ($NA)

- ○ 3D-Printer (*may be optional if other casing solutions are available*)
- ○ Soldering Iron / Solder / Connecting Wire (22AWG Solid Core)
- ○ Phillips Screw Driver

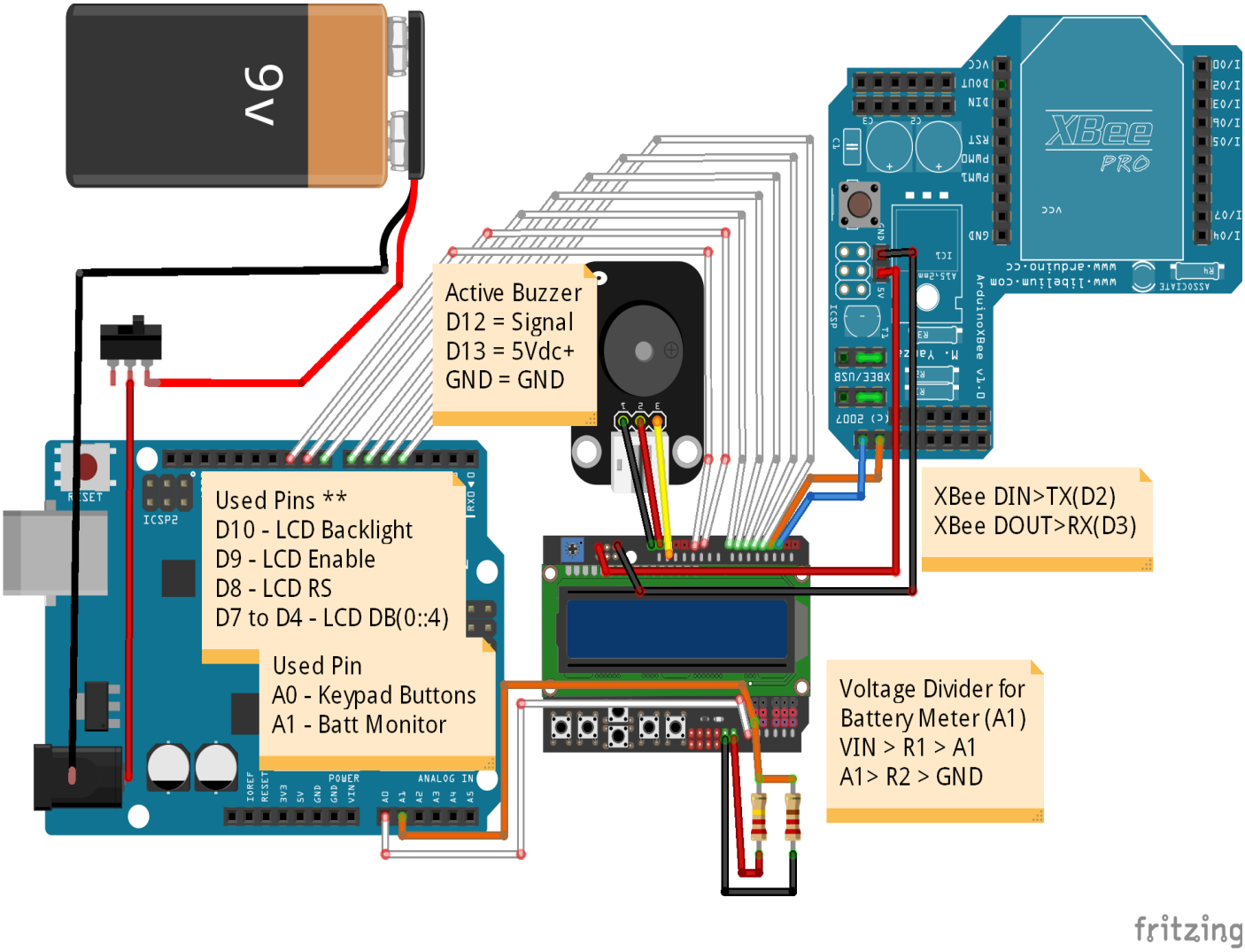
4.2 HAND-REMOTE

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

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

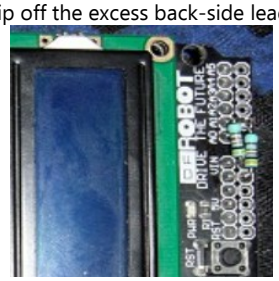
4.2.1 Electronics Assembly

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



A. LCD Keypad Shield

1. Wire the Battery-Meter Voltage-Divider
   a) Select (2) Resistors; One-Value being 1/3 the other (10K-ohm up to 1M-ohm is recommended; A1 draws 1.6nA)
   b) Solder First Resistor (Full Value) on the top of the LCD Shield from **VIN → A1**
   c) Solder Other Resistor (1/3 Value) from **A1 → GND** (Solder hole just left of VIN)
   d) Snip off the excess back-side leads



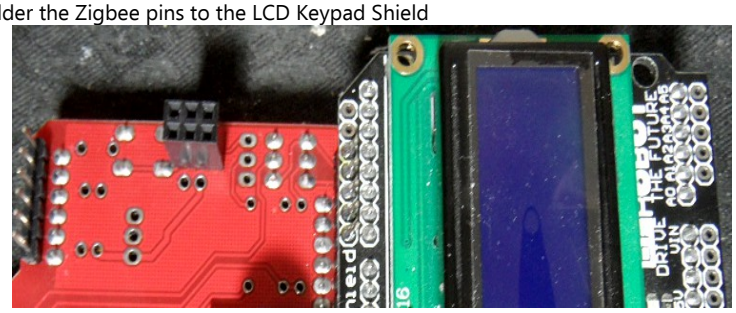
The two resistors setup a voltage divider from the 9Vdc battery (which we will attach to VIN). Analog inputs on the Arduino should never exceed 5Vdc. The voltage divider will split the battery voltage to 1/3 its actual voltage. This will allow the hand-remote to monitor the battery level. The larger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

2. Plug the LCD Keypad Shield onto the Arduino UNO

B. Zigbee/Xbee Shield

1. Remove Header Pins **D5, D6, and D7** (last pin as shown in picture) 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
   c) Repeat steps for D6 and 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 into LCD Keypad Solder Pin-holes D2→D6 (**Shifted towards center by 2-Pin-holes 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

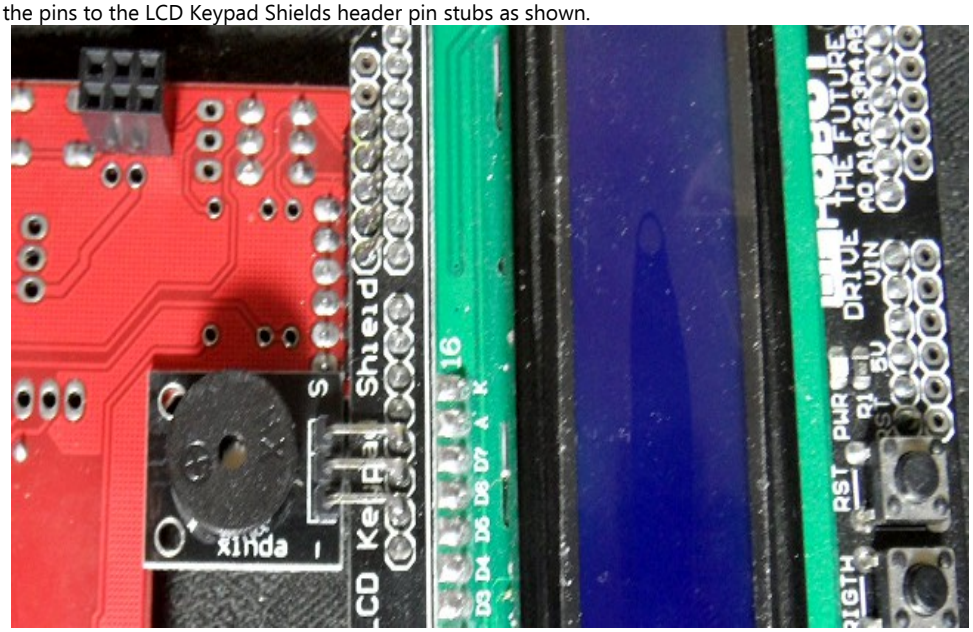


✔ Further Reading for the Curious
- https://www.arduino.cc/en/Main/ArduinoXbeeShield

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

C. Active Buzzer

1. Solder Active Buzzer to LCD Keypad Shield
   a) Place active buzzer on edge of the Zigbee Shield with its **'S' pin** to **D12**, **'+' pin** to **D13**, and **'-' pin** to **GND**
   b) Solder the pins to the LCD Keypad Shields header pin stubs as shown.



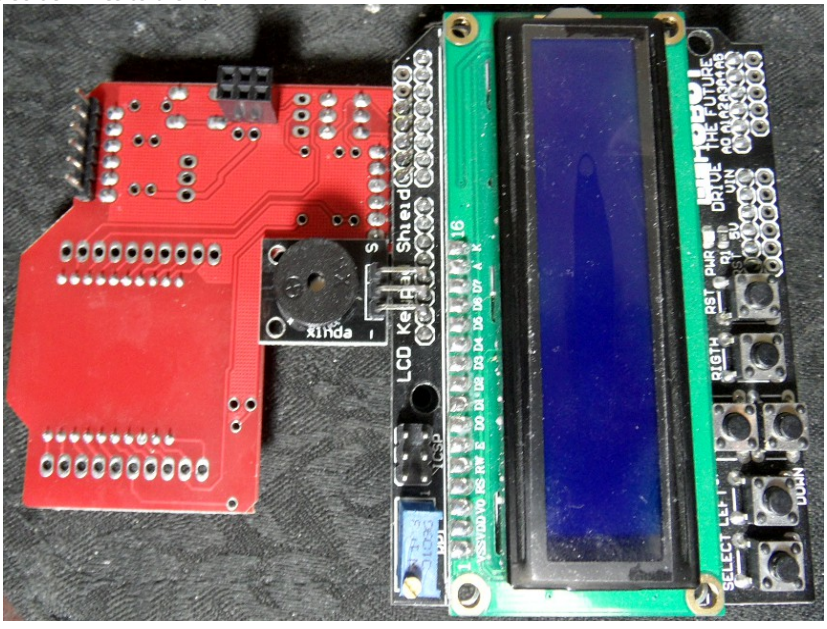
2. Wire Power to the Zigbee Shield

   a) Remove the LCD Keypad Shields ICSP pin header by heating and pulling all 6-Pins

   b) Connect Positive Power
   - Solder one end of a **Red** wire → Left-Most Top Pin of the LCD Keypad Shields ICSP pin header (**5V+**)
   - Solder other end of the wire → Pin hole labeled (5V) on the front of the **Zigbee Shield** next to the ICSP pins and solder

   c) Connect Negative Power
   - Solder one end of a **Black** wire → Right-Most Top Pin of the LCD Keypad Shields ICSP pin header (**GND**)
   - Solder other end of the wire → Pin hole labeled (GND) on the front of the **Zigbee Shield** next to the ICSP pins and solder

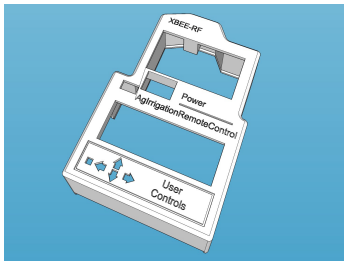
Red/Black wires were not cut but the insulation was split using automatic wire strippers. The Center and Left Pins of the Buzzer were cut just long enough to solder wires to them.



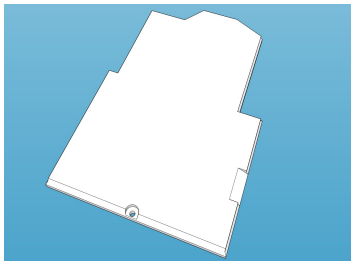
4.2.2 Hardware Assembly

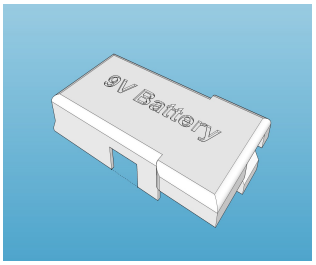
A. 3D-Prints ($2.82)

✔ 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

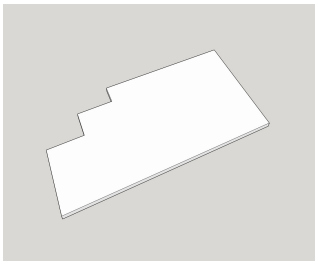


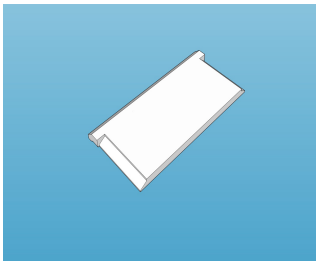
HandRemote-Case.stl (62g)

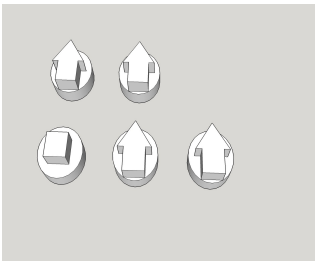

HandRemote-BackCover.stl (23g)



HandRemote-BatteryCover.stl (4g)

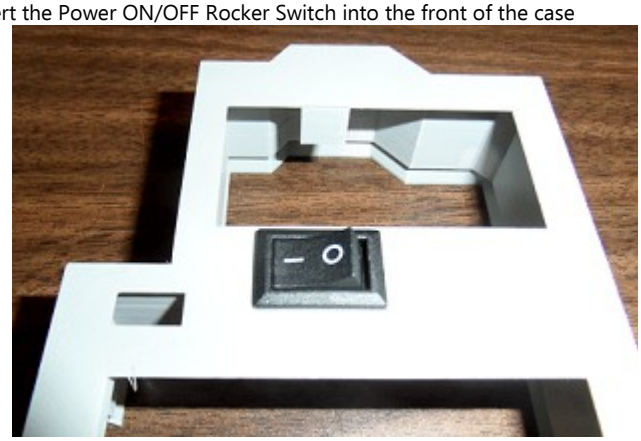

HandRemote-BatteryInsulatorPlate.stl (2g)


HandRemote-SideCover.stl (2.5g)


HandRemote-Buttons.stl (1g)

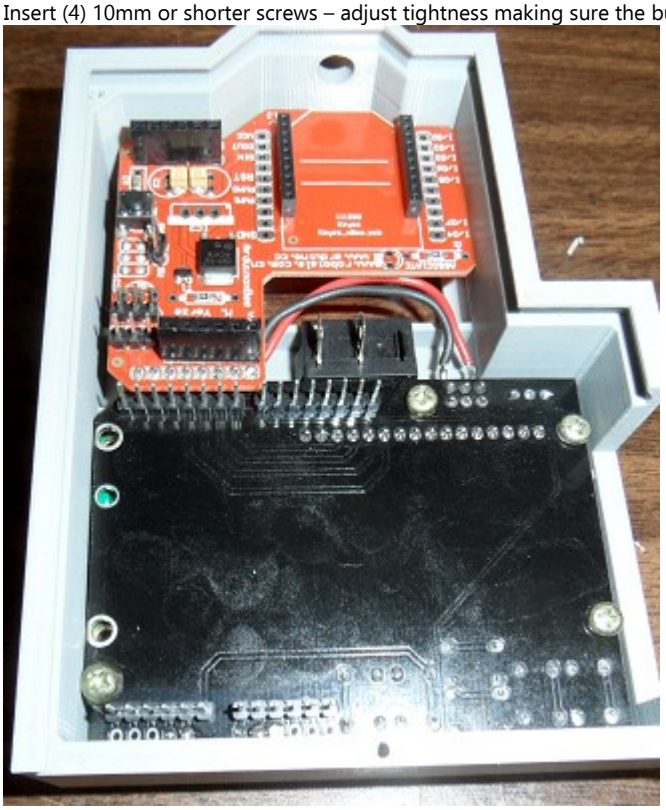
B. Install Electronics

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

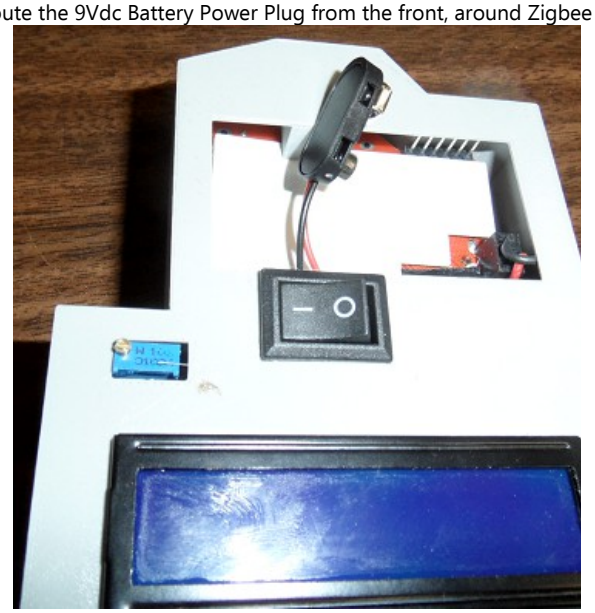
3. Install the LCD, Buzzer, Zigbee **Electronics Assembly** (from Step #2.1) into the Case (Arduino UNO Removed)
   a) Insert (4) 10mm or shorter screws – adjust tightness making sure the buttons still click (*too tight will pinch buttons down permanently*)



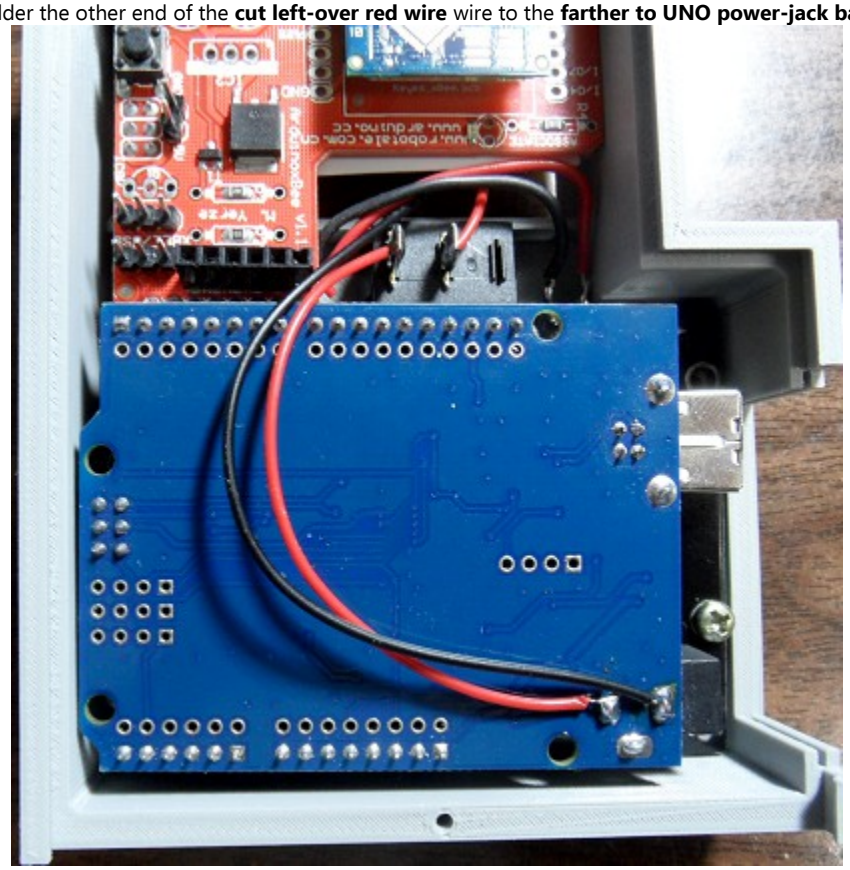
4. Plug the Arduino UNO back onto the LCD Keypad Shield
5. Slide the Xbee into the Top-Hole and then plug into the Zigbee Shield
6. Attach the Antenna to the Xbee RPSMA Connector

C. Wire in Battery Power

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



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

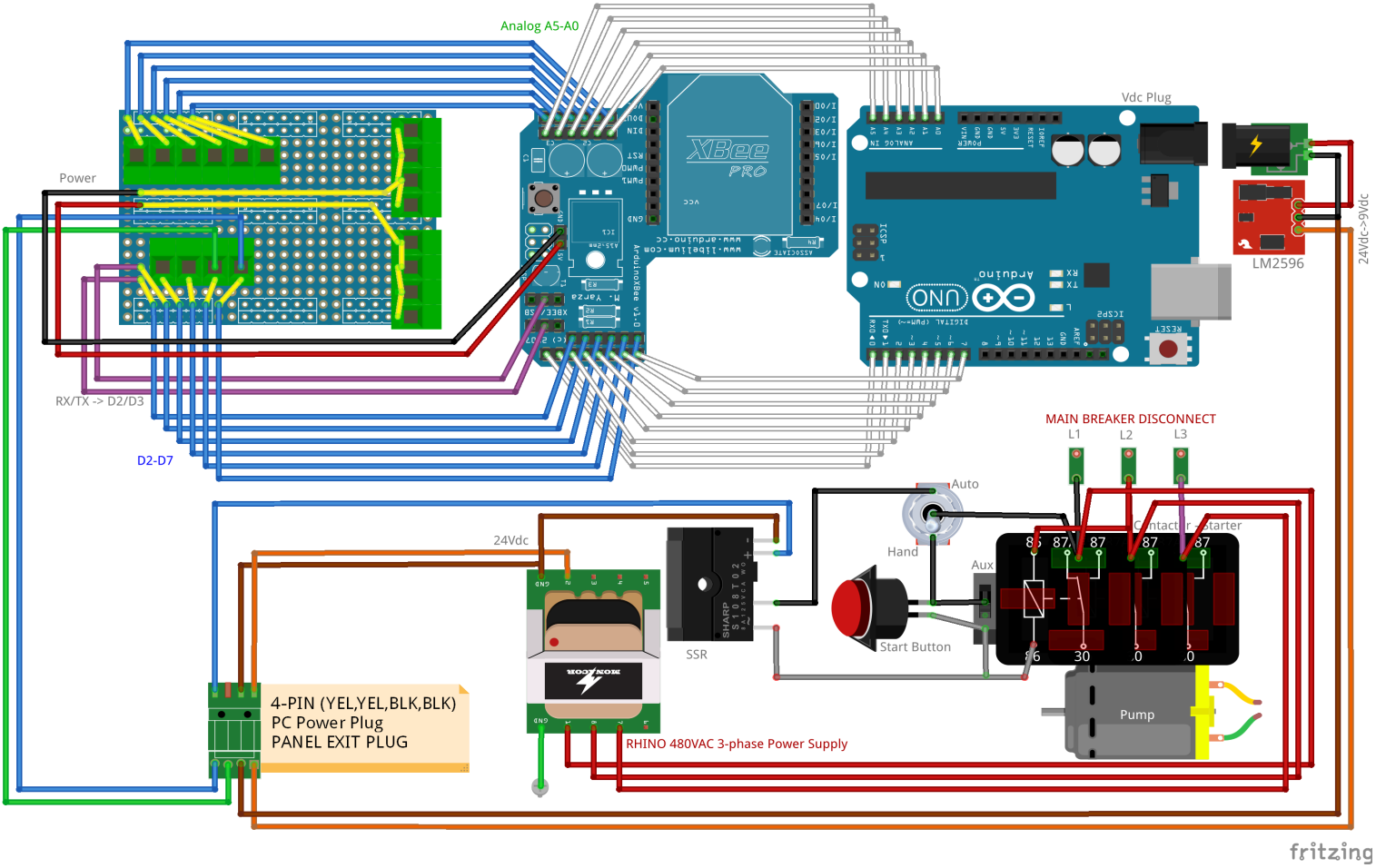


3. Check for Power
    a) Plug in a 9V-Battery into the 9V plug
    b) Flip the rocker-switch to power-on (marked with a "-") and verify that the LCD display lights

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

4.3 PUMP-CONTROLLER

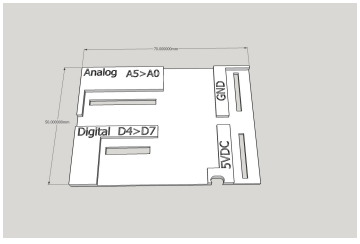
✔ Wiring Diagram of the PUMP CONTROLLER - (Screw Terminal Perfboard → Zigbee Shield → Arduino UNO)



4.3.1 Electronics Assembly

A. 3D-Prints ($0.20)

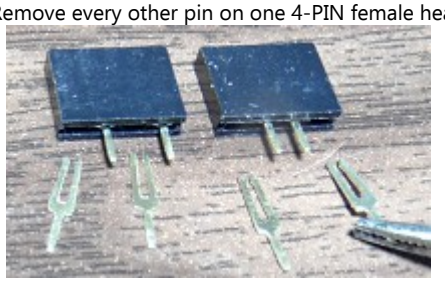
✔ Print the Following Model on a 3D-Printer
- Files Located at https://github.com/tgit23/AgIrrigationRemoteControl/tree/master/PumpController/3D-Prints-STL
- Total Filament = 6.3 cm3 @ $0.03/cm3 ~ $0.20



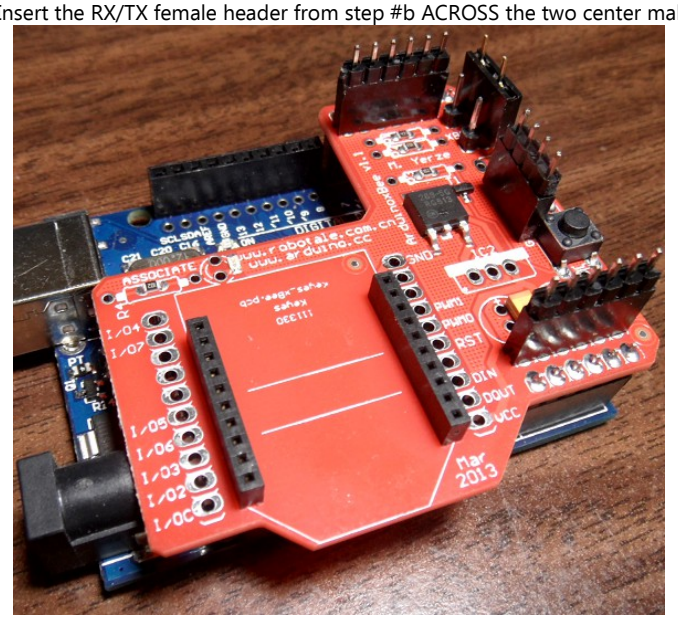
PumpController-PerfboardCover.stl (6.3 cm3)

B. Perfboard Assembly

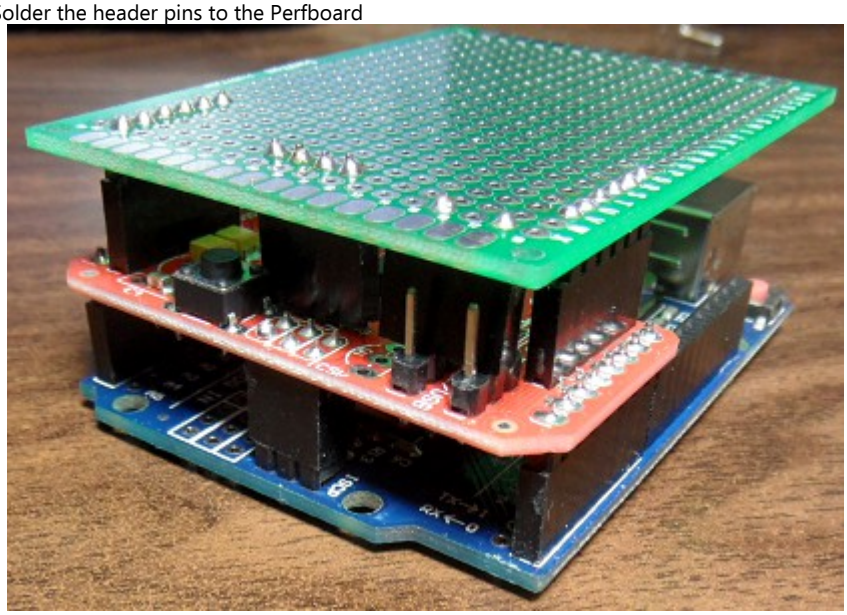
1. Install Headers
    a) Remove outer pins on one 4-PIN female header (POWER)
    b) Remove every other pin on one 4-PIN female header (RX/TX)



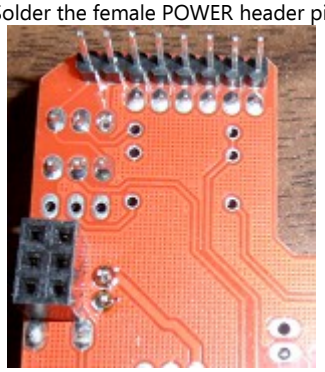
    c) Remove the (2) Green Jumpers on the Zigbee Shield and plug it onto the Arduino UNO
    d) Plug a 6-Pin Male Header into the Analog Header of the Zigbee Shield (A5 → A0)
    e) Plug a 6-Pin Male Header into the Digital Header of the Zigbee Shield (D2 → D7)
    f) Insert the POWER female header from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



2. 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 (12) wires approximately 1" long and strip one end.
   b) Cut (2) wires approximately 3" long and strip one end.

   c) Unplug the Perfboard from the Zigbee Shield
   d) Insert (6) 1" wires from the Zigbee-side of the Perfboard into the holes just **below** the analog male pin headers, bend and solder
   e) Insert (4) 1" wires from the Zigbee-side of the Perfboard into the 4-right (D4-D7) holes just **above** the digital male pin headers, bend and solder
   f) Insert (2) **3"** wires from the Zigbee-side of the Perfboard into the right of the POWER male pin headers (Center-2), bend and solder
   g) Attach (2) 1" wires from the Top-side of the Perfboard across RX/TX female pin headers and solder to D2, D3 Digital pins as shown and trim.

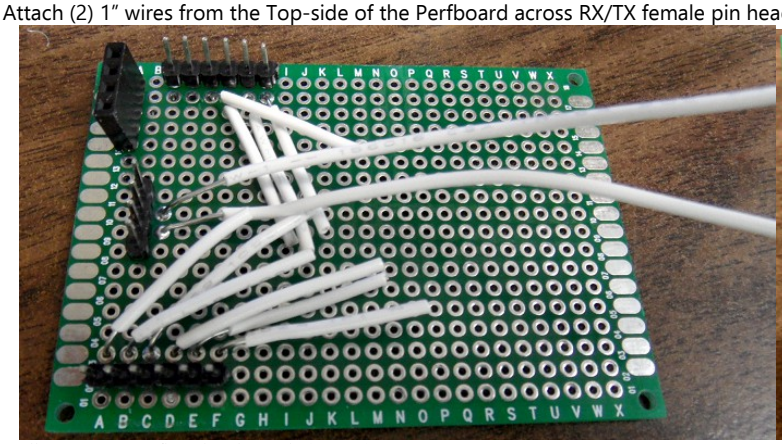


Illustration 4: Perfboard Zigbee-Side (Bottom)

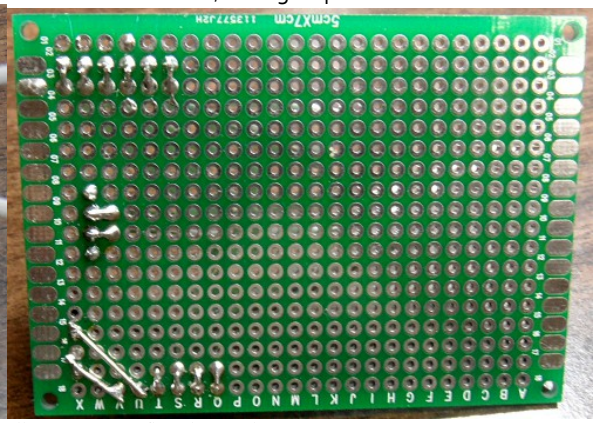
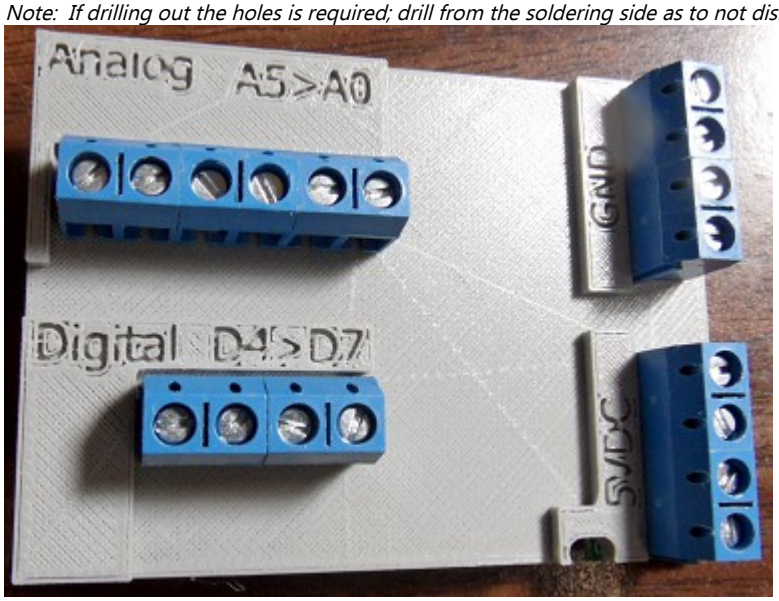
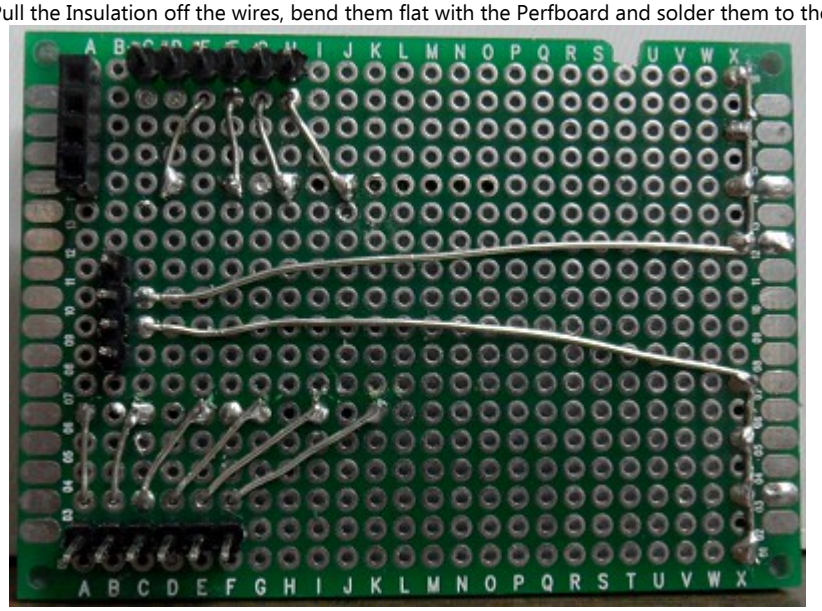


Illustration 5: Perfboard Top-Side

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



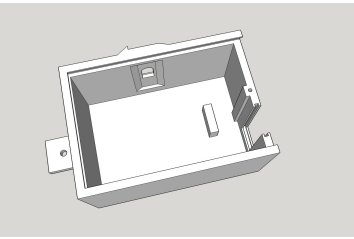
6. Solder the Screw Terminals to their corresponding header wires
    a) Turn the Perfboard over holding the screw terminals into place
    b) Sold each screw terminal pin to the Perfboard
    c) Pull the Insulation off the wires, bend them flat with the Perfboard and solder them to their corresponding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



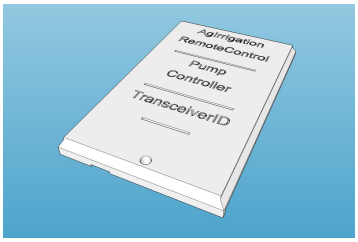
4.3.2 Hardware Assembly

A. 3D-Prints ($3.87)

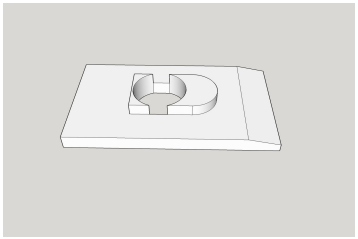
- ✔ Print the Following Models on a 3D-Printer
  - Files Located at https://github.com/tgit23/AgIrrigationRemoteControl/tree/master/PumpController/3D-Prints-STL
  - Total Filament = 129-grams @ $0.03/gram ~ $3.87



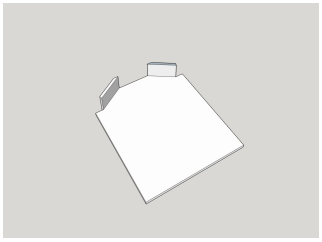
PumpController-Case.stl (98g)



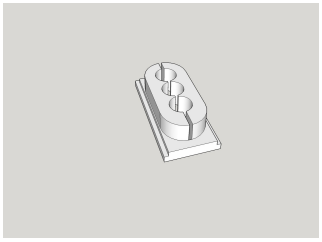
PumpController-Cover.stl (25g)



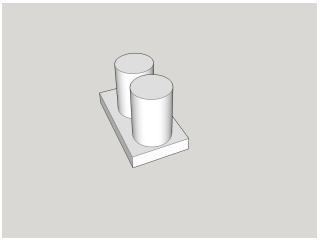
PumpController-AntennaCover.stl (1g)



PumpController-XbeeInsulatorPlate.stl (1g)



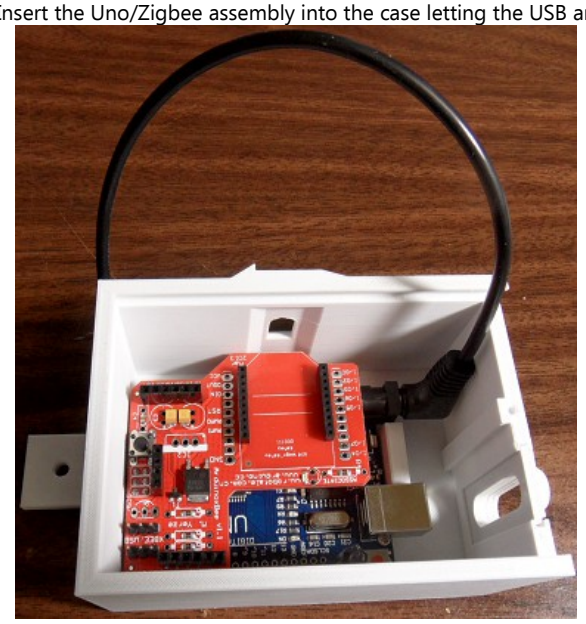
PumpController-CableClampCover.stl (3g)



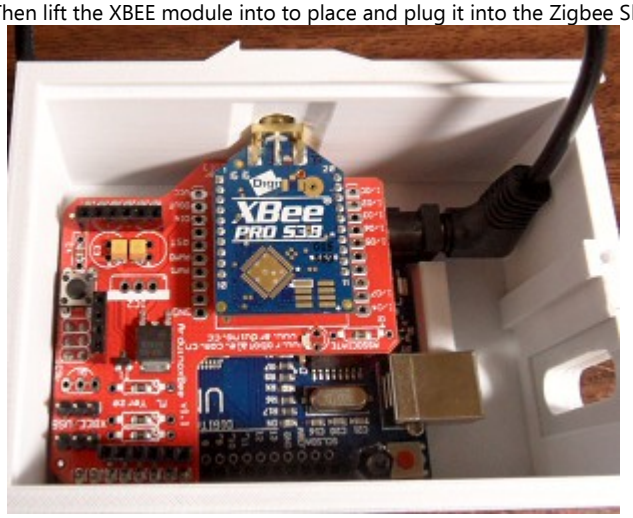
PumpController-CableClampCover-2HoleInsert.stl (1g)

B. Install Electronics

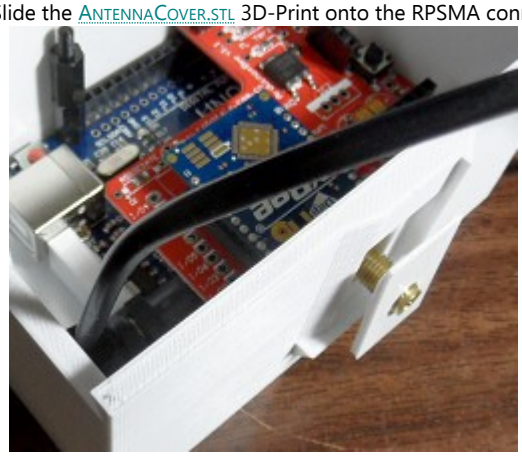
1. Install UNO and Zigbee connected into the Case
   a) Remove the Screw Terminal Perfboard if attached and connect the Arduino UNO & Zigbee Shield
   b) Plug in the Right-Angle 9Vdc Power Plug for Arduino Power Jack
   c) Insert the Uno/Zigbee assembly into the case letting the USB and power Jack sit on each side of the case stub.



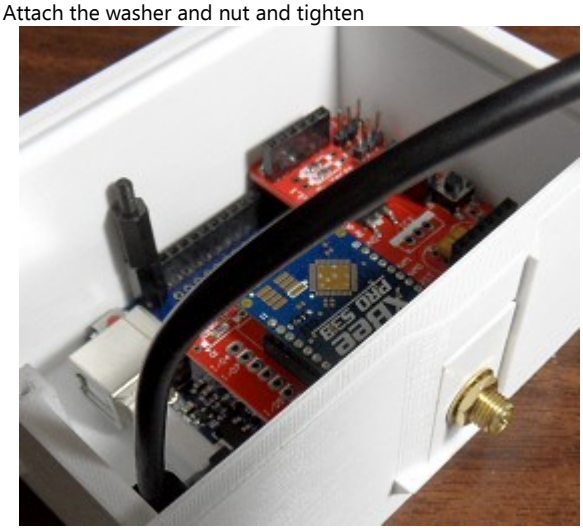
2. Install the XBEE module
    a) First slide the XBEE RPSMA Connector into the case slot
    b) Then lift the XBEE module into to place and plug it into the Zigbee Shields Xbee plug-in



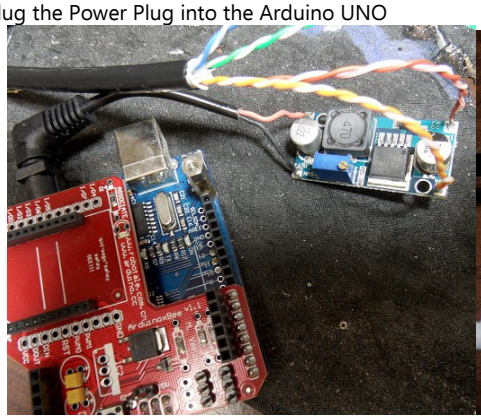
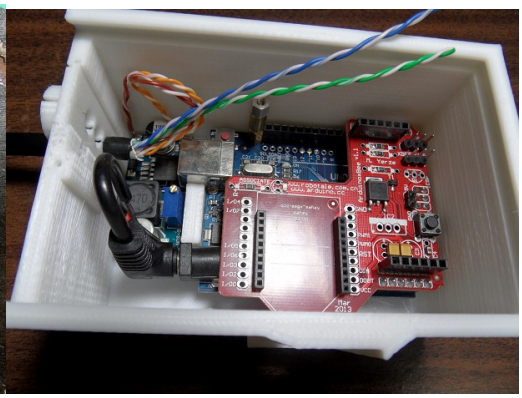
3. Install the Xbee Antenna Cover
    a) Slide the ANTENNACOVER.STL 3D-Print onto the RPSMA connector from the outside of the case



    b) Attach the washer and nut and tighten



4. Install the Cable and DC Converter Module
   a) Cut a 10' length of Outdoor data cable
   b) Strip off the outer insulation of one end approximately the length of the Pump-Controller case.
   c) Strip off the ends of the Orange and Brown twisted pairs
      - Solder the **Orange** twisted pair to the LM2596 Module pad labeled IN+
      - Solder the Brown twisted pair to the LM2596 Module pad labeled IN-
   d) Cut the Right-Angle DC Power plug cords length to approximately 2”
      - Solder the Red Wire to the LM2596 Module pad labeled OUT+
      - Solder the Black Wire to the LM2596 Module pad labeled OUT-
      - *NOTE – The picture below shows these wires the WRONG way*
   e) Insert the cable into the CableClampCover.stl 3D-Print and insert it with the LM2596 Module into the Case as shown
   f) Plug the Power Plug into the Arduino UNO

4.4 INSTALLATION KIT

A. DIN Rail Components

1. Plan location for Power Supply and SSR and cut DIN-Rail

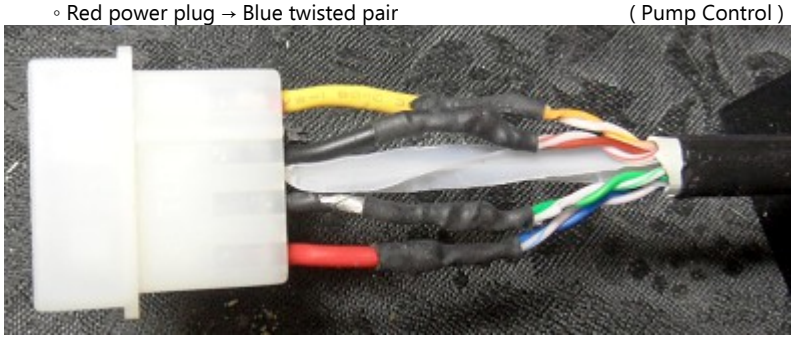
   a) Cut to length a port of DIN Rail long enough to hold the Power Supply and SSR (Recommend 6”)

   b)

2. Connect a Wire (Recommend BROWN, WHITE or GREY) from the **24VDC(-)** on top of the Power Supply to **A1** of the Solid State Relay

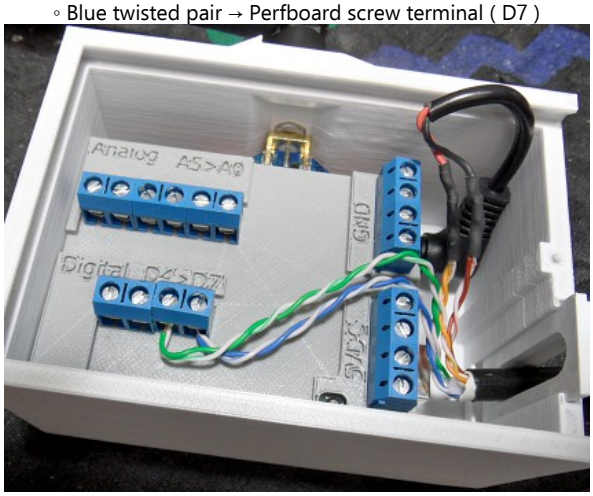
B. Pump-Controller Power Cable

1. Create Cable to connect the Pump-Controller → Pump-Panel-**Exit Plug** which will be installed in 3.3.2-Section D-Step#3
   a) Separate the two connecting ends of a Computer power plug (shown in the BOM) by cutting the wires in the middle
   b) Wire up the **MALE end** of the cable and save the FEMALE end for the Pump-Panel-**Exit Plug**
      - Strip the wire ends of the Cat5e twisted pair wires
      - Strip the wire ends of the male Computer power plug
      - Solder the Computer power plug to the twisted pair wires as shown
        - Yellow power plug → Orange twisted pair (9Vdc+)
        - Black next to Yellow power plug → Brown twisted pair (9Vdc GND)
        - Black next to Red power plug → Green twisted pair (Extra Accessory Wire)
        - Red power plug → Blue twisted pair (Pump Control)



c) Wire up the Pump-Controller end (Other end of the same wire in step#a above)

- Strip the outer insulation from the Cat5e Cable approximately 4" from the end
- Strip and wire the twisted pairs as follows
  - Orange twisted pair → Red wire of the angled 9Vdc UNO Power plug
  - Brown twisted pair → Black wire of the angled 9Vdc UNO Power plug
  - Green twisted pair → Perfboard screw terminal (D6)
  - Blue twisted pair → Perfboard screw terminal (D7)



d) Tape off the Black wire next to the Red wire (*Can be used as with an "optional" auxiliary contact to verify pumps power status*)

5. SYSTEM SETUP

5.1 XBEE Setup

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

1. 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
      ```

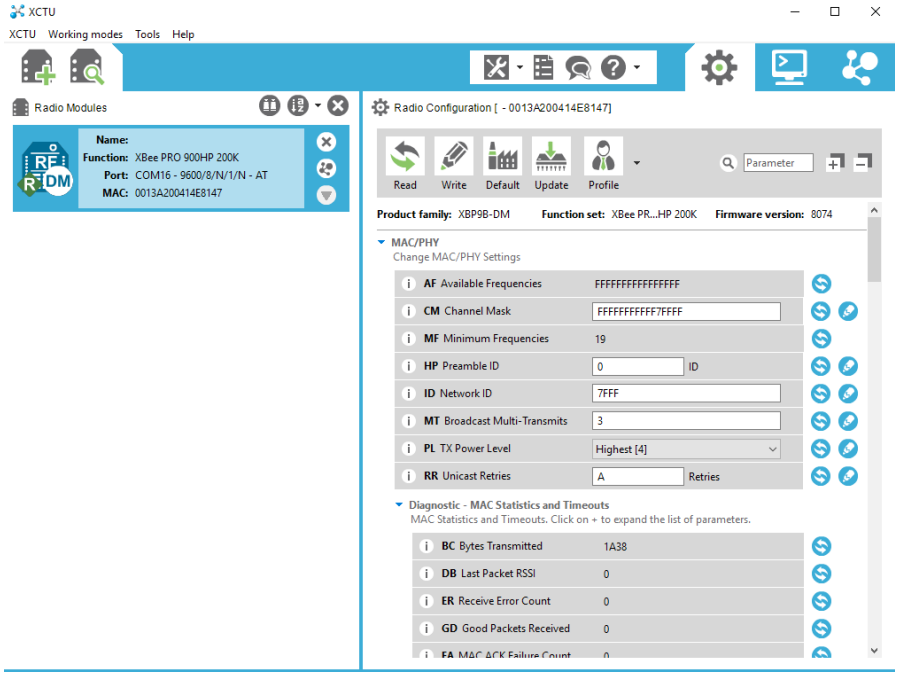
      Change it to

      ```
      #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

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

3. 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

4. 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

5.2 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 Customizing Firmware

5.3.1 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

}
```

5.3.2 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-Values of each Menu-Item
  - SELECT (square) user button will preform a value **SET**; or update MAIN values when they are being displayed

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

- 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 or PUMP-FIELD, PUMP-GARDEN and etc...
  - 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 items index (ie. Menu[index]) you'd like to delete by identifying it by its' Menu[Index].Text setting
2. Select ALL Menu[index] lines with the same index
3. Press delete
4. 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
5. Proceed to #4.3.2.3.Updating Changes|outline

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).*

```
/******************************************************************************************************************//**
 * @brief  Setup the LCD menu
 * @remarks
 * - Allows a single spot customization to users configuration
 * - Be sure to change 'NUM_MENU_ITEMS' when adding/deleting items
 * - Display will show the items in the same order as they are defined here
 * @code
 *   exmaple code
 * @endcode
*********************************************************************************************************************/
void SetupMenu() {

  //vvvvvv[ HAND-REMOTE MENU ITEMS ]vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
  Menu[MONITOR].Text = "Monitor";                      // Create a menu item for monitoring ALL Pump-Controllers
  Menu[MONITOR].Location = HAND_EPROM;                 // Value is stored in the Hand-Remotes EEPROM memory
  Menu[MONITOR].Sub[SET].State = SETTABLE;             // User can SET the value

  Menu[MONITOR].Option[0].Text = "Sel";                // Monitor "Sel" Pump at a time   - Option #0 = Selected
  Menu[MONITOR].Option[0].Value = 0;                   // "Sel" will be the value 0      - One = 0
  Menu[MONITOR].Option[1].Text = "All";                // Monitor "All" Pumps            - Option #1 = All
  Menu[MONITOR].Option[1].Value = 1;                   // "All" will be the value 1      - All = 1
  Menu[MONITOR].LastOptionIdx = 1;                     // Last Option Index defined      - Number of Options - 1

  //------------------------------------------------------------------------------------------------------------
  Menu[PUMPIDX].Text = "Pump";                         // Menu Item used to select the Pump-Controller
  Menu[PUMPIDX].Location = HAND_EPROM;                 // Selected pump is stored in EEPROM
  Menu[PUMPIDX].Sub[SET].State = SETTABLE;             // Allow this value to be 'SET' by the user

  // Define the Pump-Controllers ( A maximum of 4; index 0->3 )
  // Check that "#define MAXOPTIONS" allows the number of indexes used ( default; 2 )
  Menu[PUMPIDX].Option[0].Text = "Canal";              // Pump can be "Canal"            - Option #0 = Canal
  Menu[PUMPIDX].Option[0].Value = 10;                  // "Canal" Value=TRANSCEIVER_ID   - Canal = TransceiverID #10
  Menu[PUMPIDX].Option[1].Text = "Ditch";              // Pump can be "Ditch"            - Option #1 = Ditch
  Menu[PUMPIDX].Option[1].Value = 11;                  // "Ditch" Value=TRANSCEIVER_ID   - Ditch = TrasceiverID #11
  Menu[PUMPIDX].LastOptionIdx = 1;                     // Last Pump Option Index         - # of Pump-Controllers - 1

  //----------------------------------------
```

```
  Menu[2].Text = "Battery(B)";                     // Create a menu item for monitoring the Battery
  Menu[2].Location = HAND_PIN+ A+1;                // Battery level is gotten from the Hand-Remote pin A1
  Menu[2].Sub[LOALARM].ID = 'b';                   // A Low Alarm is identified by a lower-case 'b'

  //vvvvvv[ PUMP-SPECIFIC MENU ITEMS ]vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
  //-----------------------------------------
  Menu[3].Text = "Power(P)";                       // Create a menu item for Power Control
  Menu[3].Location = ALL_PUMPS_PIN+ D+7;           // Power is set/got on all Pump-Controller's on pin [D7]

  Menu[3].Sub[SET].State = SETTABLE;               // Allow this Value to be 'SET' by the user
  Menu[3].Sub[LOALARM].ID = 'p';                   // A Low Alarm is identified by a lower-case 'p'
  Menu[3].Sub[HIALARM].ID = 'P';                   // A High Alarm is identified by an upper-case 'P'

  Menu[3].Option[0].Text = "Off";                  // Power can be "Off"             - Option #0 = Off
  Menu[3].Option[0].Value = LOW;                   // "Off" will be the value 'LOW"  - Off = LOW
  Menu[3].Option[1].Text = "On";                   // Power can be "On"              - Option #1 = On
  Menu[3].Option[1].Value = HIGH;                  // "On" will be the value 'HIGH'  - On = HIGH
  Menu[3].LastOptionIdx = 1;                       // Last Option Index defined      - Number of Options - 1

  //-----------------------------------------
  Menu[4].Text = "Water (L)";                      // Create a menu item for Water Level Transducer
  Menu[4].Location = PUMP1_PIN+ A+64;              // Water Level is read from VIRTUAL (Pump-Controllers firmware) pin 64
  Menu[4].Sub[LOALARM].ID = 'l';                   // A Low Alarm is identified by a lower-case 'l'
  Menu[4].Sub[HIALARM].ID = 'L';                   // A High Alarm is identified by an upper-cse 'L'

  //-----------------------------------------
  Menu[5].Text = "Pressure(R)";                    // Create a menu item for the Primary Pressure Transducer
  Menu[5].Location = ALL_PUMPS_PIN+ A+3;           // The 'signal' is gotten on all Pump-Controllers on pin [A3]
  Menu[5].Sub[LOALARM].ID='r';                     // A Low Pressure alarm is identified by a lower-case 'r'
  Menu[5].Sub[HIALARM].ID='R';                     // A High Pressure alarm is identified by an upper-case 'R'

  //-----------------------------------------
  Menu[6].Text = "Pressure(S)";                    // Create a menu item for the Secondary Pressure Transducer
  Menu[6].Location = PUMP0_PIN+ A+4;               // Menu-item is for Pump-Option #0 (Canal) on Pin (A4)
  Menu[6].Sub[LOALARM].ID='s';                     // A Low Pressure alarm is identified by a lower-case 's'
  Menu[6].Sub[HIALARM].ID='S';                     // A High Pressure alarm is identified by an upper-case 'S'

  //------------[ Start-Up the Display ( DO NOT CHANGE! )]-------------
  GetItem(PUMPIDX);                                // Read the last selected Pump-Controller from EEPROM
  idx = STARTIDX;SubIdx=MAIN;                      // Set where the Menu will start
  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.

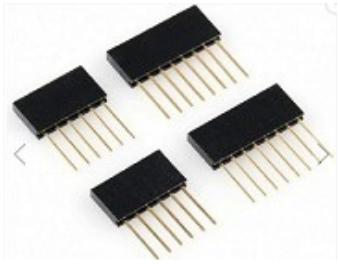
http://forum.arduino.cc/index.php?topic=376384.0

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)

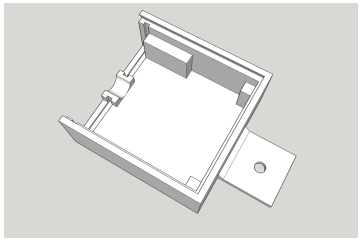


(1) 4-PIN Female Dupont Header
to connect to Circuit Board Pins
(See Dupont Headers in BOM)

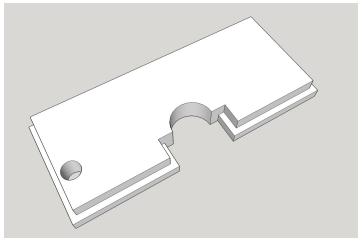
(~30-feet) 4-lead outdoor wire
required to reach from the ditch to the Pump-Controller
(See Outdoor Data Wire in BOM)

B. 3D-Prints ($1.30)

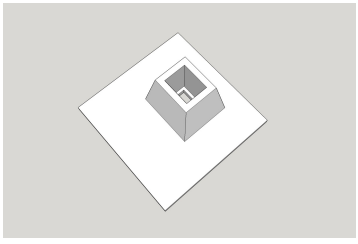
✔ Print the Following Models on a 3D-Printer
- Files Located at https://github.com/tgit23/AgIrrigationRemoteControl/tree/master/Accessories/Sonic%20Water%20Level%20Meter/3D-Prints-STL
- Total Filament = 43.7 cm^3 @ $0.03/$cm^3$ ~ $1.30



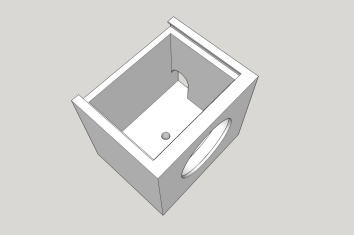
SonicCircuitBoardCase.stl (14.3 cm^3)



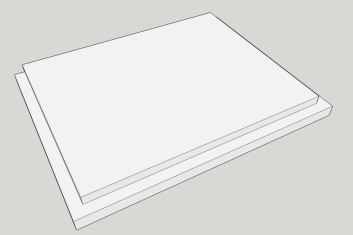
SonicCableCover.stl (2.5 cm^3)



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