

Instructions for building a 4 plot irrigation controller using an Arduino Uno

Rhuanito Soranz Ferrarezi and Marc van Iersel, hortphys.uga.edu

Step 1: Stacking the Adafruit datalogging board

This board has two functions:

- 1) It has an SD card holder and the SD card will be used to store all data collected by the system. Obviously, an SD card needs to be inserted for this to work
- 2) The board has a 'real-time clock' (RTC) built in. This helps the system keep track of time. A small coin battery needs to be inserted into the battery holder. The system may malfunction without a battery, so it is best to never remove this battery. Keeping the battery inserted also assures that the RTC will be able to keep time, even if the power to the system is disconnected. The RTC uses analog pins A4 and A5: these pins can thus NOT be used for anything else.

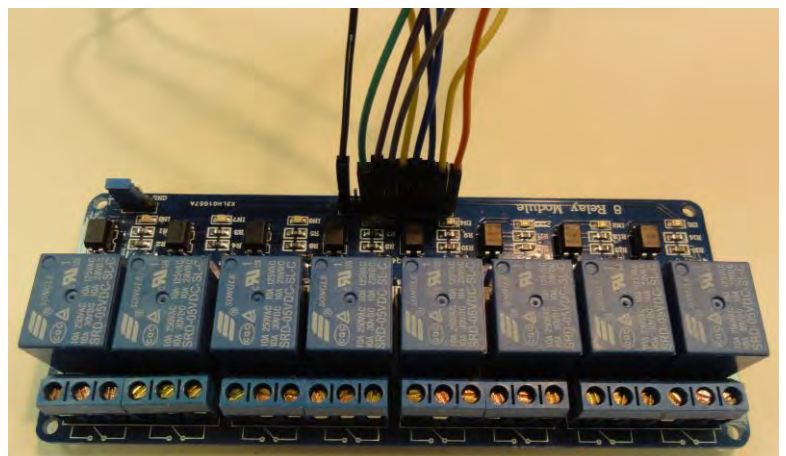
The datalogging board is designed to be plugged onto the Arduino Uno, which will make all of the needed connections. Make sure to align all the pins correctly. For more information about this board see: <http://learn.adafruit.com/adafruit-data-logger-shield/overview>

Step 2: Connecting a 8 relay module to the Arduino Uno

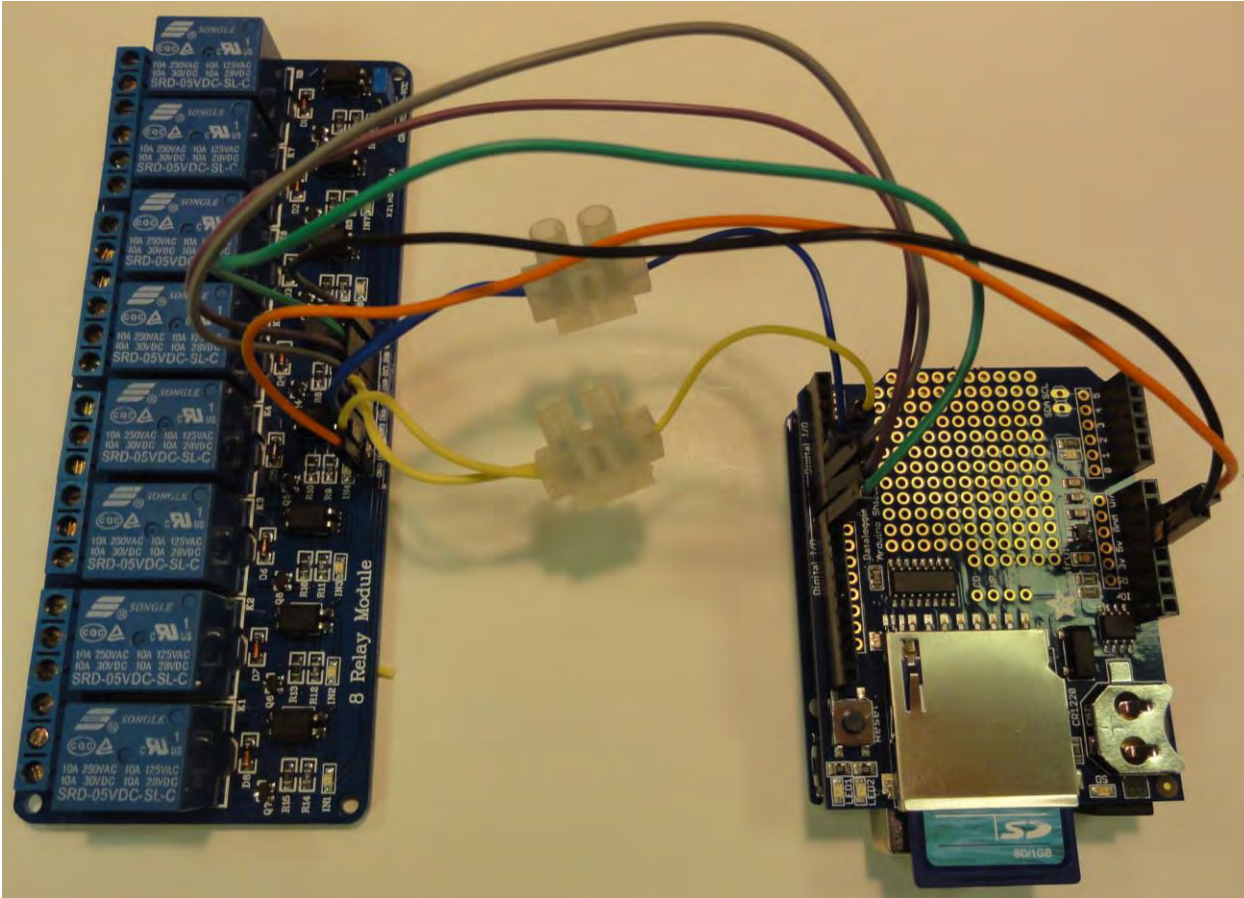
Note: the module has 8 relays, in this example we use only relays 1 - 7

Jumper cables: Ribbon wire (male – female) with individual jumper wires. Color does not matter, but in these instructions, we use orange and red wires for power connections (+5 VDC) and black wires for GND connections.

At the left side of the Adafruit datalogging board, you will find digital pins D2 - D9 (top left in picture below). At the right side, 5V and GND pins (middle right in picture).



Make the following connections between the Arduino Uno and the relay module: an orange jumper cable from 5V on the Arduino Uno to 5V on the relay module. One black jumper cable from GND on the Arduino Uno to GND on the relay module. Tie relays 1 and 4 together using a yellow wire and relays 2 and 3 using a blue wire (here with 'euro-style' connectors, you can use wiring nuts as well). Connect the yellow wire to digital pin D4 and the blue wire to digital pin D5. Connect relay 5 to digital pin D6 using a gray wire, relay 6 to digital pin D7 using a purple wire, and relay 7 to digital pin D8 using a green wire. The final connections should look like the picture below:



Step 3: Connecting the 10HS soil moisture sensors to the Arduino Uno

The 10HS sensors have three wires: red, white, and bare. The red wire provides a voltage output that will be measured by the Arduino, the white wire provides power to the sensor, and the bare wire needs to be connected to GND. Number the sensors 1 - 4.

Connecting the red wires:

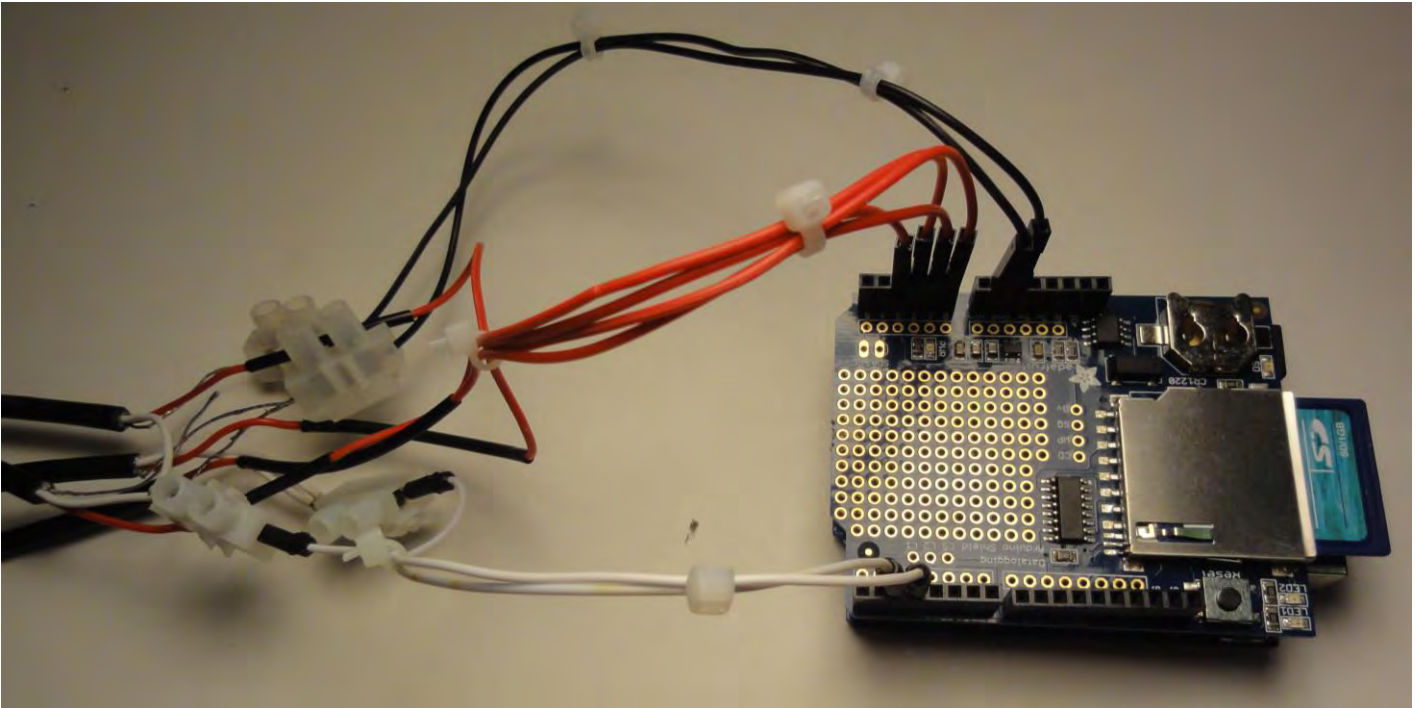
Attach red jumper wires to the red wire from each sensor. You do need to keep track of which wire is connected to which sensor. Connect the wires from sensors 1 - 4 to analog pins A0-A3 on the Adafruit datalogging board on top of Arduino Uno.

White wires from the sensors:

Connect the white wires from sensor 1 and 2 together (using "euro-style" terminal blocks or wiring nuts) and connect a jumper from the two white wires to digital pin D2 on the Arduino Mega. Likewise, connect the white wires from sensor 3 and 4 to digital pin D3.

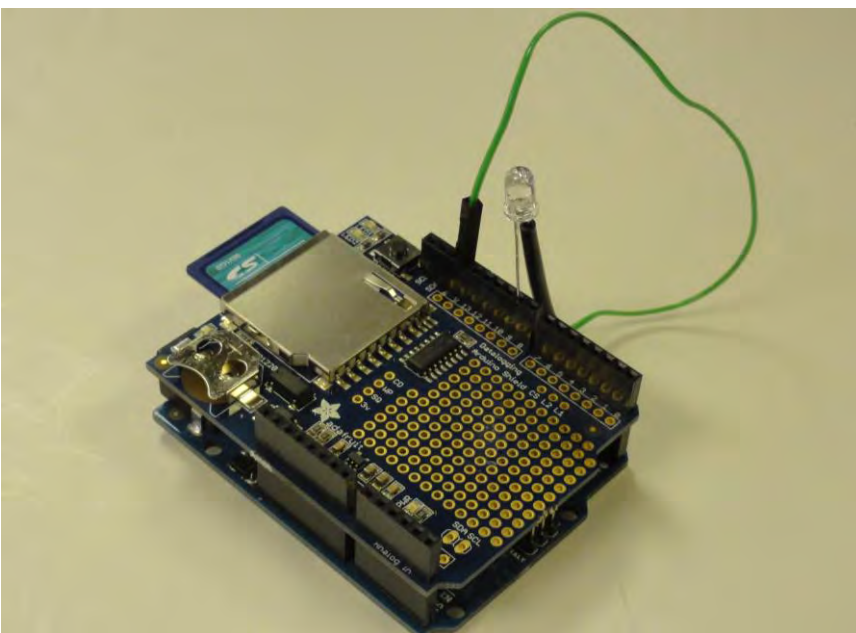
Bare (ground) wires:

All bare wires from the sensors need to be connected to a GND on the datalogging shield. We did this by connecting the bare wires from sensor 1 - 4 together (using a 'euro-style' connector) and connecting those to a green jumper wire.



Step 4: Connecting the LED

Although not needed for the system to work, connecting a LED to the setup is useful for error checking. This LED will come on when the sensors give a volumetric water content reading outside the normal range (less than 0 or more than $0.8 \text{ m}^3 \text{ m}^{-3}$). The shorter wire (negative) of the LED is connected to a 4,600 Ohm resistor. The exact resistance does not matter; anything from 500 to 10,000 Ohm should work fine. All exposed wire is then covered with heat shrink tubing.



The bare wire on the right comes from the resistor and gets inserted into a GND on the datalogging shield (not clearly labeled, but right next to digital pin D13). The long wire from the LED is connected to digital pin D9 (directly insert the LED wires into the pin, without using any additional wire).

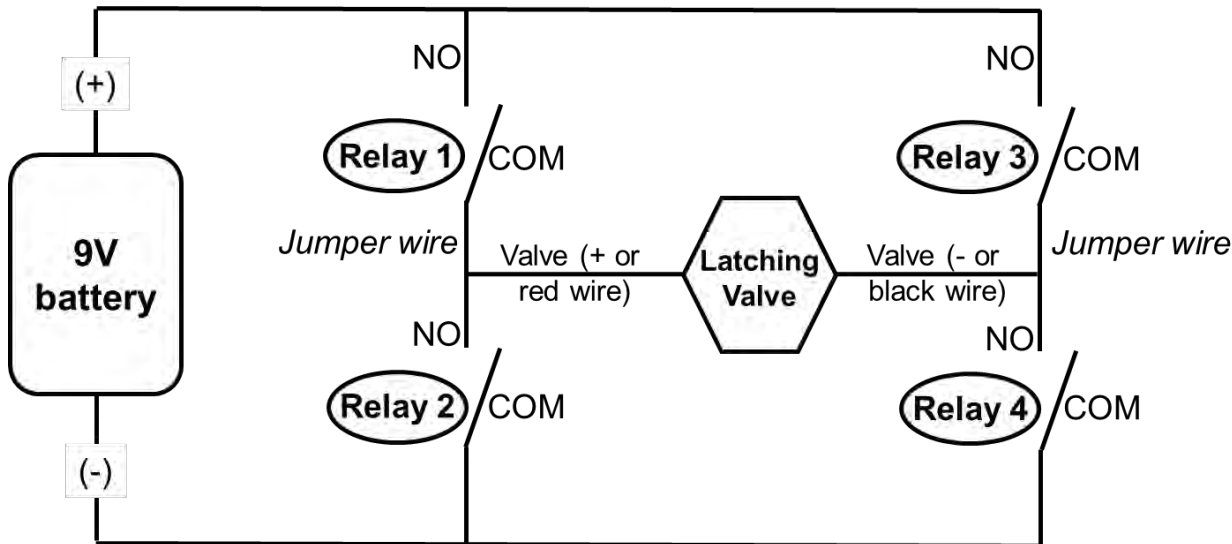
After making all connections, and making sure the connections are correct, use electrical tape to secure the wires to the boards. You may want to do this after testing the entire setup.

Step 5: Connecting the solenoid valves

The relay board needs to be powered continuously by connecting the 5V and GND on the relay board to the 5V and GND on the SD shield using jumper wires.

Latching solenoid valves

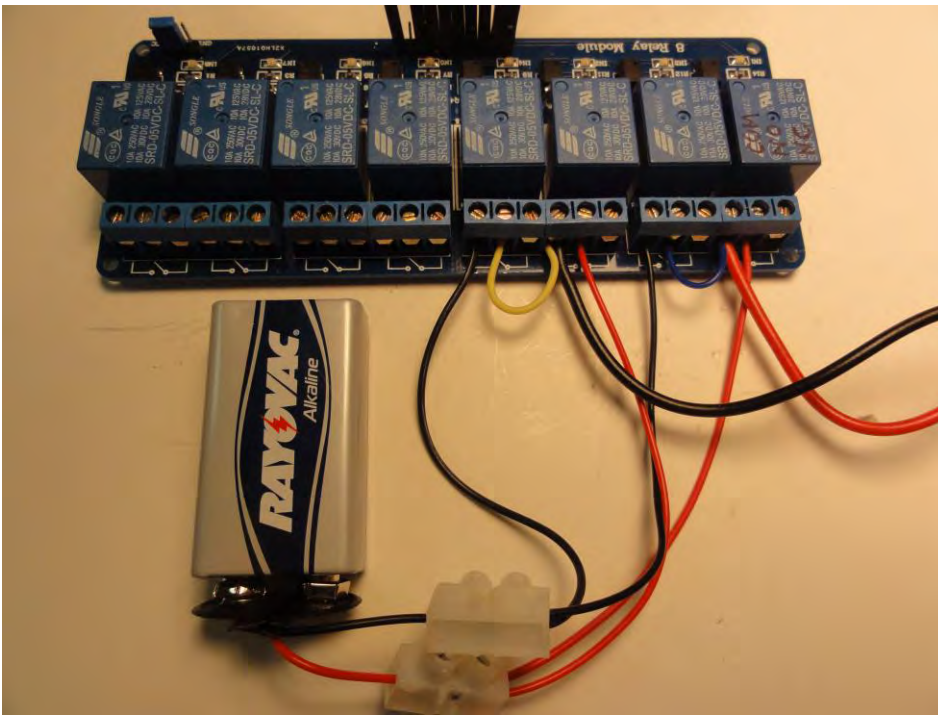
Latching solenoid valves are ideal for applications where power is limited, because continuous power is not required to maintain the open position. To reduce power consumption, an internal magnet can hold the solenoid in the open position. These valves also require a short pulse (50 ms) to open or close, which can be delivered using a 9V battery. To open the valve, the red wire from the valve needs to be connected briefly to the positive wire of the power supply (9V battery in this case), and the black wire to the negative one. To close the valve, the polarity needs to be reversed, meaning that the red wire from the valve needs to be connected to the negative side of the power supply and the black wire to the positive side. We used four relays to build an H-Bridge, which can be used to do this.



Electrical diagram of an H-Bridge with four relays, designed to control power to a latching solenoid valve. When the COM and NO terminals of relays 1 and 4 are connected (and those of relays 2 and 3 are not), +9 VDC will be applied across the latching valve, opening the valve. When the COM and NO terminals of relays 2 and 3 are connected, -9 VDC is applied, closing the valve.

The control pins for relays 1 and 4 (pins K1 and K4) both are connected to Arduino digital pin D4 and the control pins for relays 2 and 3 (K2 and K3) to Arduino digital pin D5. We used an H-Bridge (made with four-relays) and one 9V battery to be able to reverse the polarity and open and close the latching solenoid valve:

- Relay 1: Latching valve positive (red) wire to COM, battery positive to normally open terminal (NO)
- Relay 2: Battery negative to COM, jumper wire from NO to relay 1 COM (thus connecting to the red wire of the valve)
- Relay 3: Latching solenoid valve negative (black) wire to COM, battery positive (red) to normally open terminal NO
- Relay 4: Battery negative wire to COM, jumper wire from NO to relay 3 COM (thus connecting to the black wire of the valve)



Regular solenoid valves

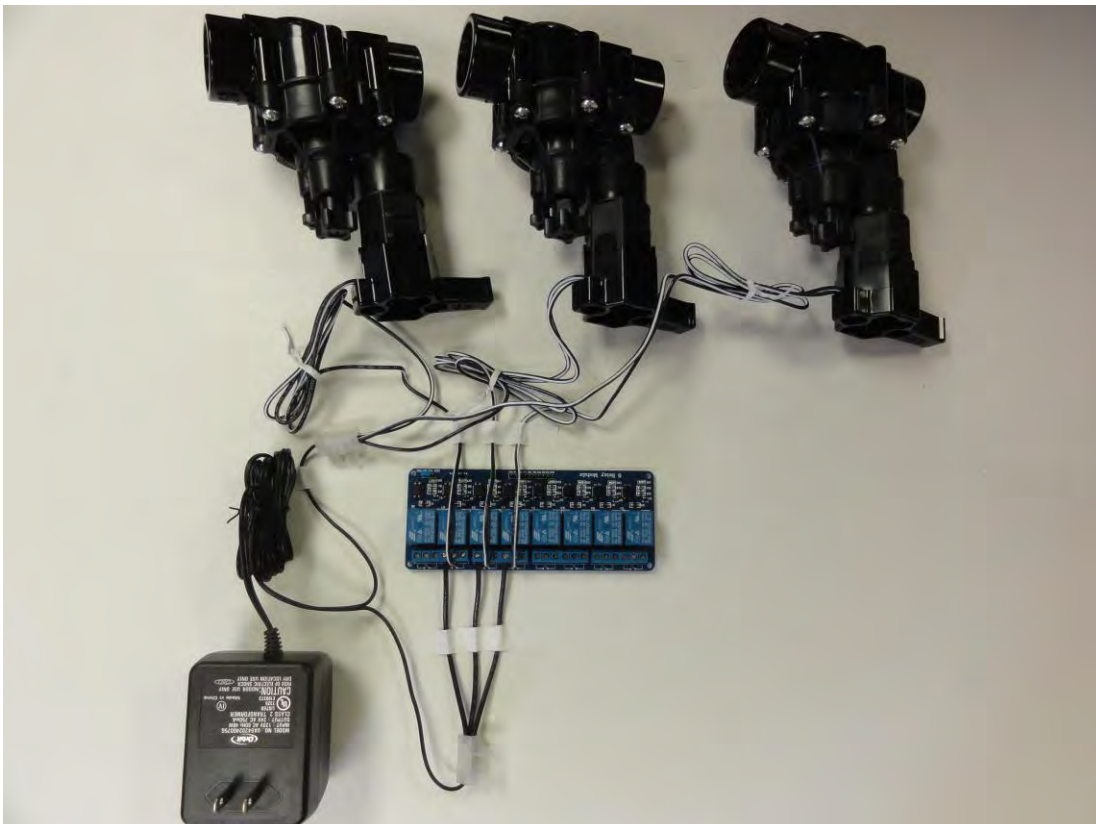
We are showing just one valve because all valves are connected the same way. The control pin of relay 5 (K5) needs to be connected to Arduino Uno digital pin D6, pin K6 (controlling relay 6) to Arduino digital pin D7, and pin K7 (controlling relay 7) to Arduino digital pin D8. One wire from the 24VAC power supply has to be divided into three wires and be connected to one of the wires of valves 2, 3, and 4. It does not matter which valve wire is connected, since these valves do not have any particular polarity. The other wire from the 24VAC power supply also has to be split in three wires and these are connected to the COM of relays 5, 6 and 7.

- Relay 5: One wire from valve 2 to NO, one wire from 24VAC power supply to COM
- Relay 6: One wire from valve 3 to NO, one wire from 24VAC power supply to COM
- Relay 7: One wire from valve 4 to NO, one wire from 24VAC power supply to COM

One solenoid and 24VAC adapter connected to a relay:



Three regular solenoid valves and 24VAC adapter connected to the relay board:



View of one latching and one regular solenoid valves connected to relays

