Overview:

This project assumes the students have a working knowledge of the programming concepts in the Digital Mood Ring project. Please refer to that project as a review of concepts*. In the smart irrigation project, students are challenged to build and program a smart irrigation system based on the engineering concept of a feedback and control loop. The system uses three input modules; a temperature-humidity sensor, a soil moisture sensor, and an ambient light level sensor. The system has a single output to a power control unit with a water pump connected. A program is written to read all four input parameters and logically compare them with critical set-point values and among each other to determine when to turn the water pump on and off. The project is presented in a series of small challenges that build the knowledge and skills required for the final open-ended challenge. The final challenge also relies on the students' understanding of important biology and ecology topics that are relevant to optimizing the system the students ultimately design and refine. *Digital Mood Ring link http://bit.ly/2GBIsUN

Possible NGSS topics to explore with students:

Disciplinary Core Ideas
- MS-LS2-3,LS2-5-Ecosystems Interactions
- MS-ESS2-4 – Water Cycle
- MS-ETS1-2, MS-ETS1-4 Design and evaluate solution to problem

Crosscutting Concepts
- Stability & Change
- Cause & Effect

Science and Engineering Practices
- Constructing explanations & designing solutions
- Evaluate competing designs
- Asking questions & defining problems

Background:

Identify the problem

Humans invented agriculture over 6000 years ago as a way to produce more food than could be hunted and gathered from the environment. This increase in food availability produced a rapid increase in population that was totally dependent on agriculture for survival. Today, the world's large population requires sophisticatedlarge-scale agriculture to keep everyone fed. Climate changes can have severe consequences on the food production required to keep the large human populations fed. Natural and man-made ecological disasters, such as the Dust Bowl of the 1930's, can have severe consequences on the dependent populations. A more recent example, told in April 2016 The Guardian article*, describes the devastating effects on the local population of a severe drought in Zimbabwe. Science and technology can help to optimize food production and mitigate the effects of climate change and poor farming practices.

Build a Solution

In this project, students are challenged to use science and technology to design and build a system that utilizes a limited amount of water in a "smart" way to grow crops. To achieve this, they need STEM skills and knowledge from the disciplines of ecology, biology, chemistry, electronics, logic and computer programming. To enable students to complete this project, they are challenged to complete smaller tasks that will provide the programming and electronics knowledge required. In addition, during the project, the teacher has the opportunity to teach scientific principles in a relevant and meaningful setting. Some of the following science principles could be worked into the activity.
- The effect of temperature on the evaporation rate of water? Should water be used when it is hot or cool?
- How does soil type affect the percolation of water into the ground? Should water be delivered in a strong rush? Or should it be delivered slowly and in pulses?

- Does relative humidity affect the evaporation rate of water? Does warm air hold more water than cool air? What is relative humidity? Should water be delivered when relative humidity is high or low?
- How does the soil moisture level affect the rate that water can be absorbed? At what soil moisture level should water be delivered to crops?
- Does photosynthetic activity affect the uptake of water? Should plants be watered during daylight or night?
- Link to The Guardian article referenced above.

https://www.theguardian.com/global-development/2016/apr/21/drought-southern-africa-heavy-toll-students-fainting-malawi-zimbabwe

Commands used in this project

See TI-Innovator Technology eGuide for more background https://education.ti.com/html/webhelp/EG_Innovator/EN/index.html

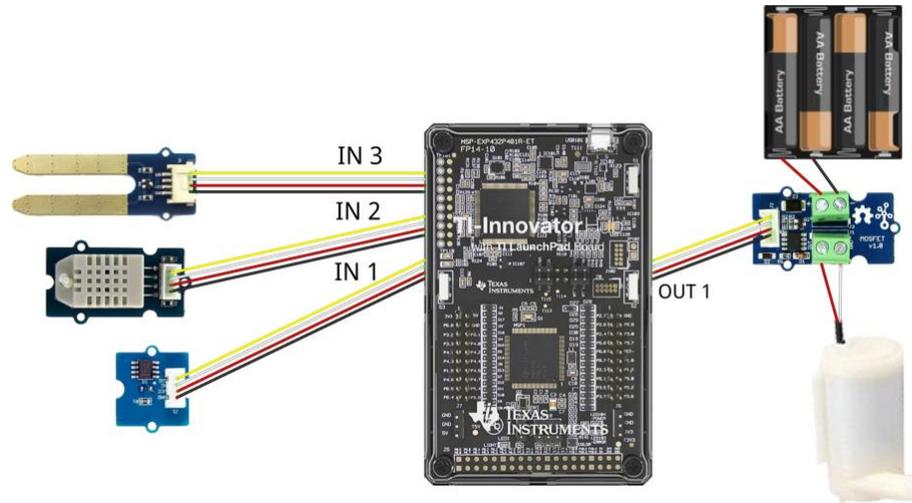| Command	Example	Behavior
CONNECT <type> <number> TO <port>	`Send("CONNECT LIGHTLEVEL 1 TO IN1")`	Associates the first LIGHTLEVEL object with a light sensor plugged into port IN1 on the Hub.
SET <type> <number> TO <value>	`Send("SET ANALOG.OUT 1 TO 128")`	Turns on an analog.out1 object, such as a pump, to a power setting of 128
RANGE <type> <number> <min value> <max value>	`Send("RANGE LIGHTLEVEL 1 0 100")`	Scales the measured values read from LIGHTLEVEL 1 to return in the range 0 to 100.
READ <type> <number>	`Send("READ MOISTURE 1")`	Reads one measurement from the first moisture sensor.
Get <variable>	`Get(M)`	Stores the moisture measurement into the variable named M. *Note a get command must immediately follow a read command. The value stored will contain the measurement from the immediately preceding READ command."
Output(<row 1-10>, <column 1-26>,<expression> <variable name> or <string>) **Note:** It is recommended to use the ClrHome command before using Output commands	`Output(3,1,"MOISTURE LEVEL= ")` `Output(3,17,M)`	When variable M has a value of 26, "MOISTURE LEVEL = 26" is displayed on line 3 of the calculator.

While <Boolean expression>
  <statements>
End
	`0→K`
	`While K≠45`
	`Send("READ MOISTURE 1")`
	`Get(M)`
	`Output(3,1,"MOISTURE LEVEL= ")`
	`Output(3,17,M)`
	`getKey→K`
		The commands inside the While structure are looped until the clear key is pressed. The loop continues while logical expression K ≠ 45, is true. The variable K is initially assigned a value of zero so the while loop will execute at least once. The getKey() function monitors the keypad and returns a two digit value of the row and column of the last key pressed. The clear key is located at row 4 and column 5.

| | ```
Wait 1
End
``` | |
| <Boolean expression 1> and <Boolean expression 2> | ```
If H≥ 50 and L<10
Then
Output(3,1,"PUMP IS ON")
End
``` | The "and" operator compares two separate Boolean expressions concurrently. When both expressions are true, the operation is true. |
| <Boolean expression> or <Boolean expression> | ```
If L< 50 or M<10
Then
Output(3,1,"PUMP IS ON")
End
``` | The "or" operator compares two separate Boolean expressions concurrently. When either or both expressions are true, the operation is true. |
| If <Boolean expression><br>Then<br>  <statements 1><br>Else<br>  <statements 2><br>End | ```
If L<20 and T<25 and H>80 or M<10
Then
Output(6,1,"GOOD TIME TO WATER")
Else
Output(6,1,"BAD TIME TO WATER ")
End
``` | The example decision tree has a Boolean expression with corresponding statements to execute if true. It also has an Else condition that executes corresponding statements when the Boolean expression is false. This Else condition ensures that a set of statements will always be executed. When this decision tree executes, focus proceeds from top-down. If the Boolean expression is true, the corresponding statements are executed and the decision tree is immediately exited. In the example, if T=30 then the first expression is false and the <statements 1> are skipped and the <statements 2> after the Else are executed. |

Setup Project: Supplies:

Students may work in groups of two, three or four.

- TI-Innovator Hub and cable
- TI Nspire calculator
- Grove - Temperature & Humidity Sensor
- Grove – Soil Moisture Sensor
- Grove – Light Sensor
- Grove – MOSFET for power control module with 4xAA battery holder
- 4 x AA batteries to provide power to the pump (required)
- Water Pump with the plastic tube
- Possible supplies for building garden model:
-   Drinking straws
-   Duct Tape (always useful)
-   Container for the plants, such as a 1-gallon milk
-   Soil, perlite or some other growth medium

Student Activity:	Teacher Notes:

Challenge 1: Write a program named C1 that continuously measures and displays the ambient light level. The program should:

1. connect a light level sensor to IN 1.
2. range the light level reading from 0 to 100.
3. use a while loop to continuously read and display light level every ½ second on the calculator screen.
4. enable the clear key to quit the while loop and end the program.

Teacher Guidance during Challenge 1:

- The light level sensor is a device that has a large range of responses to light intensity. As a result, the output is not linear with light intensity. In addition, the device does not measure with a particular unit. Instead, the device returns a raw value from 0 to 2^{14}. It is useful to use the RANGE command to scale the output from 0 to 2^{14} to 0 to 100.
- The sensor could be attached to any of the input ports 1, 2 or 3. It is essential the port the sensor is attached to is the same as the port used in the CONNECT command.
- Connect the sensor as indicated in the picture above.
- The ambient light level is an important biological and ecological factor that affects phototaxis, primary productivity, photosynthetic rate, and ambient heating.
- Discuss with students how the ambient light level can affect when to water a plant.
- Use the command descriptions described above to review the use of a while loop along with getKey to continuously perform a task.
- Use the command descriptions above to review the use of the Output(command for displaying values on the screen.
- Discuss with students how the use of the Wait command affects the rate the system monitors light level.

Example program:

```
PROGRAM: C1
Send("CONNECT LIGHTLEVEL 1 TO IN 1 ")
Send("RANGE LIGHTLEVEL 1 0 100")
ClrHome
0→K
Output(8,1,"PRESS CLEAR TO QUIT")

While K≠45
Send("READ LIGHTLEVEL 1")
Get(L)
Output(3,1,"LIGHT LEVEL= ")
Output(3,14,L)
Wait .5
getKey→K
End
```

Challenge 2: Write a program named C2 that measures soil moisture every two seconds for a total of twenty times. The program should display if the soil is dry or moist based on the sensor reading.

Teacher Guidance during Challenge 2:

- In Challenge 1, all of the necessary steps of the program were given to the students. In Challenge 2, help the students to analyze the task and make a list of the necessary step to accomplish the task.
- Discuss the differences in the control structure between Challenge 1 and Challenge 2. The While loop provides indefinite program execution and the For loop provides definite program execution.
- The soil moisture sensor produces an output from 0 to 2^{14} based on the conductivity (dielectric permittivity) of the soil which is dependent on water content. The sensor output is proportional with the volumetric water content; however, it is not calibrated in a particular unit or percentage. Like the light level sensor, it is useful to use the RANGE function to scale the sensor output from 0 to 100.
- The sensor could be attached to any of the input ports 1, 2 or 3. It is essential the port the sensor is attached to is the same as the port used in the CONNECT command.
- Connect the sensor as indicated in the picture above.
- Discuss the different types of soils; clay, sand, and loam and how water interacts with each. Features to consider are the rate of absorption, water capacity, compaction, and runoff.

Clay	Sand	Loam
Soaks in water slowly	Soaks in water very quickly	Soaks moderately
Holds water very well	Holds water poorly	Holds water moderately
Fine grain texture	Coarse grain texture	Mixed grain texture
Severe erosion	Low erosion	Low erosion

- Discuss the best conditions for plant soil watering. Should the soil be allowed to dry out between watering? Do different plants require different condition?
- As an extension, you may determine values to indicate the relative ranking of soil moistures. To do this:
  1. Place soil into three different pots labeled, dry, moist and soaked.
  2. Add water to each pot to the labeled soil moisture.
  3. Consistently insert the soil moisture sensor tines (prongs) halfway into the soil of the pot labeled "dry" and run the sensor program and record the displayed soil moisture value.
  4. Repeat for the other two soil moisture levels.
  5. Use the three measured values in your program to rank the soil moisture of your plants and decide when to water the plant.

Example program:

```
PROGRAM: C2
Send("CONNECT MOISTURE 1 TO IN 3")
Send("RANGE MOISTURE 1 0 100")
ClrHome

For(N,1,20)
Send("READ MOISTURE 1")
Get(M)
Output(3,1,"MOISTURE LEVEL= ")
Output(3,17,M)
If M<10
Then
Output(5,1,"THE SOIL IS DRY  ")
Else
DispAt 5,1,"THE SOIL IS MOIST")
End
```
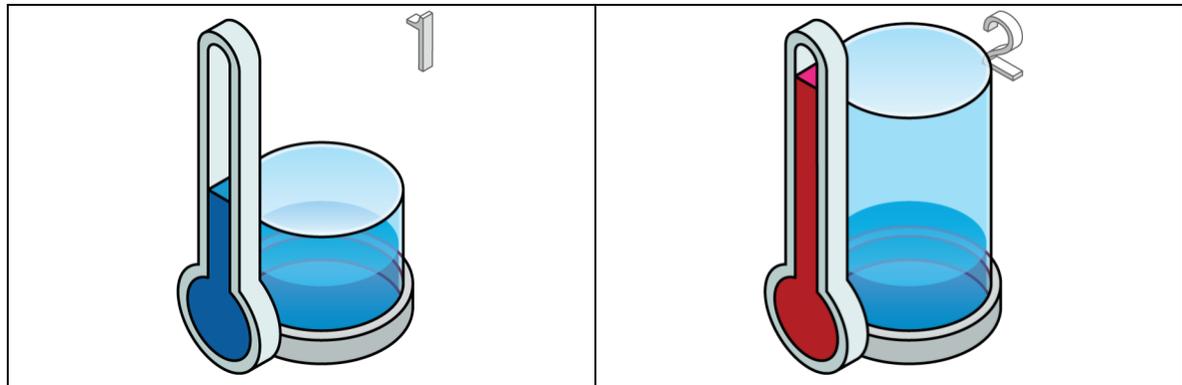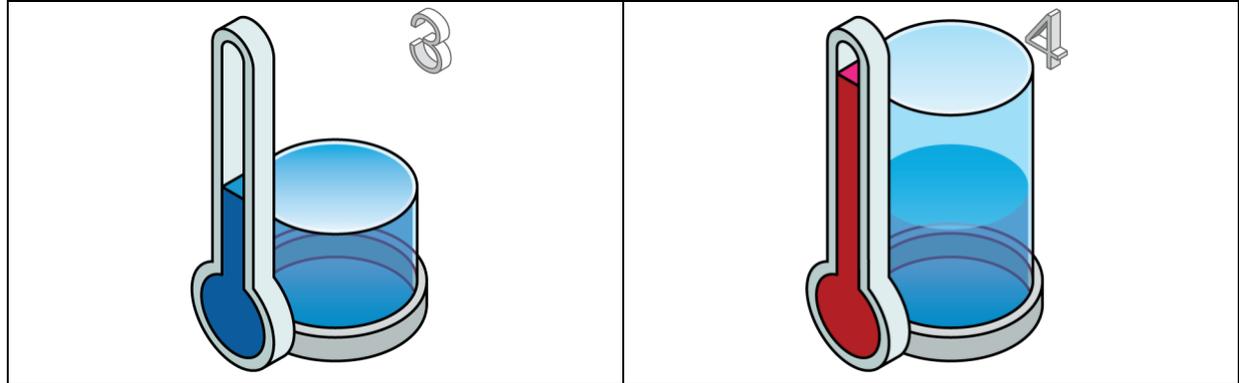
```
Wait 2
End
```

Challenge 3: Write a program named C3 that connects the Digital Temperature and Humidity (DHT) sensor. Read 20 measurements at two-second intervals and display with an appropriate message. Use a decision tree based on the temperature and relative humidity measurements to determine the present watering conditions as indicated in the graph below. Display an appropriate message with each of the four cases.

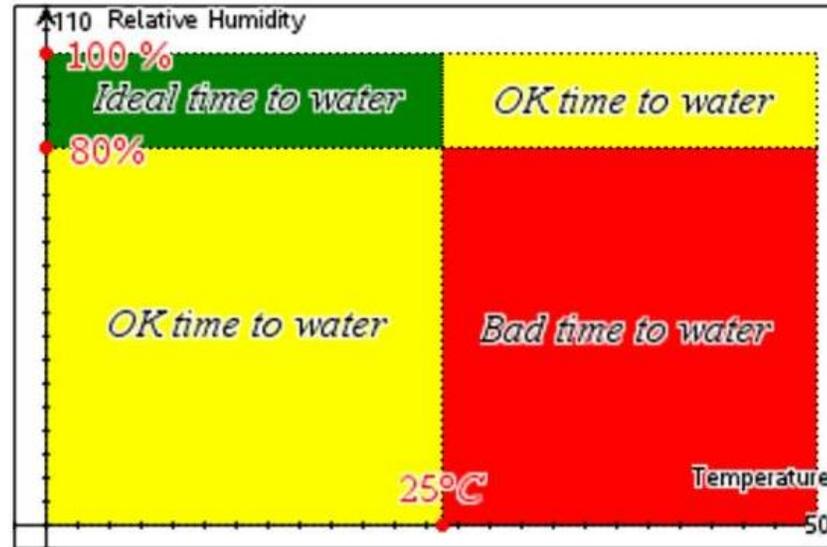**Teacher Guidance during Challenge 3:**

- The amount of water from the ground that evaporates into the air is dependent on the ambient temperature and relative humidity.
- For a particular temperature, there is a maximum amount of water that can evaporate and exist as vapor in the air. This is referred to as the saturation vapor pressure and is calculated using the Clausius-Clapeyron relation.
- Image #1 illustrates when the temperature is low, the corresponding maximum amount of water vapor the air can hold (saturation vapor pressure) is also low as represented by the small container. In image #2, the temperature is high and the maximum amount of water vapor the air can hold is also high as shown by the larger container.

- On any given day, the air will contain a certain amount of water, this is the water vapor pressure as illustrated by the darker blue volume in images 1, 2, 3, and 4. The water vapor pressure may be less than or equal to the maximum saturation vapor pressure.
- Image #3 illustrates a cool day with much water in the air. Notice the container is small and the water in the air just fills its container. Contrast with image #4 that illustrates a warm day with the same amount of water in the air as image #3.

- The relative humidity in image 3 is 100% because the amount of vapor in the air is 100 % of the maximum the air can hold.
- The relative humidity in image #4 is 50% because the amount of vapor in the air is 50% of the maximum the air can hold
- Relative humidity is the percentage of the measured vapor pressure divided by the theoretical saturation vapor pressure. Thus, relative humidity does not inform how much water is in the air, instead, the percentage of the theoretical maximum amount of water that is in the air at a particular temperature.
- When relative humidity is 100%, and water evaporates from the ground, water vapor will precipitate back into the liquid phase and fall as rain and there will be no net transfer of liquid water from the ground into the air.
- There are many inferences that can be made from the above concept.
    - On a hot day, more water will evaporate from the ground to produce a relative humidity of 70% compared to a cold day with 70% relative humidity. This is because on a hot day there is a greater saturation vapor pressure and more water in the air is needed to establish 70%.
    - When relative humidity is high, less evaporated water is required to reach the maximum saturated vapor density at a given temperature.
    - To minimize the net transfer of liquid water on the ground into water vapor in the air (evaporation), water should be delivered when it is cool and the relative humidity is high.

- Help the students to interpret the graph above and determine the conditions for each of the four regions. Use these conditions to construct a decision tree in their programs with a series of If-Then-End statements.
    - When the temperature is less than or equal to 25 and the relative humidity is greater than or equal to 80, it is cool and humid and an ideal time to water (green).
    - When the temperature is greater than 25 and relative humidity is greater than or equal to 80, it is hot and humid and an OK time to water (yellow).
    - When the temperature is less than or equal to 25 and relative humidity is less than 80 it is cool and dry and an OK time to water (yellow).
    - When the temperature is greater than 25 and relative humidity is less than 80, it is hot and dry and a bad time to water (red).
- Connect the sensor as indicated in the picture above.
- The DHT sensor requires a few seconds to "warm up" and begin communicating with the Hub. During this period, the temperature is reported to be absolute zero (-273 **°C**). The program uses a While loop to continuously read the DHT every two seconds until it reports a value greater than absolute zero.

Example program

```
PROGRAM: C3
Send("CONNECT DHT 1 TO IN 2 ")
ClrHome

Output(3,1,"DHT WARMING UP")
−273→T
While T<−270
Send("READ DHT 1 TEMPERATURE")
Get(T)
Wait 2
End
Output(3,1,"DHT IS NOW READY")

For(N,1,30)
Send("READ DHT 1 TEMPERATURE")
Get(T)
Send("READ DHT 1 HUMIDITY")
Get(H)
Output(4,1,"SAMPLE NUMBER= ")
Output(4,16,N)
Output(5,1,"TEMPERATURE (°C)= ")
Output(5,19,T)
Output(6,1,"HUMIDITY (PCT)= ")
Output(6,17,H)
If T≤25 and H≥80
Then
Output(8,1,"COOL AND HUMID")
Output(9,1,"BEST TIME TO WATER")
End
If T>25 and H≥80
Then
```

```
Output(8,1,"HOT AND HUMID ")
Output(9,1,"OK TIME TO WATER  ")
End
If T≤25 and H<80
Then
Output(8,1,"COOL AND DRY  ")
Output(9,1,"OK TIME TO WATER  ")
End
If T>25 and H<80
Then
Output(8,1,"HOT AND DRY   ")
Output(9,1,"BAD TIME TO WATER ")
End
Wait 2
End
```

Challenge 4: Write a program named C4 using a While loop to continuously measure and display a dashboard of all of sensor value readings. The user should be able to stop the monitoring by pressing the ESC key.

Teacher Guidance during challenge 4:

- This challenge could be skipped if there is insufficient time to complete the project. However, it is a major step toward the final project.
- The program uses most of the code from the previous three challenges.
- All of the sensors need to be connected as they were in the previous challenges.
- Use a while loop with the escape key as an exit.
- This program could be modified for the final challenge by adding the next challenge and control logic.

Example program:

```
PROGRAM: C4
Send("CONNECT LIGHTLEVEL 1 TO  IN1 ")
Send("RANGE LIGHTLEVEL 1 0 100")
Send("CONNECT DHT 1 TO  IN2 ")
Send("CONNECT MOISTURE 1 TO IN3")
Send("RANGE MOISTURE 1 0 100")
ClrHome

Output(3,1,"DHT WARMING UP")
```

```
−273→T
While T<−270
Send("READ DHT 1 TEMPERATURE")
Get(T)
Wait 2
End
Output(3,1,"DHT IS NOW READY")
Wait 1
Output(8,1,"PRESS CLEAR TO QUIT")

0→K
While K≠45
Send("READ LIGHTLEVEL 1")
Get(L)
Send("READ MOISTURE 1")
Get(M)
Send("READ DHT 1 TEMPERATURE")
Get(T)
Send("READ DHT 1 HUMIDITY")
Get(H)
Output(2,1,"LIGHT LEVEL= ")
Output(2,14,L)
Output(3,1,"MOISTURE LEVEL= ")
Output(3,17,M)
Output(4,1,"TEMPERATURE (°C)= ")
Output(4,19,T)
Output(5,1,"HUMIDITY (PCT)= ")
Output(5,17,H)
getKey→K
Wait 2
End
```
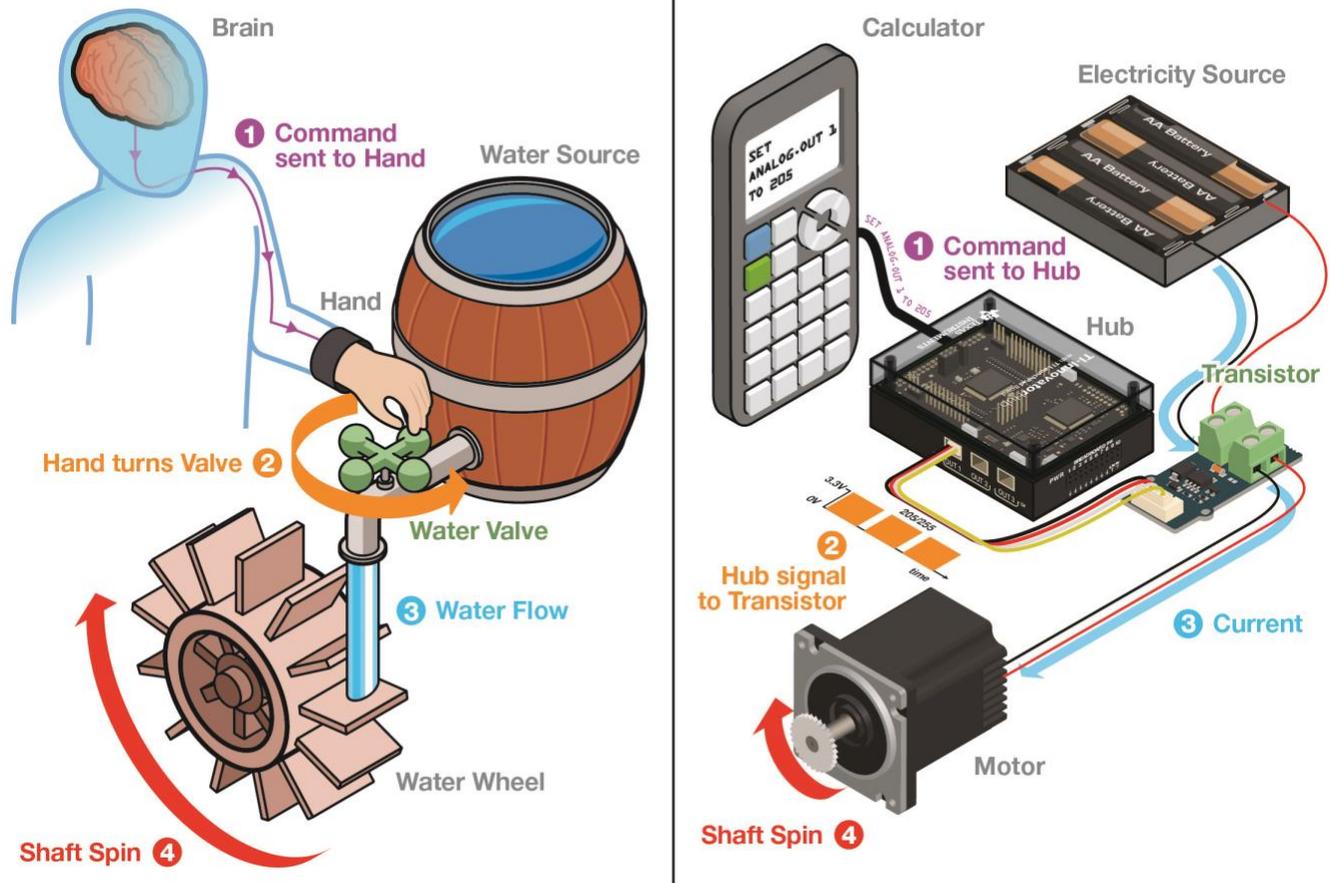
Challenge 5: Write a program named C5 to connect the pump power module and run the pump for 20 seconds. Be sure to turn the pump off. Try setting the pump power to different values. Try to estimate the pump flow rate in mL/sec.

Teacher Guidance during challenge 5:

- The Innovator Hub can control the amount of power delivered to the submersible pump using the power module with an attached battery pack. The power module has a special type of transistor called a MOSFET mounted on the unit. The illustration below makes an analogy between electricity flow in a circuit and water flow through a system to help students to better understand how a transistor works in an electrical circuit.

- When the TI-BASIC command "SET.ANALOG.OUT 1 TO 205" is sent to the Hub, the Hub creates a control signal on the OUT port. Like step 1 in the analogy the brain sending a command to the hand.
- When a power module is connected to the Hub's OUT port, that control signal is sent to the transistor. This signal causes the transistor to adjust the amount of electricity that will flow through it. In step 2 of our

water analogy, this signal can be thought of as the hand on turning the faucet.

- When the transistor is turned on, more current flows through it and into the motor. Like in step 3 of the analogy, when the faucet is opened, more water flows onto the water wheel.
- When more electrical current flows through the motor, greater power is delivered to the shaft causing it to turn faster. As in step 4 of the analogy, greater water flow onto the paddles of the water wheel cause it to turn faster.
- In this challenge, the power module is used to control the pump. A pump is just a motor with an impeller and housing attached to its shaft.
- Note: The power module only needs to be SET once. It will keep the present setting until a new command is sent to the module or until power is removed from the Hub. Be sure to SET the module off "SET ANALOG.OUT 1 TO 0" before exiting the program. If your program fails to do this, the pump will continue to run even after the program has quit.
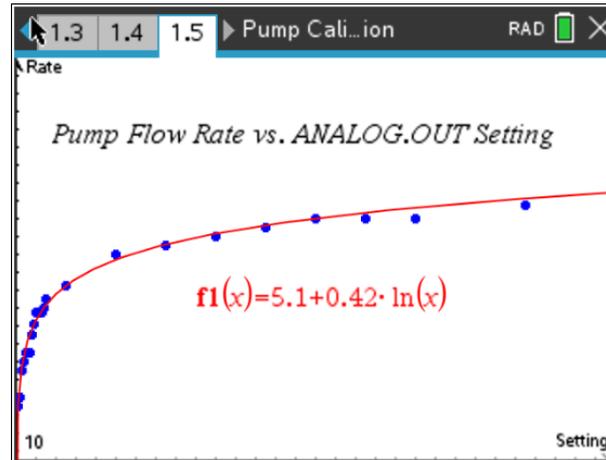
Example program:

```
PROGRAM: C5
Send("CONNECT ANALOG.OUT 1 TO OUT 1")
Send("SET ANALOG.OUT 1 255")
Wait 10
Send("SET ANALOG.OUT 1 0")
```

- Once students understand how to control the water pump, discuss how excessive watering may lead to water runoff, soil compaction, and nutrient leaching.
- Discuss with students the best flow rate of watering and if watering should be pulsed with a soak-in period.
- As an extension, the students could calculate the flow rate of the pump. To do this, run pump tubing into an empty graduated cylinder and fill the cylinder for a measured amount of time. Use the water volume and run time to calculate the flow rate. Challenge students to find how changing the ANALOG.OUT value changes the flow rate.

Example Extension Program:

```
PROGRAM:PUMPCAL
Send("CONNECT ANALOG.OUT 1 TO OUT 1")
For(N,0,10)
Input "PUMP SPEED? ",S
Send("SET ANALOG.OUT 1 eval(S) TIME 10")
Wait 10
```

```
End
Send("SET ANALOG.OUT 1 0")
```


Pump Flow Rate vs. ANALOG.OUT Setting

$$f1(x)=5.1+0.42\cdot \ln(x)$$

Final Challenge: Write a program that uses all of the sensors in the previous challenges. Continuously monitor light level, soil moisture, temperature, and humidity and display the current value on the display. Use your knowledge of ecology, biology, and Earth science to determine what the best conditions are to water your garden. When the condition is correct, set the pump to deliver water at a rate that is best for your garden.

Test your system with these conditions:

- light level < 20
- soil moisture < 10
- temperature > 25
- humidity > 80

Teacher Guidance during the Final Challenge:

- Discuss with students the concept of optimization and how that is expressed in the logic of the program.
- For each sensor, have students consider the conditions suited to watering. These values are called sensor set-points. For example, a set-point for temperature could be 25°C and watering should only occur when the measured temperature is less than the temperature set-point.

Sensor	Sensor Set-Point	Set-point condition	Conditional expression
Light Level	20	<	light<20
Air Temperature	25	<	Temperature<25
Air Humidity	80	>	humidity>80
Soil Moisture	10	<	moisture<10

- Once the four conditional expressions are determined in the table above, use the logical operations of AND and OR to link those expressions together to be evaluated by a single If-Then-End statement.
- When AND is used, both conditional expressions must be TRUE for the statement to be TRUE.
- For example, to construct a statement that requires the temperature to be less than the set-point of 28°C and the humidity is greater the set-point of 85% to water is:

```
If T<25 and H>80
```

When these conditions are true, set the pump on to 255. When the conditions are false, set the pump to zero.

```
Then
Send("SET ANALOG.OUT 1 255")
End
```

This statement ensures that the pump will turn on at maximum flow rate only when the temperature is cool and the humidity is high.

- When OR is used, either or both conditional expressions must be TRUE for the OR statement to be TRUE.
- For example to construct a statement that requires the light to be less than the set-point of 10 or the moisture to be less than the set-point of 15 to water.

```
If L<20 or M<10
Then
Send("SET ANALOG.OUT 1 255")
End
```

The pump will turn on when either it is dark or the soil is dry. Note, the pump will also turn on when it is dark and the moisture is low.

- It may be difficult for a student to consider 4 conditional expressions concurrently. It may be best, to begin with only one condition and then add an additional condition with a logical operation. Once successful with to conditional expression, a student can add additional conditions until all four are utilized.
- The pump control statement in the example below does the following: 1) runs the pump under any environmental conditions if the soil is very dry, soil moisture measurement less than 10, or 2) runs the pump when the environmental conditions are good for watering if the soil is moderately dry, soil moisture measurement less than 60. The pump will not run under any environmental conditions if the soil is wet. Note: Experiment by taking soil moisture measurements with your soil to determine whether 10 and 60 are good values for your garden environment.

```
If (L<20 and T<25 and H>80 and M<60) or M<10
Then
Output(6,1,"GOOD TIME TO WATER")
Else
Output(6,1,"BAD TIME TO WATER ")
End
```

Example program:

```
PROGRAM:SMARTH2O
Send("CONNECT LIGHTLEVEL 1 TO  IN1")
Send("RANGE LIGHTLEVEL 1 0 100")
Send("CONNECT MOISTURE 1 TO IN3")
Send("RANGE MOISTURE 1 0 100")
Send("CONNECT DHT 1 TO  IN2")
Send("CONNECT ANALOG.OUT 1 TO OUT1")
ClrHome

Output(3,1,"DHT WARMING UP")
−273→T
While T<−270
Send("READ DHT 1 TEMPERATURE")
Get(T)
Wait 2
End
Output(3,1,"DHT IS NOW READY")
Wait 1
Output(8,1,"PRESS CLEAR TO QUIT")

0→K
While K≠45
Send("READ LIGHTLEVEL 1")
Get(L)
Send("READ MOISTURE 1")
Get(M)
Send("READ DHT 1 TEMPERATURE")
Get(T)
Send("READ DHT 1 HUMIDITY")
Get(H)
Output(2,1,"LIGHT LEVEL= ")
```

```
Output(2,14,L)
Output(3,1,"MOISTURE LEVEL= ")
Output(3,17,M)
Output(4,1,"TEMPERATURE (°C)= ")
Output(4,19,T)
Output(5,1,"HUMIDITY (PCT)= ")
Output(5,17,H)
If (L<20 and T<25 and H>80 and M<60) or M<10
Then
Output(6,1,"GOOD TIME TO WATER")
Send("SET ANALOG.OUT 1 255")
Else
Output(6,1,"BAD TIME TO WATER ")
Send("SET ANALOG.OUT 1 0")
End
getKey→K
Wait 1
End
Send "SET ANALOG.OUT 1 0"
```