Intel Wireless Irrigation Capstone

Austin Morlan, Bryan Martin, Dylan Johnson, Taylor Griffin

Abstract
Current irrigation methods consume a lot of water and require a lot of sensors. A low-cost solution was needed that could accurately determine how much water was lost due to evapotranspiration and replenish that amount of water. The result was a device that could accurately supply the amount of water lost while providing a potential reduction in total irrigation of 10.2%.

Introduction
Water is a precious resource and current irrigation practices are managing water consumption poorly. There should be more efficient ways to administer water without proving too costly to the user.

Irrigation should be scheduled in such a way that minimizes water consumption without compromising crop yield. Current irrigation controllers require frequent calibration and employ an array of environmental sensors that increase unit cost. The use of soil moisture sensors in current implementations is problematic due to sensor corrosion and variance in readings at different depths and locations.

There is a need for a low cost, low maintenance irrigation controller that can effectively manage water resources regardless of location, terrain, and local climate. Such a device would need to minimize the use of sensors to lower unit cost and be able to operate in areas without internet connectivity.

Needs Assessment
Irrigation policy needs to be determined and executed as inexpensively as possible.

In order to carry this out the device needs to:
- Accept user configuration
- Determine irrigation needs
- Actuate and measure irrigation flow

Customer Needs:
- Intuitive interface
- Autonomous irrigation of site

Objective Statement

Our device is intended to provide autonomous irrigation to a site selected by user. The user will need to install and configure the device. After the installation and configuration, the device will determine the irrigation needs of the site and determine irrigation schedule. The device will actuate irrigation flow to the site according to the irrigation needs.

- **Installation/Configuration:**
    - The device will be placed in the irrigation site and will be connected to irrigation source. The device will then be supplied latitude, calendar date, and size of site to be irrigated via smartphone app.
- **Determining irrigation needs:**
    - The device will calculate extraterrestrial radiation using the current day and latitude. The device will collect daily high and low ambient air temperature. The device will calculate evapotranspiration in mm/day using extraterrestrial radiation and temperature data. The device will detect precipitation events and use this data along with evapotranspiration to determine irrigation schedule.
- **Actuate irrigation:**
    - The device will actuate and measure irrigation flow at the time specified by configuration until the calculated need is met.

Requirements

Algorithm:

MUST

- Minimize inputs
- Minimize program size
- Determine irrigation policy
- Not use soil moisture sensors

SHOULD

- Be effective in as many environments as possible

Device:

MUST

- Acquire sensor data
- Compute algorithm
- Actuate flow
- Measure flow rate
- Not activate flow during precipitation event
- Temperature must be sensed within a degree
- Must know day of the year

SHOULD

- Be re-programmable while deployed
- Wake up, pull (sample the data), sleep
- Store sensor data
- Use BLE for communication
- Use a MPU comparable to the Atmel SAM3X(Arduino Due)
- Interact with users via smartphone app
- Battery life of at least 6 months

MAY

- Push sensor and flow data to the cloud
- Auto configure device

Design Considerations

Functionality

- The device will have a temperature sensor to record the ambient temperature throughout the day.
- The device will have a solenoid valve to deliver the water when activated.
- The device will have a flow sensor to help determine the amount of water being delivered.
- The device will have a rain drop sensor to check for precipitation events.

Economic

- The device will have a Cost of Ownership equal to the price of the unit.

Energy

- The device will run off of mains power
- Energy use is not of concern

Health & Safety

- The device will be enclosed within a sealed, weather-resistant case. This enclosure should never be opened during raining periods when outside.

Maintainability

- If the device ceases to function due to hardware failure, it will need to be replaced.

Manufacturability

- The device will use various passive components and the BL600 module.
- The device will have a sealed, weather-resistant case made of acrylic.

Operational

- The device will be intended for outdoor use when determining the algorithm for watering a field.
- The device will be weather-resistant due to the acrylic casing. The device will not be drop resistant.

Reliability & Availability

- The device will be a prototype
- The device will work as long as none of the hardware is damaged.

Social & Culture

- The device will have an Android app associated with it. This app will only be available in English.

Usability

- The device will power on automatically as soon as power is applied via power cord.
- The device's associated Android app will display various types of data after activation (temperature low/high, precipitation event, etc.).

Solution

The proposed solution uses water-budgeting irrigation scheduling. This method attempts to track the amount of water entering and leaving the soil through various processes. Our water-budgeting scheme tracks the following factors:

- Evaporation of water from soil to environment
- Transpiration of water from crops to environment
- Precipitation of water from environment to soil

Evaporation and transpiration are determined using the *Hargreaves Reduced-set Method* which estimates evapotranspiration using ambient air temperature, latitude, and day of the year. This method reduces the number of sensors to two: a temperature sensor and a precipitation sensor are all that are required since latitude and calendar day can be supplied during configuration of device via app.

Device:

- Minimum amount of sensors required for algorithm calculation
- Printed circuit board for component connections
- Enclosure to properly protect fragile components from the weather

Android App:

- Communicates with device via Bluetooth Low-Energy (BLE)
- Sends configuration information for irrigation policy (time, date, latitude)
- Receives irrigation statistics
- Displays sensor readings

Algorithm

The water-budget irrigation method can be summarized as:

$$Irr = E_{to} - P$$

Where Irr is the amount of water to be applied, Eto is the crop evapotranspiration, and P is precipitation. All values are in mm.

Eto is estimated using the *Hargreaves Reduced-set Method*:

$$E_{to} = HC \cdot R_a \cdot (T_{max} - T_{min})^{HE} \cdot (\frac{T_{max} + T_{min}}{2} + HT)$$

HC, HE, and HT are constants specified by the method and can be calibrated. Ra is the solar radiation that is calculated using the latitude and calendar day set during configuration of device.

The device samples temperature and precipitation sensors every hour. The precipitation sensor returns a wet or dry value in order to reduce complexity and cost. Eto calculations are performed daily using the daily high and low temperature values. This value is multiplied by the area to be irrigated in order to find the volume of water needed. A solenoid is then opened to actuate water flow and amount of water dispensed is measured using a hall-effect flow sensor.

Results

Over the 20-day testing period the device calculated a total evapotranspiration of 67.91mm and irrigated a total of 46.3 gallons of water. Over the same period a total of 10.6 gallons of water was applied to the vegetable bed via precipitation(2). A total of 56.9 gallons of water was applied to the vegetable bed. This result can be compared favorable to historical data and to general irrigation guidelines. 10-year historical evapotranspiration for Forest Grove, Oregon, over the same period is 84.4mm(3). General gardening guidelines(4) typically recommend 1” of watering per week plus an additional ½” weekly for average temperatures above 70 oF. Using these guidelines, a total of 63.4 gallons of water would need to be applied by either irrigation or precipitation.

Compared to the rule-of-thumb estimate of 63.4 gallons, our result of 56.9 gallons saved 6.5 gallons of water without compromising crop health or yield.

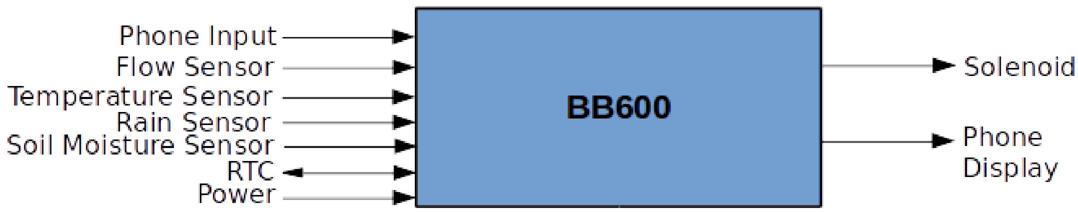
Conclusion

Overall, irrigation scheduling using water-budgeting and evapotranspiration estimations has been demonstrated successfully. The device calculated reasonable evapotranspiration values compared to historical data and also supplied a reasonable amount of irrigation water compared to general guidelines. While initial results are very promising, more testing data is required. Further testing should be conducted at sites varying in location, soil quality, and crops grown.

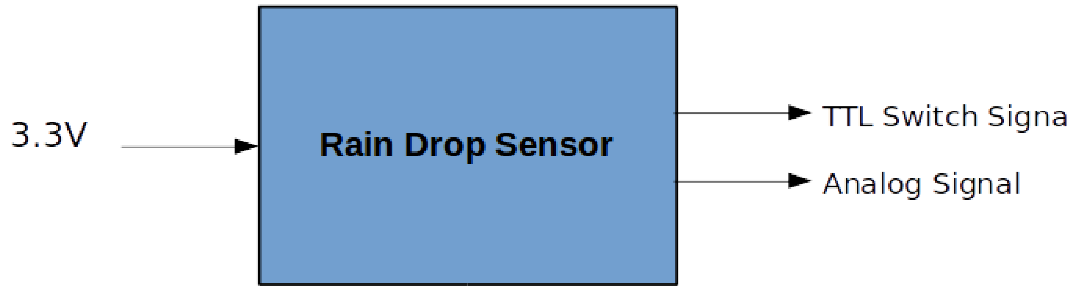
After collecting more testing data, it may be desirable to adjust constants used in calculations to compensate for differences in soil type, crops grown, environmental factors, etc.

The rain detection algorithm may be improved by scaling the evapotranspiration term by a wet-to-dry ratio rather than the current binary implementation.

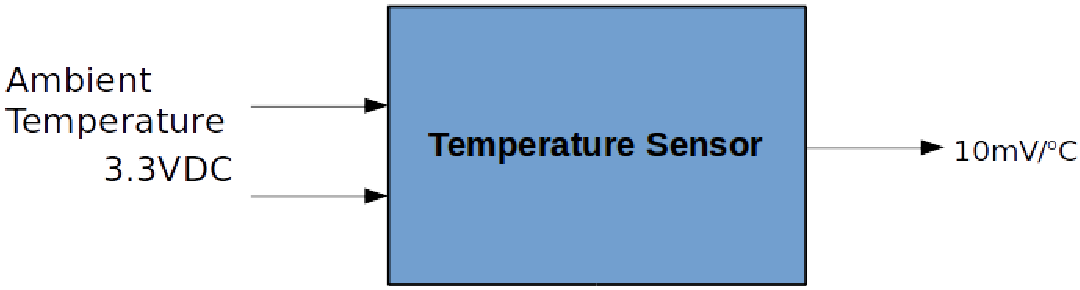
System Design Model



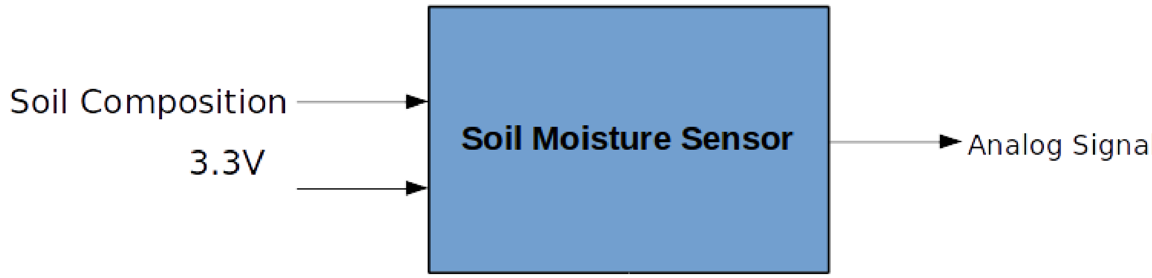
Module	BB600
Inputs	**Phone Input** – app on phone sends the values needed for the algorithm over BLE: latitude and calendar day **Flow Sensor** – sensor attached to spigot measures how much water has been distributed and reports back the value **Temperature Sensor** – measures ambient sensor used by moisture algorithm **Rain Sensor** – determines whether it is currently raining; if it's raining, suspend irrigation **Soil Moisture Sensor** – measures actual soil moisture for debugging and verification of algorithm **RTC** – once the phone app provides the calendar day the RTC ensures the algorithm is always provided with the correct day of the year **Power** – 1.8VDC – 3.6VDC
Outputs	**Solenoid** – holds a GPIO output high to turn the solenoid on when water starts flowing and holds a GPIO output low to turn it off **Phone Display** – sends information to app on phone for display
Functionality	Speaks to a phone app over BLE and a spigot actuator. It computes an algorithm using information provided by the phone app (latitude and calendar day) and the sensors (temperature) which provides an estimate of the soil moisture level. When the soil moisture level is low enough it turns on the spigot and keeps it on until the calculated amount of water has flowed into the soil. It reports its sensor information back to the phone app for display. If it rains it suspends water flow.



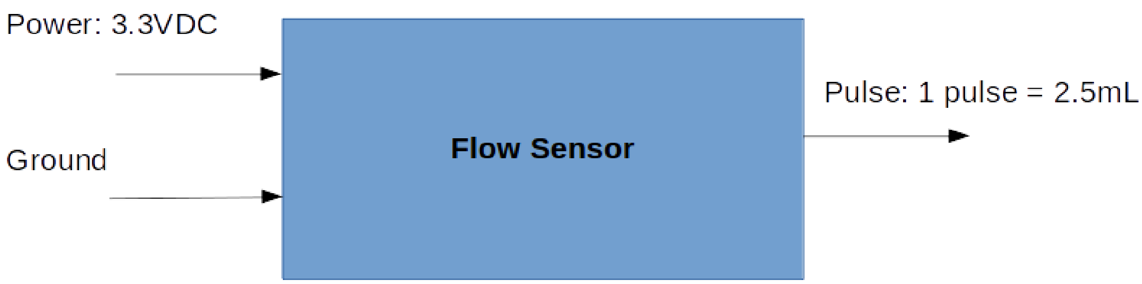
Module	Rain Drop Sensor
Inputs	3.3V DC
Outputs	TTL Switch Signal (Digital High/Low Signal) Analog Signal
Functionality	Detects water on the surface of the rain pad and constantly outputs separate digital and analog signals.



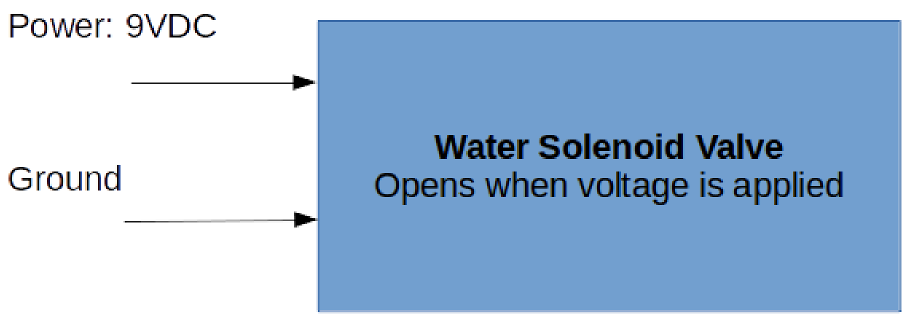
Module	Temperature Sensor
Inputs	Ambient Temperature 2.7V DC to 5.5V DC
Outputs	10mV per degree Celsius
Functionality	Outputs a voltage based on the sensed ambient temperature.



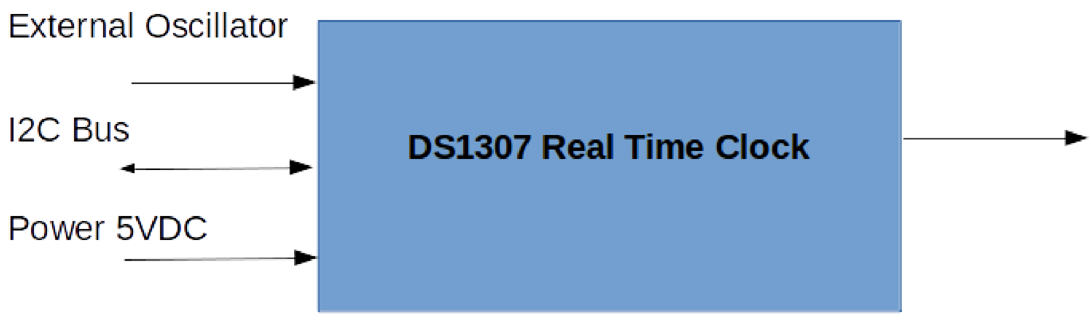
Module	Soil Moisture Sensor
Inputs	Soil Composition 3.3V DC
Outputs	Analog Signal
Functionality	Gives a value for the sensed soil moisture. This value is very dependent on the supplied voltage and the type of soil used. This device will need to be calibrated to detect dry and wet values if used in new areas.



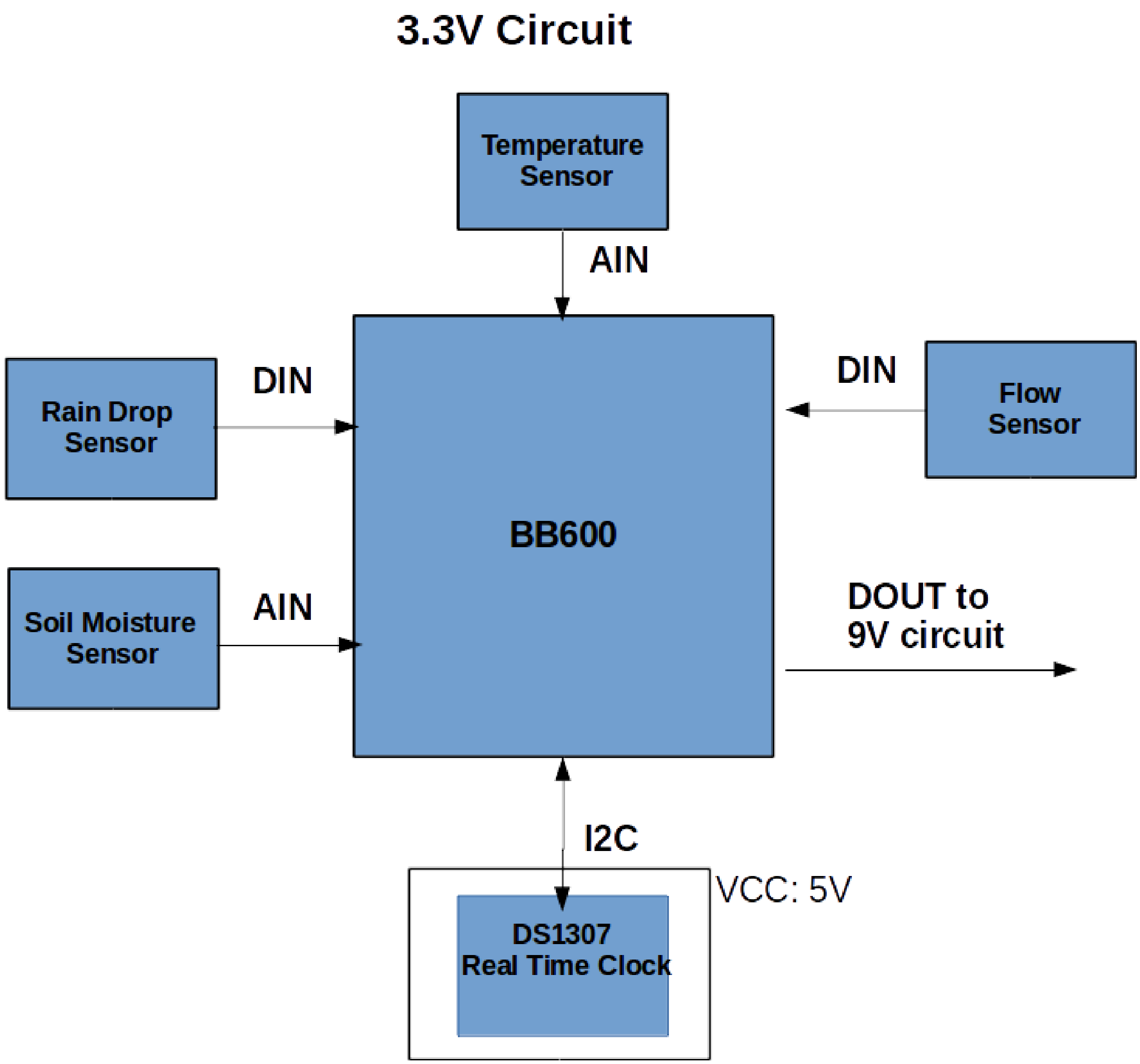
Module	Water Flow Sensor
Inputs	3.3VDC
Outputs	Signal output of pulses where each pulse should correspond to about 2.5mL of fluid dispensed.
Functionality	Outputs a voltage pulse for each 2.5mL of fluid dispensed.



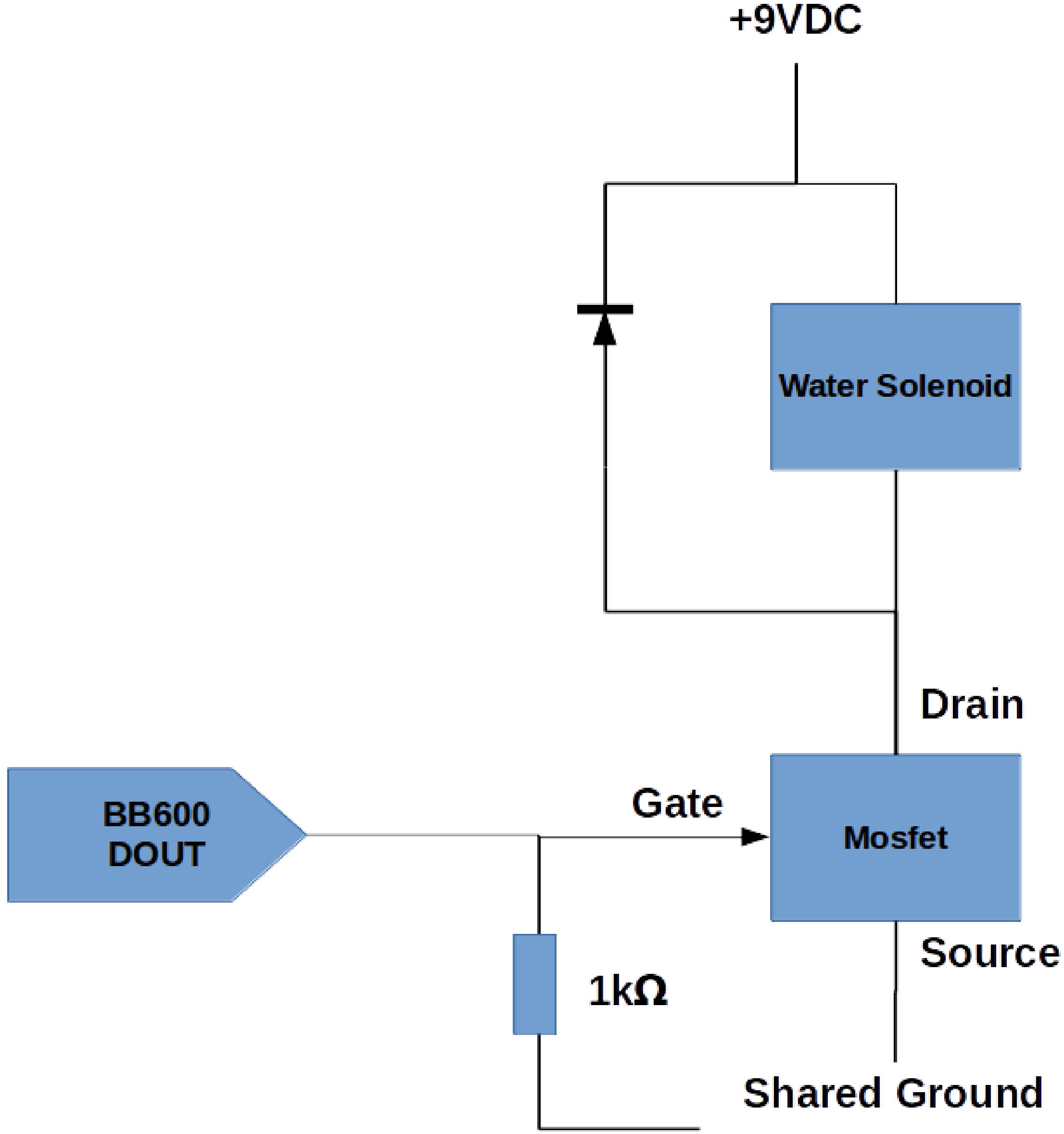
Module	Water Soleniod Valve
Inputs	9VDC
Outputs	None
Functionality	When a voltage is applied, the solenoid pulls open allowing water to flow. The valve is closed when no voltage is applied.



Module	DS1307 Real Time Clock
Inputs	5VDC, External Oscillator
Outputs	Square Wave output
Functionality	12 or 24 hour mode real time clock communicates hours:seconds:days over I2C interface.


3.3V Circuit
Temperature Sensor
AIN
Rain Drop Sensor
DIN
BB600
DIN
Flow Sensor
Soil Moisture Sensor
AIN
DOUT to 9V circuit
I2C
VCC: 5V
DS1307 Real Time Clock

9V Circuit


+9VDC
Water Solenoid
Drain
BB600
DOUT
Gate
Mosfet
Source
1kΩ
Shared Ground