OpenSprinkler Bee (OSBee) Arduino Shield v1.0 User Manual

(*Updated May 24, 2014*)

Contents

Introduction

OpenSprinkler Bee (OSBee) Arduino shield is an open-source Arduino shield for switching battery-operated, latching sprinkler valves. It's an easy and low-cost solution to transform your Arduino into a flexible sprinkler controller with 4 independent zones, for garden and lawn watering, plant and flower irrigation, and other watering applications. By stacking additional Arduino shields, you can enable web-based control or wireless control as well.

The OSBee Arduino Shield includes one assembled and tested circuit board, and terminal blocks.

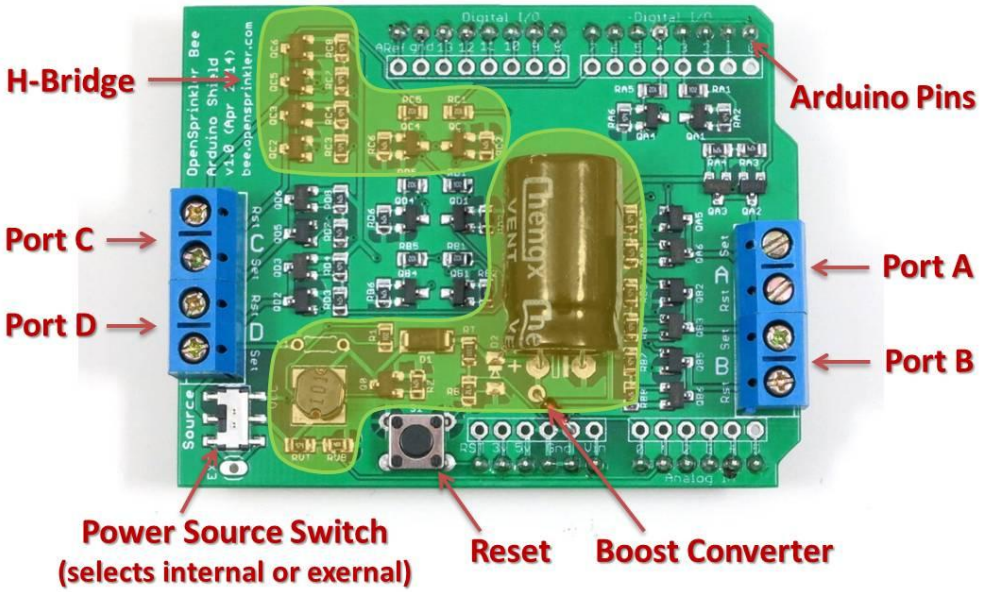
To get started, you will also need the following, which are **NOT included in the kit** and have to be purchased separately.

- An Arduino (any version)
- Up to 4 battery-operated, latching valves (**note: DC latching valves only, not AC!**). Examples:
  - Orbit 58874N Yard Watering Valve (24V, garden-hose thread, or GHT)
  - Orbit 57861 battery-operated solenoid for inline valves (17V, inline NPT)
  - Hunter 9V DC solenoid (9V, inline NPT)

*Different from AC valves, battery-operated, latching DC valves usually **have special plugs, or have two wires that are colored differently**. Each valve uses a latching solenoid and only draws power when it opens or closes. No power is drawn when it remains in the same state. The latching solenoid is made of a single coil: apply a positive voltage is on the coil, the valve opens; reverse the polarity of the voltage, the valve closes. A high impulse current (up to a couple amperes) is usually required to reliably open or close the valve.*

Hardware Interface

The image below shows a close-up image of the OSBee Shield and marks the built-in components.



Hardware Setup

1. Attach OSBee Shield to Arduino

Insert the OSBee Shield into the Arduino. If you have other Arduino shields you'd like to stack, such as a WiFi shield or an Ethernet shield, keep OSBee Shield on the top layer. You need to find out the pins reserved by the other shields (for example, the Ethernet shield uses pin 10 for Chip Select), and leave the corresponding pins out on the OSBee Shield. Details are explained later in this document.

2. Select the Power Source

The power source switch (lower-left corner) is used to select between Arduino's VCC pin (internal) or a battery (external). When using external source, solder a wire from the 'EXT' pin to your battery's positive lead, and any 'GND' pin to your battery's negative lead. The internal source is useful for testing, but keep in mind that some Arduino's (particularly the Pro Mini) have built-in fuse that prevents drawing high current from the VCC pin. During operation, the boost converter on OSBee Sheild needs to momentarily draw up to several hundred milliamps from the power source. To avoid triggering the fuse, using an external source (which doesn't limit the current draw) is recommended.

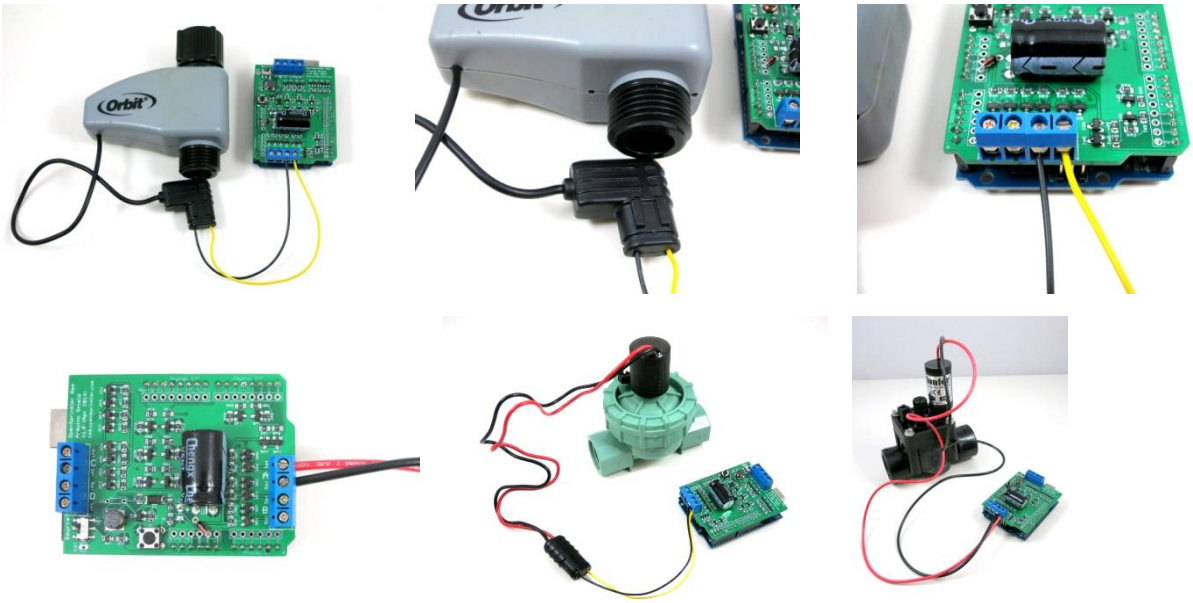
In addition, when set to internal source, the Arduino may not be able to program correctly**. If you see error messages when uploading a program, you should either unplug the shield, or set the source switch to external, until the program uploading is completed.**

3. Wiring Sprinkler Valves

OSBee Shield supports four independent zones. These are marked ports A, B, C and D. Each port has two

pins named **Set** and **Rst (Reset)**. Connect your sprinkler valve's **positive** wire (red colored) to the Set pin, and **negative** (black colored) to the Rst pin. If you accidentally reversed the polarity, you won't damage anything – it will just lead to reverse logic (i.e. 'opening' the valve actually closes it etc.).

If your sprinkler valve comes with a special plug, such as the Orbit 58874N valve, you can connect it as follow: the top pin on the plug is the positive wire, and the bottom pin is the negative wire. You can cut two pieces of wires (22 to 24 AWG recommended), strip the wires; then insert one end to the plug, the other end to the screw terminals on OSBee Shield, as shown in the picture on below.



Software Setup

1. Pin Assignment

OSBee Shield makes use of the following pins on the Arduino:

- Port A: **D3** (set), **D4** (rst)
- Port B: **D5** (set), **D6** (rst)
- Port C: **D7** (set), **D8** (rst)
- Port D: **D10** (set), **A2** / D16 (rst)
- **D9**: boost converter PWM pin
- **A0**: boost converter feedback pin
- **A1**: battery voltage feedback pin

Precaution: make sure you NEVER set the pair of pins on the same port high at the same time. For example, setting D3 and D4 high at the same time can cause damage to the Port A H-bridge.

2. Principle of Operation

During initialization, all port pins are set low. To open / close a valve, first use the boost converter to generate a voltage required by your valve (for example, Orbit valves usually require 17 to 24V, and Hunter valves 9V; refer to your valve's electric spec for details). To open a valve, make the **Set** pin high for 50 to 200 milliseconds, and then set it back to low; to close a valve, use the **Rst** pin instead. Again, **NEVER leave both the Set and Rst pins high at the same time** – that will damage the H-bridge.

If you are not sure about the correct voltage and timing, don't worry about it. Because the circuit builds up charges into the on-board capacitor and supplies the voltage to your valve from there, you won't damage your valve even if you used a voltage higher than necessary, or timing longer than necessary. There is no persistent high voltage on the circuit, and the capacitor can only store a limited amount of charge, so it's unlikely you will damage your valve. If the valve cannot properly open or close, just experiment with the parameters until you find settings that can reliably open / close the valve.

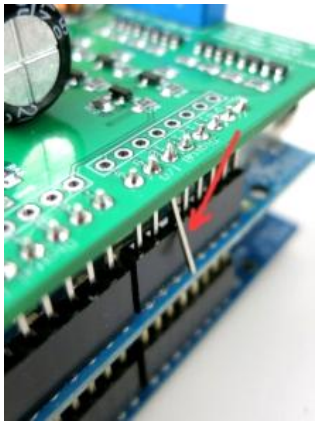
Please check the provided Demo Programs to see how to use the boost converter. Feel free to modify the programs, but do read the comments in the demo programs carefully, to understand what sections are absolutely necessary.

Warning: leaving the valve open unintentionally and not being able to close it can lead to water waste and even worse, flood damage. For best practice, periodically perform a 'close' operation on all valves to make sure they are off. Also, before opening a valve, you can use pin A1 to check the battery voltage and make sure the battery is in good condition to close the valve later.

3. Arduino Library and Demo Programs

An Arduino library for OSBee Shield is provided, with several start-up examples. One is a simple testing program that opens each valve for 10 seconds, one after another. The second is a program-based scheduling algorithm – it allows the user to use Serial input to set multiple programs, and open / close valves based on the program data. The third is a web-based program – it uses the standard Arduino Ethernet shield to create a simple web interface, and allows the user to click buttons on the webpage to open / close valves. More full-featured programs will be added in the future.

4. Using OSBee Shield with other Arduino Shields



It's possible to use OSBee Shield with other Arduino Shields, such as the Ethernet Shield, WiFi Shield etc. It's important to find out what pins the other shields require, and avoid any pin conflict. For example, the Ethernet shield uses D10 as Chip Select. Because D10 is also used by OSBee Shield for Port D Set pin, you have to leave this pin out on the OSBee Shield, which will disable Port D functionality. To leave the pin out, you can either cut this pin on OSBee Shield, or bend the pin so that it's not physically inserted into the female pin headers (see picture on the right). You can also cut the PCB trace of D10 (but this will result in permanent change).

Specifications

- **Source Voltage:** 2V DC to 9V DC.
  (Power source is selectable, either from Arduino's VCC or an external battery).
- **Boost Voltage:** up to 24V DC.
- **Over-voltage Protection:** 24V zener diode.
- **Output Current:** > 10A impulse; > 3A continuous.
- **Number of Zones:** 4 independent.
- **Size:** 70mm x 55mm x 28mm (2.75" x 2.2" x 1.1")
- **Weight:** 28g (1oz) w/o Arduino

Terms and Conditions

Open-Source Links

- **OSBee Shield Github Repository**
- **OSBee homepage**