Drought in Africa inspires students to invent a smart irrigation system

Background resources for a project-based camp, club or in-class lesson based on experiences with a coding club at Sachse High School, Sachse, Texas.

Background material:

NBC5 Dallas-Fort Worth, Texas video report

Hunger

'It's a disaster': children bear brunt of southern Africa's devastating drought

Jonathan Manyowa walks on parched land in Chivi, Zimbabwe. His maize crops have failed three times. Drought has prompted Robert Mugabe to declare a state of disaster.
Photograph: Tsvangirayi Mukwazhi/AP

Footballer and student Prudence Zurumba with agriculture teacher and football coach Yeukai Musorowegomo.
Photograph: Henry Makiwa/World Vision UK

Football and famine in Zimbabwe

Chidyamakondo high school, near Masvingo in southern Zimbabwe, won the national girls' football championships three years in a row. But that cherished record – and far, far more – is now at risk.

"Students are fainting, struggling to concentrate in lessons, dropping out of school," says head teacher Morrison Musorowegomo. "We're having to shorten our assemblies and cut back on sport. Some of our players would rather leave the school and get married, or they will simply stay in the villages because they have no food."

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

Weather and agriculture in Zimbabwe

Weather and Climate Trends in Southern Africa

http://www.weathersa.co.za/

Conservation Agriculture in Zimbabwe

http://www.fao.org/in-action/conservation-agriculture-contributes-to-zimbabwe-economic-recovery/en/

Climate and Agriculture of Zimbabwe

http://www.our-africa.org/zimbabwe/climate-agriculture

Capturing Rainwater in Cisterns

https://www.thestandard.co.zw/2017/01/15/zimbabweans-capture-rainwater-future-use/

History of the Drought in Africa project

In August 2016, members of the Texas Instruments (TI) team that developed the TI-Innovator™ Hub with TI LaunchPad™ Board met with technology and curriculum coordinators of the Garland (Texas) Independent School District.

The district had adopted TI's TI-Nspire™ CX graphing technology in middle grades and high school, and was interested in using the compatible TI-Innovator™ Hub to increase students' exposure to science, technology, engineering and math (STEM), and coding.

The Garland team expressed a strong desire for experiences that truly integrate science and math concepts – along with engineering design and coding – to address a problem that students care about.

Fred Fotsch, STEM education specialist at Texas Instruments, had been looking into a project based on an article about a drought in Zimbabwe, and the impact that the drought was having on high school students there.

Along with GISD, Fred's project idea became the basis for an after-school STEM and coding club at Sachse High School.

The club ran for approximately 10 meetings in the fall of 2016. The students were very positive about their experiences. The educators involved were also very happy with the experience and expressed the desire to replicate the club in more schools in the district.

When shared at conferences, the project generated significant interest among educators. (Please refer to links on the title page of this slide deck for examples.)

The positive response from students and educators prompted the development of materials that would enable other schools to build their own STEM clubs, camps and classroom lessons around the project.

The resources that follow are based on the materials that we created for the club, and some additional materials that would have been helpful.

BUILD Student knowledge

Preparing for the project design challenge

When structuring the club session agendas, start with ultimate design challenges students will take on in the last few sessions.

Those challenges include designing a system that will:

» Monitor environmental conditions that are relevant to food production agriculture

» Control the flow of water from a collection cistern

» Incorporate an algorithm in a TI-BASIC program that optimizes the effective delivery of a limited amount of water based on the present environmental conditions

The sessions build knowledge and skills in several strands:

» Higher-level concepts in programming/computer science, including sensors and control

» Environmental science and biology that are the basis the design challenge

» Technology knowledge of TI-Basic, the TI-Innovator™ Hub with TI LaunchPad™ Board, and devices that are used in the project

Create a STEM experience

Resource materials

This slide deck provides materials for assembling a customized club, camp or classroom experience **based on the educators' and their students' objectives and interests.**

For example, projects can be designed to:

- Emphasize the programming and computer science aspects of the project, with only a small emphasis on the science concepts
- Focus almost entirely on the science concepts with coding limited to editing decision parameters in the final control program
- Expand the project into the social aspects of life in Zimbabwe

These materials provide a start for you to take the project where you decide to take it.

(Continued on next page)

Create a STEM experience

Resource materials

(cont.)

The slide deck is organized in two sections:

- **The overview, which includes this page**
- **Detailed agendas for implementation guidance**
  - **“Handout”** pages are references or guiding assignments we used
  - **“Background”** pages provide information about:
    - Programming concepts and syntax
    - Electronics concepts
    - Curriculum references
    - Science concepts related to the project

Smart **Irrigation** Project

Session Outline

Session 1

- » Acquaint students with the club, the project and the graphing calculator platform
- » Create and execute “Hello World” program

Session 2

- » Acquaint students with the TI-Innovator™ Hub with TI LaunchPad™ Board
- » Create/execute an LED "Blink" program to demonstrate outputs (and loops for advanced students)

Session 3

- » Continue programming the built-in features of the TI-Innovator™ Hub
- » Create a program that reads inputs from sensors and use If-Then-Else logic to control outputs

Sessions 4 and 5

- » Present students with the Science and Engineering Design Challenge, and then:
  - » Apply concepts from science to understand the system that needs to be controlled
  - » Draw or write a model of the system

(Continued on next page)

Smart Irrigation Project

Session Outline

(cont.)

Sessions 5 and 6

- Acquaint students with input sensors and output controllers for their project, and learn how to use them within a program

Session 7

- Ensure students have a graphing calculator, TI-Innovator™ with TI LaunchPad™ Board, sensors and a relay
- Challenge students to design their solutions using the Engineering Design Cycle
- Overall structure of the control program should be part of the design process

Sessions 8 and 9

- Students build their projects. Supplies should be made available.

Session 10

- Gallery-walk and presentations with peer review of solutions, certificates, etc.

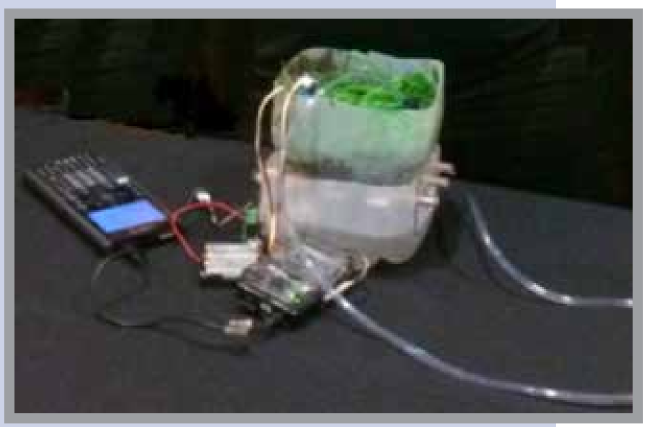
Smart **Irrigation** Project

Tips

- Have students work in teams of two or three, sharing a TI-Innovator™ Hub with TI LaunchPad™ Board, sensors and other materials. Three students per team enables progress in the event a student misses a session.
- Each student should have a graphing calculator so everyone can be actively engaged in programming.
- Have additional challenges in mind for the students who finish ahead of the others.
- During building, put two or more teams at large tables to spur discussion and idea sharing.
- Save a few minutes at the end of each session for students or teams to present their designs or programs to the group.
- Ensure students save their programs (and .tns TI-Nspire™ files) along the way. Also, if possible, back-up the students' programs to a computer or another calculator.
- Students appreciate a certificate of completion for the project and are proud to share their project at "Techfests" and other events.
- Be ready to extend the number of sessions and adapt the agenda based on progress, absences, etc.

Smart **Irrigation** Project

Parts List



- TI-Nspire™ CX family or TI-84 Plus CE Graphing Calculators
- TI-Innovator™ Hub
- External Battery for TI-Innovator™ Hub (TI accessory)
- Seeed Grove – Temperature & Humidity Sensor (mfg part#: 101020011)
- Seeed Grove – Moisture Sensor (mfg part#: 101020008)
- Seeed Grove – Light Sensor (mfg part#: 101020014)
- Seeed Grove – MOSFET (mfg part#: 103020008)
- Fafada® Water Pump Motor DC 3V Mini Submersible Water Pump for Fountain Aquarium 120L/H Max Lift 3.6FT (mfg part #:44020256)
- 2 male-to-male breadboard jumper cables
- 1/4” I.D. (inner dimension) x 3/8” O.D. (outer dimension) vinyl tubing
- Drinking straws
- Duct tape
- One-gallon milk jug
- Perlite or other medium for the plants to grow in

Session Agenda

Examples and Resources

Session 1 Agenda Example

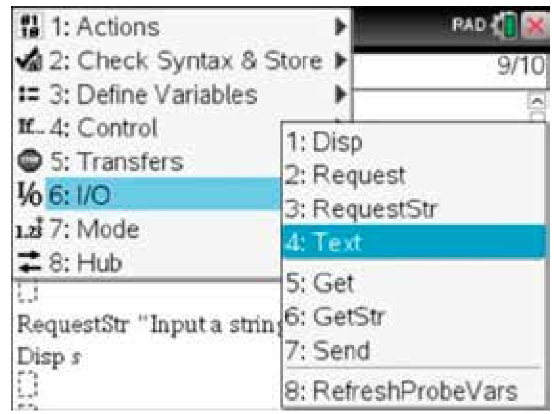
Acquaint students with the club, the drought project and the graphing calculator platform.

- **Introduce the project and the drought problem**
  - **Use project description handout as a starting point**
- **Create and execute "Hello World" program**
  - **Use the 10 Minutes of Code Unit 1, Skill Builder 1**
  - **https://education.ti.com/en/activities/ti-codes/nspire/10-minutes**
    - TI Technology Concepts
      - Inserting a program editor page
      - Creating, editing, and executing a program
      - Saving a TI-Nspire™ document file
      - TI-Nspire™ keyboard zones, menu key and menus, vars key for finding program names, Disp command syntax
    - Computing Concepts
      - Inputs, outputs, displaying strings (vs. values)

Session 1 Background

Input/Output commands for displaying and requesting text strings and numeric values

Program Editor Input/Output Menu




```
Disp "hello world"
Disp 2+3

Request "Input a number",n
Disp n

RequestStr "Input a string",s
Disp s
```

Display a string

Display a numeric value

Request a numeric value input

Prompt

Variable for the input numeric value

Request a string input

Prompt

Variable for the input text string

Example “Hello World” Program

```
Prgm

Disp "hello world”

EndPrgm
```

Some possible challenges/tasks

- » Make your own message
- » Display multiple messages
- » Display the result of a mathematical expression versus a string
  - » What is the difference between Disp 9*9*9 and Disp “9*9*9”?

Session 2 Agenda Example

Acquaint students with the TI-Innovator™ Hub with TI LaunchPad™ Board

» Do a few beginning activities.

» Create and execute an LED "Blink" program to demonstrate outputs (and loops for more advanced students).

Discuss drought problem situation

» Follow-up discussion based on interests of the students and the aspects that the teacher chooses to focus on.

Blink an LED

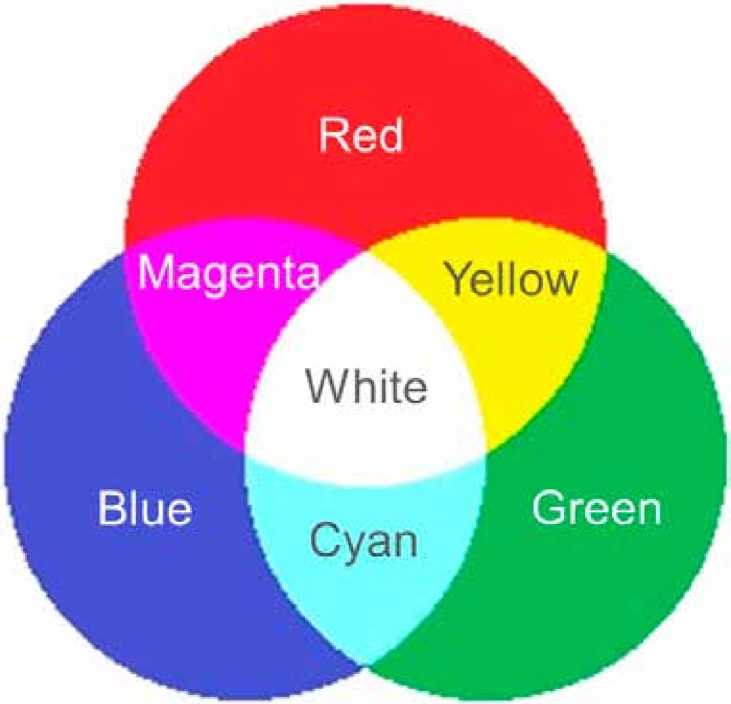
» Create and execute a program to turn on and off COLOR, BLUE or one of the other LED's built into the TI-Innovator™ Hub. If there is time, create a repeated blink pattern with a For loop.

(Continued on next page)

Session 2 Agenda Example

Creating and executing a program that blinks an LED engages students in:

- **TI Technology concepts**
  - TI-Innovator™ Hub overview
  - Connecting the TI-Innovator™ Hub to a calculator
  - TI-Innovator™ Hub menu for creating commands to send to the TI-Innovator™ Hub
- **Computing concepts**
  - Physical outputs
  - For loop
  - ON or OFF arguments for commands



COLOR (Red Green Blue) Output LEDs (Built into the TI-Innovator™ Hub)

The COLOR object built into the TI-Innovator™ includes three LEDs that can be set together or individually. The settings are from 0, off, to 255, full power.

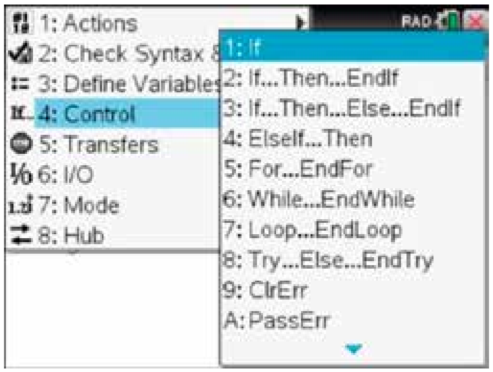
Note that there are 256 (2^8) states for the setting. This is referred to as an 8-bit color setting.

Source of background information:
http://www.cyberphysics.co.uk/topics/light/colorAddition.html

(Continued on next page)

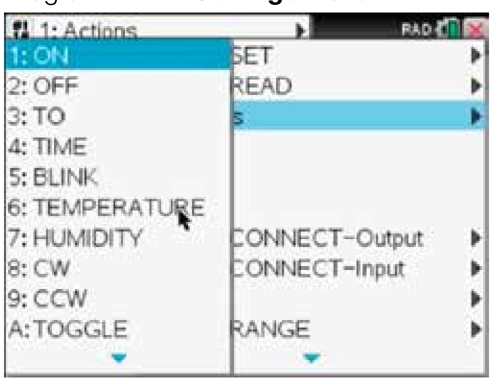
Session 2 Handout

Program **Control** menu

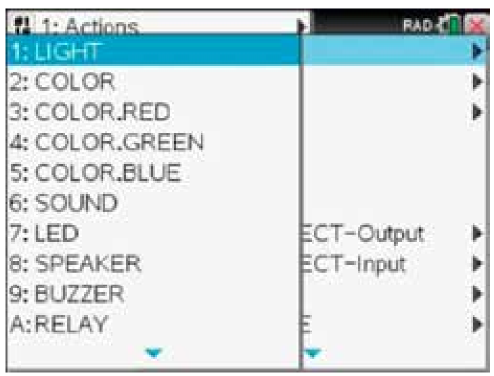


Program **Hub** menu

Program **Hub > Settings** menu



Program **Hub > Send "SET** menu



Command syntax:

Send “SET COLOR red green blue”

Code snippet to turn the BLUE light ON and OFF:

```
For c,1,10
  Send "SET COLOR.BLUE ON ”
  Wait 1
  Send "SET COLOR.BLUE OFF ”
  Wait 1
EndFor
```

Code snippet to CHANGE the COLOR of the light:

```
Send "SET COLOR 255 0 0”
Wait 1
Send "SET COLOR 255 0 255”
Wait 1
Send "SET COLOR 255 127 80”
Wait 1
Send "SET COLOR 0 0 0"
```

Code snippet to let the USER INPUT color values:

```
For a,1,3
  Request "Input Red value, 0-255",r
  Request "Input Green value, 0-255",g
  Request "Input Blue value, 0-255",b
  Send "SET COLOR eval(r) eval(g) eval(b)”
  Wait 2
  Send "SET COLOR 0 0 0”
EndFor
```

Session 2 Background

Example Blink Programs

Code snippet to turn one of the COLOR options ON and OFF.
You could also use **COLOR.RED** or **COLOR.GREEN**.

```
Send "SET COLOR.BLUE ON”
Wait 2
Send "SET COLOR.BLUE OFF”
Wait 2
```

Code snippet using a For loop to blink an LED 10 times.

```
For c,1,10
  Send "SET COLOR.BLUE ON”
  Wait 2
  Send "SET COLOR.BLUE OFF”
  Wait 2
EndFor
```

Resources for programming and the TI-Innovator™ Hub with TI LaunchPad™ Board:

10 Minutes of Code with TI-Innovator™ Hub introductory videos

https://www.youtube.com/watch?v=anbCjEUAkmQ

TI-Innovator™ Hub introductory video

https://education.ti.com/en/activities/stem/path-to-stem/nspire-cx#lightbox=ti-innovator-intro-videos

TI-Innovator™ System online Programming eGuides

https://education.ti.com/html/webhelp/EG_Innovator/EN/index.html

Session 3 Agenda Example

- Continue programming the TI-Innovator™ Hub with TI LaunchPad™ Board
- Create a program that reads inputs from an external sensor, and then use If-Then-Else logic to control outputs
- Read an external light level sensor value and make a decision to set an output on (COLOR or SOUND)

(Continued on next page)

Session 3 Agenda Example

To read an external light level sensor value and make an output decision:

» Create a program to connect an external light level sensor

» Set up a loop to repeatedly read the light level value, display the light level value

» Use If-Then(-Else) decision making to determine

» Set the value of a TI-Innovator™ Hub physical output (COLOR.BLUE 0-255 or SOUND)

» If there is time, replace For loop with a While loop

This activity engages students in:

» **TI Technology concepts:**

- » TI-Innovator™ Hub with TI LaunchPad™ Board input and output ports
- » CONNECT command to associate an external device with a port, numeric arguments for COLOR command

» **Computing concepts:**

- » Reading sensor values, variables, displaying values (vs. strings), If-Then-Else decision making, setting analog output levels (0-255, 2^8 states, 8-bit) using Pulse Width Modulation (PWM), While loop.

(Continued on next page)

Session 3 Agenda Example

Discussion suggestion:

Potential fit of sensor inputs and physical outputs to the problem

» **Science concepts:**

Measuring phenomena using instruments and sensors

» **Engineering concepts:**

Using sensors to monitor a system, feedback and control

Session 3 Handout

Light Level Sensor (External)

```
Send "CONNECT LIGHTLEVEL 1 TO IN 3 ”
Send “RANGE LIGHTLEVEL 1 0 100”

For c,1,50
  Send "READ LIGHTLEVEL 1”
  Get v
  Disp v
  Wait 1
EndFor
```

Measurements
Light level: relative brightness value in 0-16384 range (2^14, “14 digit resolution”)

Technical Notes:
The 0-16384 results can be mapped to 0-100 (if desired) using the RANGE command.
Set “RANGE LIGHTLEVEL 1 0 100”

Brightness Light Level Sensor (Built into the TI-Innovator™)

```
For c,1,50
Send "READ BRIGHTNESS”
Get b
Disp b
Wait 1
EndFor
```

Measurements
Measures light level: relative brightness value in 0-100 range

Technical Notes:
The 0-16384 (2^14, “14 digit resolution”) results are mapped by default to 0-100. The values can be set to the full set of values with the RANGE command.
Set “RANGE BRIGHTNESS 0 16384”

(Continued on next page)

Session 3 Handout

SOUND (Built into the TI-Innovator™ Hub)

The speaker built into the TI-Innovator™ Hub is controlled by the SOUND command. The SOUND command includes a frequency for the sound and a time value to play the frequency. If no time value is provided, the frequency will be played for 1 second.

Command syntax:

Send "SET SOUND frequency TIME seconds"

```
Send "SET SOUND 440"
Wait 1

Send "SET SOUND 261 TIME .5"
Wait .5
```

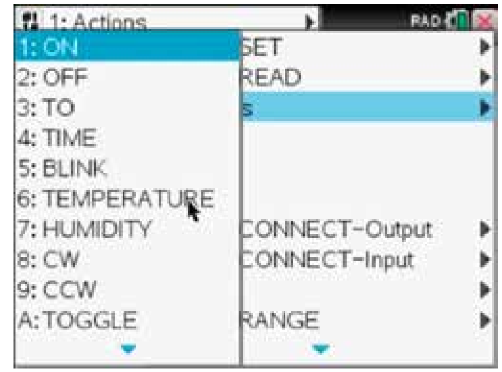
```
Request "Input sound frequency value",f
Request "Input time to play the frequency
",t
Send "SET SOUND eval(f) TIME eval(t)"
Wait t
```

Session 3 Handout

Program **Hub** menu

Program **Hub > Send "SET** menu

Program **Hub > Settings** menu



Session 3 Background

For ... EndFor loop

The For loop repeats a set of commands a specified number of times.

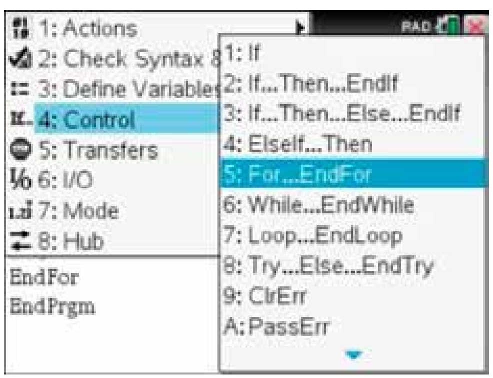
Command syntax:

For *counter* variable,*beginning* value,*end* value,[*step* value]...
EndFor

Notes:

» *counter* variable also known as *index* variable
» The *step* value is optional, the default step is 1
» EndFor is required to close the loop

Program **Control** menu



This loop will display 1,2,3, … 10.

```
For c,1,10
Disp c
EndFor
```

This loop will send a command to the TI-Innovator™ Hub with TI LaunchPad™ Board to read the built-in BRIGHTNESS sensor value.
It will get the value and store it to variable **b** and then it will print the value of variable **b**. The program waits one second after the display before proceeding to the end of the loop.
This loop repeats 50 times.

```
For c,1,50
Send "READ BRIGHTNESS”
Get b
Disp b
Wait 1
EndFor
```

(Continued on next page)

Session 3 Background

Decision Making Commands

If-Then command

```
If b>10 Then
© set of commands executed if statement is true
Disp "b is greater than 10”
Send "SET COLOR 0 255 0”
EndIf
```

If-Then-Else command

```
If b>10 Then
© set of commands executed if statement is true
Disp "b greater than 10”
Send "SET COLOR 255 0 0”
Else
© set of commands executed if statement is false
Disp "b less than or equal to 10”
Send "SET COLOR 0 0 255”
EndIf
```

Logic functions

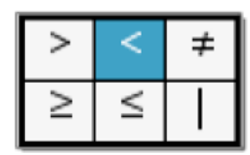
"and"

```
If b>10 and c>100 Then
© set of commands executed if both statements are
true
Disp "b is greater than 10 and c greater than 100”
Send "SET COLOR 0 255 0”
EndIf
```

"or"

```
If b>10 or c>100 Then
© set of commands executed if either of the
statements are true
Disp "b is greater than 10 or c greater than 100”
Send "SET COLOR 0 255 0”
EndIf
```

Note: Relational operators can be found on the keyboard using [=] and ctrl [=]. Logic functions, and, or, not, etc. can be typed or found in the catalog.



(Continued on next page)

While…EndWhile loop

The While loop repeats a set of commands as long as the While condition is true.

Command syntax:
While *condition*
Commands in While block
EndWhile

This While loop is set to continuously ask the TI-Innovator™ Hub with TI LaunchPad™ Board to read the BRIGHTNESS sensor value, get the value to variable ‘b’, display the value of ‘b’ and then make an If-Then-Else decision to turn on or off a green LED based on the value of ‘b’. The loop can be stopped by a ‘b’ value of zero, which can be achieved by completely covering the sensor. The While loop is able to run the first time because an initial value greater than zero is stored to ‘b’. When the loop stops a message is displayed to inform the user that the loop is complete.

```
b:=1
While b>0
Send "READ BRIGHTNESS ”
Get b
Disp b
If b>10 Then
Send "SET COLOR.GREEN ON”
Else
Send "SET COLOR.GREEN OFF”
EndIf
EndWhile
Disp "While loop is complete"
```

(Continued on next page)

Session 3 Background

» To **stop a program** that is running a long or infinite loop, press and hold the HOME/ON key until you receive a dialogue box.

» To **add a comment** line to a program, select Comment from the Program Edit Actions menu.

```
Prgm
© This is a comment
EndPrgm
```

» To **enclose a block of commands in a loop**, do the following:

- » **Step 1:** Move your cursor to the line below the block
- » **Step 2:** Press and hold SHIFT, then press UP repeatedly to highlight the block
- » **Step 3:** Press MENU, Control then select For…EndFor or another loop from the Program Editor menu

(Continued on next page)

Storing to numeric and string variables

You can store to numeric and string variables using one of these three approaches:

Using Sto →

```
2→a
Disp a
"bbb"→b
Disp b
```

Using :=

```
c:=4
Disp c
d:="ddd”
Disp d
```

Using Define

```
Define e=3
Disp e
Define f=“fff”
Disp f
```

(Continued on next page)

Storing TI-Innovator™ Hub with TI LaunchPad™ Board sensor readings to a list

By using the For loop counter variable, you can store successive sensor readings into a list for later use.

This example stores BRIGHTNESS values to a list called brightlist. [n] accesses the nth element of the list.

```
For n,1,5
  Send "READ BRIGHTNESS ”
  Get b
  brightlist[n]:=b
  Wait 1
EndFor
```

This more advanced example estimates the time of the reading by using a variable w, as the time input for the Wait command.

```
w:=0.5
For n,1,5
 Send "READ BRIGHTNESS ”
 Get b
 brightlist[n]:=b
 timelist[n]:=(n-1)*w
 Wait w
EndFor
```

(Continued on next page)

Example program

```
Send "CONNECT LIGHTLEVEL 1 TO IN 1 ”
Send "RANGE LIGHTLEVEL 1 0 100”

For c,1,20
Send "READ LIGHTLEVEL 1”
Get v
Disp v
If v<20 Then
 Send "SET COLOR 255 0 0”
 Wait 1
Else
 Send "SET COLOR 0 255 0”
 Wait 1
EndIf
EndFor
```

The **CONNECT** command is needed to associate the port with a type of device. Each device type has defined properties that the TI-Innovator™ Hub uses to interpret and communicate electronic signals.

The **RANGE** command maps the light level reading to a 0 to 100 scale.

Some possible challenges/tasks:

» Have the students choose their own colors which are a mix of R,G and B

» Replace or add sound output

» Change the program to map the light level value to the color brightness value (0-255).

» Change the program to map the light level value to a sound frequency.

» Change the program to use a While loop instead of a For loop

(Continued on next page)

Session 3 Background

Pulse Width Modulation (PWM)

PWM is a term that you may hear in regard to electronically controlled outputs. Here is some background on the concept that will be of help if you choose to introduce the concept to your students.

Pulse Width Modulation (PWM) is a technique used to control the power delivered to a device. When an LED is connected to an output on the Hub and it is turned on, it produces a bright light. When it is off, it produces no light. If the LED is turned on and off very rapidly, then it will produce a light brightness somewhere between full on and off. If the LED is turned on and then off very rapidly, the eye does not notice the flicker of the light. However, if it is quickly turned on for as long as it is turned off, this is called 50% duty cycle, then the LED will appear to the eye halfway between full on and off, that is it will appear dimmer. If a motor is connected to a PWM output on the Hub, then the speed of the motor will vary with the PWM duty-cycle. In other words when the output is switched on for 50% of the time and then off for 50% of the time, very rapidly, the effect is the motor will run at half of its full speed. The red LED (LIGHT) on the Hub is connected to an analog output. The range of values for setting PWM are 0-255. This number range is due to the fact that a single binary byte is used as the argument to the PWM (ANALOG.OUT) function. A byte is 8 binary digits, for example 10101101, and an 8 digit binary number has 2^8 or 256 values, thus, 0-255.

Session 3 Handout

Input-Output Control System Program Design

Setup

- What inputs and outputs need to be connected?
- Which ports are they connected to?

Loop to monitor and decide

- What loop will you use to repeatedly check the sensors and decide when to set output values? (For, While,…)

Checking the sensor values

- What sensors need to be read?
- What variables will you “Get” the sensor values to?

Deciding to set new output values

- What If-Then or If-Then-Else statements are needed?
- What sensor values should be used as decision points in the If statements.

Output (starting, output level(s), stopping)

- What output level(s) and patterns do you want to use?
- What sensor values or amount of time should cause the output to stop?

Sessions 4 and 5 Agenda Example

Students are presented with the Science and Engineering Design Challenge and then apply concepts from science to understand the system that needs to be controlled. A model of the system is either drawn or written by the student.

» Discuss Engineering Design Challenge

» Discuss Engineering Design Process: Design, Build, Test, Optimize, Repeat

» Discuss Abiotic and Biotic Factors affecting the ecosystem and the design challenge

» Define factors and interactions in the system

» Sketch the system

(Continued on next page)

Sessions 4 and 5 Agenda Example

Engineering Design Process

- » Define problem
- » Design a proposed solution
- » Present group’s design to class
- » Listen to feedback and make appropriate changes
- » Build and test the design
- » Use test results to make appropriate and informed changes to design
- » Present your solution to class

(Continued on next page)

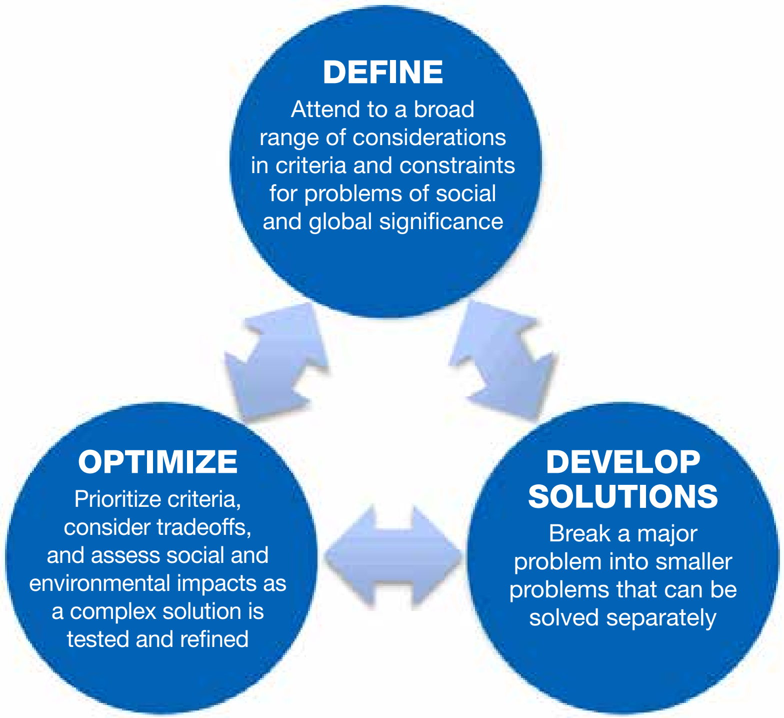
Sessions 4 and 5 Agenda Example

Engineering Design Process

» **Defining** and delimiting engineering problems involves stating the problem to be solved as clearly as possible in terms of criteria for success, and constraints or limits.

» **Designing** solutions to engineering problems begins with generating a number of different possible solutions, then evaluating potential solutions to see which ones best meet the criteria and constraints of the problem.

» **Optimizing** the design solution involves a process in which solutions are systematically tested and refined, and the final design is improved by trading off less important features for those that are more important.



Source: Next Generation Science Standards
http://www.nextgenscience.org/sites/default/files/Appendix%20I%20-%20Engineering%20Design%20in%20NGSS%20-%20FINAL_V2.pdf

It is important to point out that these component ideas do not always follow in order, any more than do the “steps” of scientific inquiry. At any stage, a problem-solver can redefine the problem or generate new solutions to replace an idea that just isn’t working out.

(Continued on next page)

Design: Factors Affecting a Home Garden Ecosystem

Abiotic Factors

- Sunlight
- Soil grain size and compaction
- Soil moisture
- Heat
- Temperature
- Humidity
- Soil nutrients
- Precipitation
- Accumulation
- Runoff
- Nutrient leeching
- Evaporation

Biotic Factors

- Photosynthesis
- Transpiration
- Nutrient uptake
- Water uptake

Interactions

- Sunlight and primary production
- Evaporation/transpiration and humidity
- Evaporation/transpiration and temperature
- Precipitation and runoff

See NGSS Middle School Life Sciences LS2: Interactions, Energy, and Dynamic Relationships in Ecosystems
http://www.nextgenscience.org/sites/default/files/MS%20LS%20DCI%20combinedf.pdf

(Continued on next page)

Sessions 4 and 5 Agenda Example

Interdependent Relationships in Ecosystems

Sketch an ecosystem diagram of the flow of water through a field of corn in Zimbabwe. Include the following:

1. Precipitation
2. Transpiration
3. Evaporation
4. Photosynthesis
5. Producer/Consumer
6. Erosion
7. Flooding

Sessions 4 and 5 Background

Science Concepts

Identify biotic and abiotic factors in a home garden.

- **Light:** the radiant output of the sun affects primary productivity due to photosynthesis
- **Soil moisture:** the available water in the soil for transport into the roots
- **Nutrients:** the available nitrogen, phosphorus and potassium in the soil
- **Soil compactness**: a measure of the porosity of the soil and determined by grain phi-size
- **Competition:** the struggle among organisms for limited resources
- **Caloric production:** the net amount of energy stored in macromolecules from photosynthesis

(Continued on next page)

Sessions 4 and 5 Background

Science Concepts

Ecosystems

- **Biotic and abiotic factor interactions:** the physical and biological interactions among living and non-living factors within the ecosystem
- **Energy flow:** the trace of energy from primary production through all trophic levels, ultimately ending in the apex predator
- **Water flow-through system:** the trace of water through the hydrologic cycle within the system
- **Nutrient flow-through system:** the trace and conservation of the essential metabolites through the system
- **Predator-prey interactions:** tracing the completion between trophic levels with the system
- **Productivity:** the amount of primary production that is moved into higher trophic levels of an ecosystem
- **Fecundity:** the net birthrate of a population within an ecosystem
- **Morbidity:** the net death rate of a population within an ecosystem
- **Carrying capacity:** the theoretical limit of a given population size within an ecosystem

(Continued on next page)

Sessions 4 and 5 Background

Science Concepts

Apply the concepts of plant physiology to a home garden management system.

- **Photosynthesis:** the chlorophyll catalyzed chemical conversion of carbon dioxide, water and light energy into high energy macromolecules
- **Transpiration:** the energy-driven movement of water from the soil to the atmosphere via capillary action within the vascular system of a woody plant
- **Gas exchange:** the absorption and release of metabolic gasses in plants
- **Turgor pressure:** the intracellular hydrostatic pressure within a plant that gives the plant its rigidity
- **Nutrient uptake:** the compound specific active transport of metabolites into the cells of plants
- **Water uptake:** the passive and active absorption of water into a plant

(Continued on next page)

Sessions 4 and 5 Background

Science Concepts

Apply the concepts of the water cycle to a home garden management system.

- **Precipitation:** the release of condensed water vapor from a saturated atmosphere
- **Evaporation:** the loss of soil moisture due to water vapor diffusion into the atmosphere
- **Run-off:** water that does not percolate into soil after precipitation and is affected by the rate of precipitation and soil compactness
- **Percolation:** surface water moving deeper into soil via hydrostatic pressure
- **Leeching:** loss of water soluble nutrients due to run-off and percolation
- **Soil moisture capacity:** the amount of water that a soil type can hold

(Continued on next page)

Define factors and interactions

Abiotic Factors

1. Sunlight
2. Soil grain size compaction
3. Soil moisture
4. Heat
5. Temperature
6. Humidity
7. Soil nutrients
8. Precipitation
9. Accumulation
10. Runoff
11. Nutrient leeching
12. Evaporation

(Continued on next page)

Define factors and interactions

Biotic Factors

1. Photosynthesis
2. Transpiration
3. Nutrient uptake
4. Water uptake

Interactions

1. Sunlight and primary production
2. Evaporation/transpiration and humidity
3. Evaporation/transpiration and temperature
4. Precipitation and runoff

(Continued on next page)

Sketch factors and interactions

For more information: UN Food and Agriculture Organization
http://www.fao.org/docrep/t0231e/t0231e05.htm

Sessions 4 and 5 Optional Exploration

NGSS Practice: Engaging in argument from evidence

Use evidence from the graph of annual rainfall in southern Africa **to construct an explanation** of why some people in Zimbabwe do not have enough to eat. Hypothesize the effect of rainfall on food supply in Zimbabwe.

NGSS Crosscutting Concept:

» Small changes in one part of a system might cause large changes in another part.

Sessions 5 and 6 Agenda Example

Students are acquainted with the input sensors and output controllers available for their project, and learn how to use them within a program.

Input Sensors

» Digital Temperature and Humidity (DHT)

» Light Level

» Soil Moisture

Output Control

» MOSFET Transistor to control pump

TI Technology concepts:

» TI-Innovator™ Hub with TI LaunchPad™ Board input and output ports

» CONNECT command to associate an external device with a port

» Numeric arguments for COLOR command

Computing concepts:

» Reading sensor values, variables, displaying values (vs. strings),

» If-Then-Else decision making

» Setting analog output levels (0-255, 2^8 states, 8-bit) using Pulse Width Modulation (PWM)

» While loop

Input-Output Control System Program Design

Setup

» What inputs and outputs need to be connected?

» Which ports are they connected to?

Loop to monitor and decide

» What loop will you use to repeatedly check the sensors and decide when to set output values? (For, While,…)

Checking the sensor values

» What sensors need to be read?

» What variables will you “Get” the sensor values to?

Deciding to set new output values

» What If-Then or If-Then-Else statements are needed?

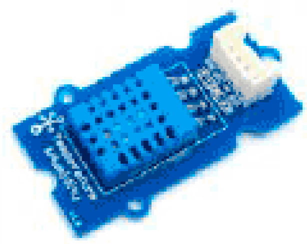
» What sensor values should start the pump?

Output (starting, output level(s), stopping)

» What output level(s) and patterns do you want to use?

» What sensor values or amount of time should cause the output to stop?

Sessions 5 and 6 Handout



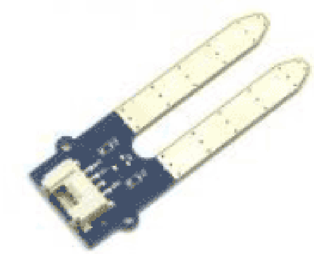
Digital Temperature and Humidity Sensor

```
Send "CONNECT DHT 1 TO IN 1 ”
Send "READ DHT 1 TEMPERATURE ”
Get t
Send "READ DHT 1 HUMIDITY ”
Get h
```

```
Send "CONNECT DHT 1 TO IN 1 ”
Send "READ DHT 1”
Get htlist
t:=htlist[1]
h:=htlist[2]
```

Measurements
DHT Temperature: temperature in degrees Celsius
DHT Humidity: relative humidity in % (0-100)

Technical Notes:
Blue DHT
Sampling Rate: 1/2 Hz (1 sample per 2 seconds)
Warmup time: ~ 4 seconds (reads -273 C until warmed up)
White DHT
Sampling Rate: 1 Hz (1 sample per 1 second)
Warmup time: ~ 2 seconds (reads -273 C until warmed up)



Soil Moisture Sensor

```
Send "CONNECT MOISTURE 1 TO IN 2 ”
Send ”RANGE MOISTURE 1 0 100”
Send "READ MOISTURE 1”
Get m
Disp m
```

Measurements
Soil Moisture: relative moisture value in 0-16384 range (2^14, “14 digit resolution”)

Technical Notes:
The 0-16384 results can be mapped to 0-100 (if desired) using the RANGE command
Set “RANGE MOISTURE 1 0 100”

Light Level Sensor (External)

```
Send "CONNECT LIGHTLEVEL 1 TO IN 3 ”
Send “RANGE LIGHTLEVEL 1 0 100”

For c,1,50
  Send "READ LIGHTLEVEL 1”
  Get v
  Disp v
  Wait 1
EndFor
```

Measurements
Light level: relative brightness value in 0-16384 range (2^14, “14 digit resolution”)

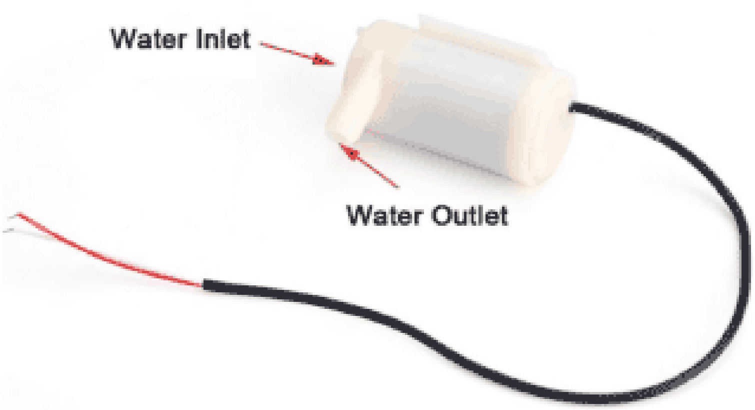
Technical Notes:
The 0-16384 results can be mapped to 0-100 (if desired) using the RANGE command.
Set “RANGE LIGHTLEVEL 1 0 100”

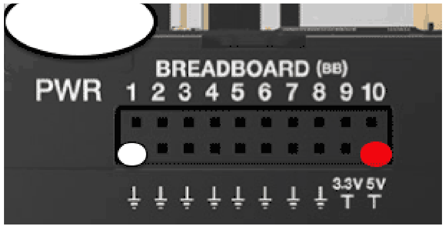
Sessions 5 and 6 Handout

Pump Control Sub-System – Rechargeable Battery

The system uses a small submersible pump. The pump flow (volume per unit time) is controlled by commands sent from the TI-Basic program through the TI-Innovator™ Hub to a MOSFET (Field Effect Transistor).

An external battery source is required to provide enough power to the pump. In the version of the project below we use a rechargeable battery connected to the PWR port of the TI-Innovator™ Hub to provide the power. The electrical current is routed from the TI-Innovator™ Hub breadboard 5V and ground pins to the MOSFET and then on to the pump.

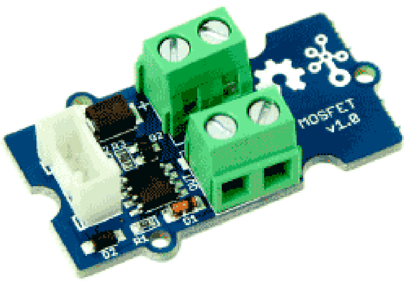




Submersible 3V DC Motor Pump
Note: If connection to MOSFET is reversed then pump flow is reversed.

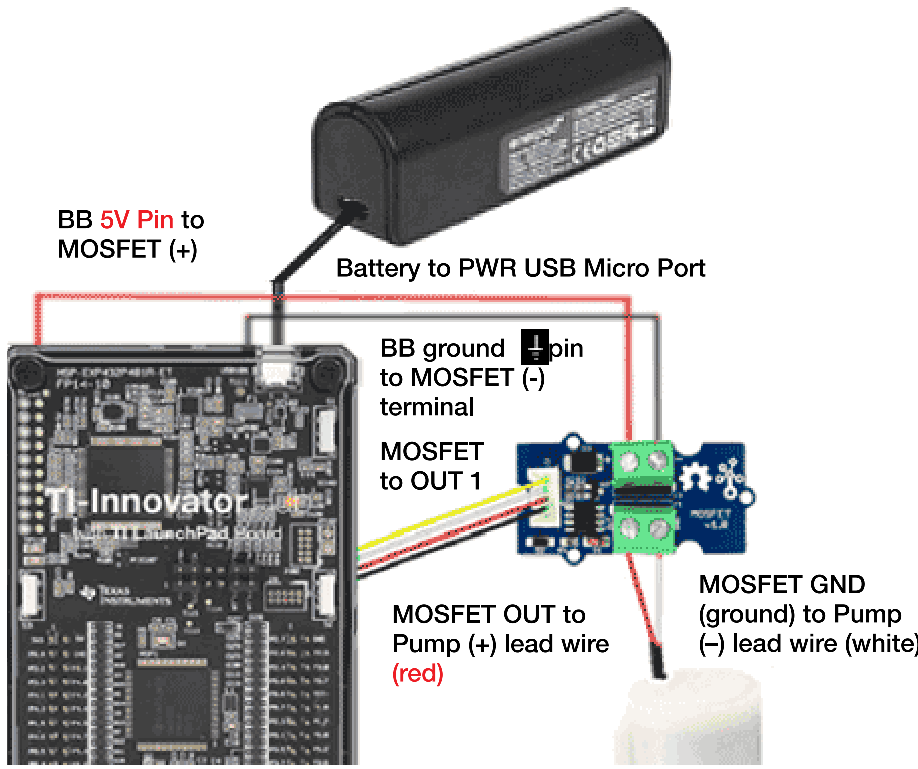
(Continued on next page)

Pump Control Sub-System – Rechargeable Battery *(cont.)*



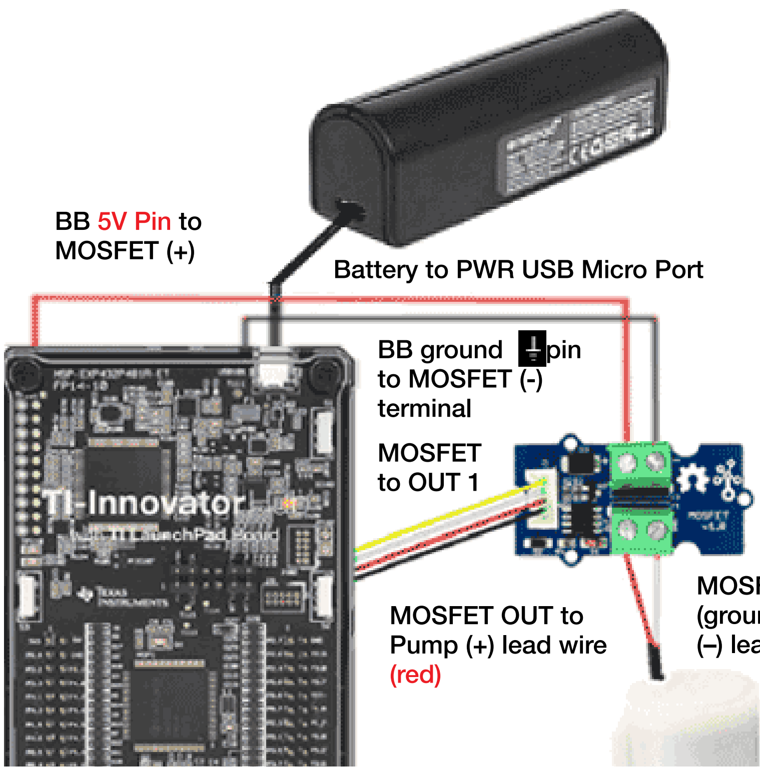
MOSFET (Field Effect Transistor)
Controls the current level to the pump

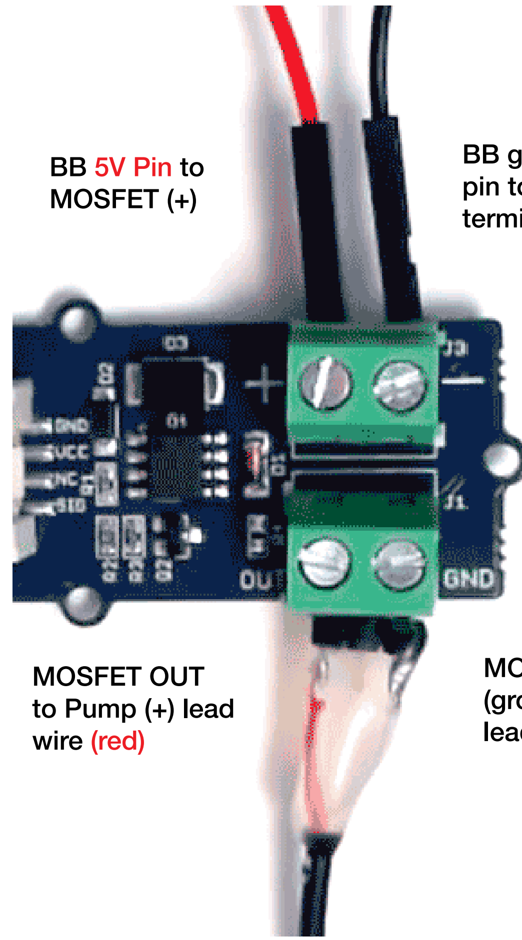
Control
MOSFET/ANALOG.OUT: control levels, 0-255 (2^8, 8-bit control)




```
Send "CONNECT ANALOG.OUT 1 TO OUT 1 ”
255→p
10→n
Send "SET ANALOG.OUT 1 eval(p)”
Wait n
Send "SET ANALOG.OUT 1 0"
```

Connecting rechargeable battery and MOSFET to pump

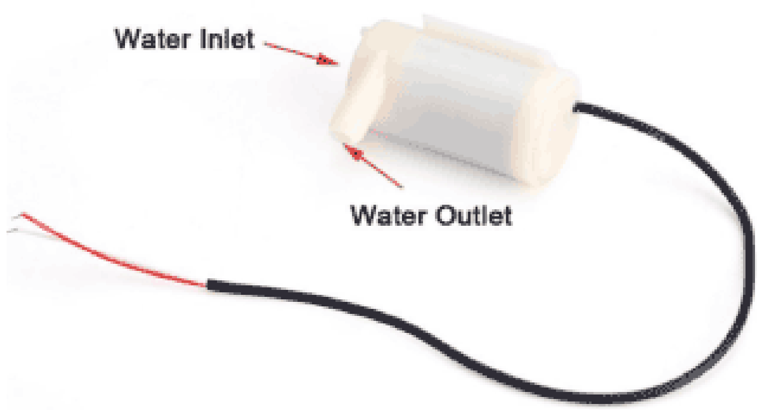




Pump Control Sub-System – AA Battery Pack

The system uses a small submersible pump. The pump flow (volume per unit time) is controlled by commands sent from the TI-Basic program through the TI-Innovator™ Hub to a MOSFET (Field Effect Transistor).

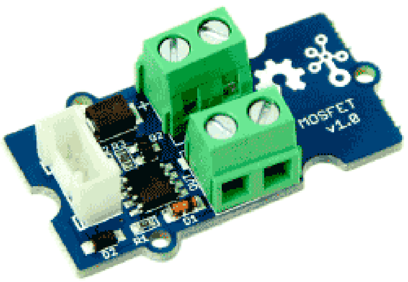
An external battery source is required to provide enough power to the pump. In the version of the project below we use a 4xAA battery pack and a MOSFET connected to the TI-Innovator™ Hub to control the pump.



Submersible 3V DC Motor Pump
Note: If connection to MOSFET is reversed then pump flow is reversed.

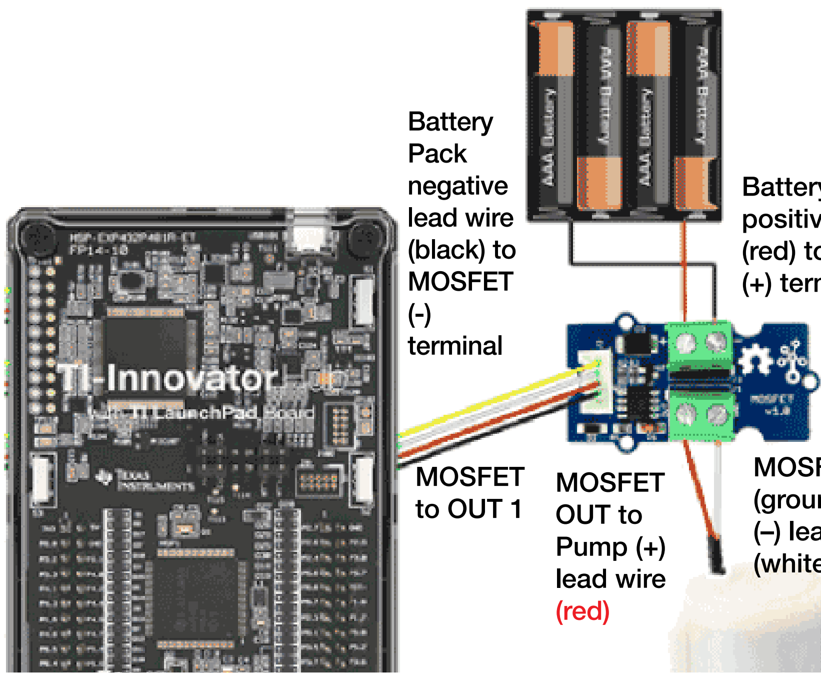
(Continued on next page)

Pump Control Sub-System – AA Battery Pack *(cont.)*



MOSFET (Field Effect Transistor)
Controls the current level to the pump

Control
MOSFET/ANALOG.OUT: control levels, 0-255 (2^8, 8-bit control)




```
Send "CONNECT ANALOG.OUT 1 TO OUT 1 ”
255→p
10→n
Send "SET ANALOG.OUT 1 eval(p)”
Wait n
Send "SET ANALOG.OUT 1 0"
```

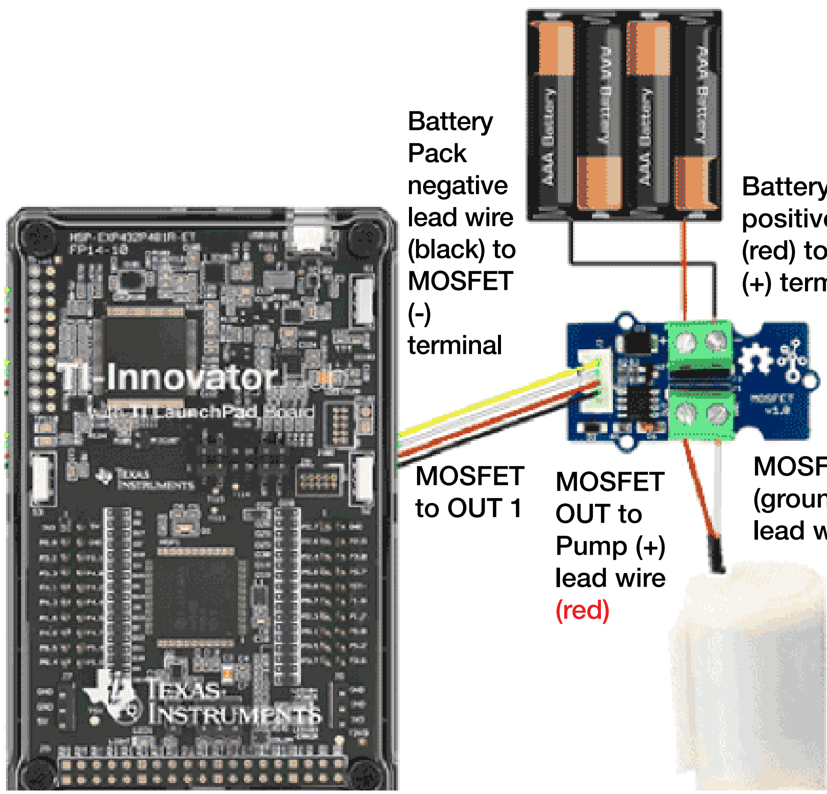
Sessions 5 and 6 Background

Connecting AA battery pack and MOSFET to pump

AA Battery Pack

Battery Pack negative lead wire (black) to MOSFET (-) terminal

Battery Pack positive lead wire (red) to MOSFET (+) terminal



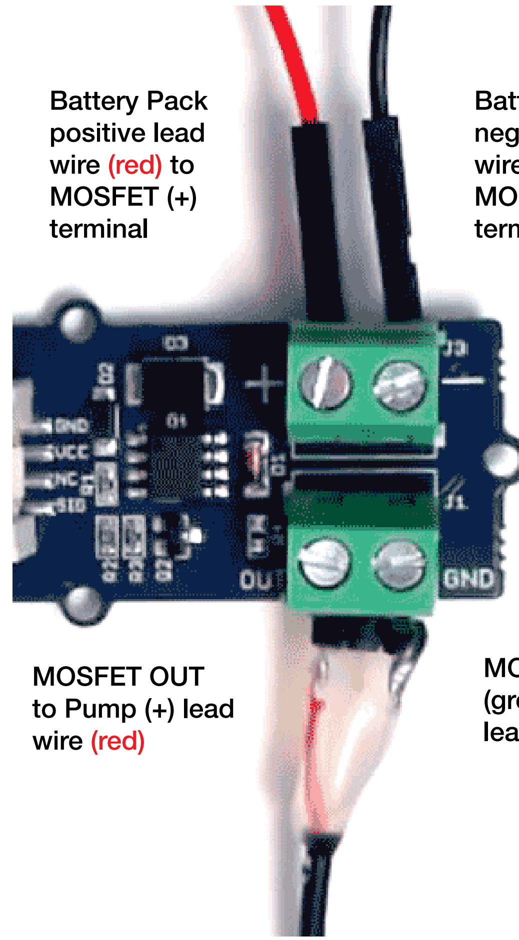
MOSFET to OUT 1

MOSFET OUT to Pump (+) lead wire (red)

MOSFET GND (ground) to Pump (–) lead wire (white)

Battery Pack positive lead wire (red) to MOSFET (+) terminal

Battery Pack negative lead wire (black) to MOSFET (-) terminal



MOSFET OUT to Pump (+) lead wire (red)

MOSFET GND (ground) to Pump (–) lead wire (white)

To pump

Example program

```
Prgm
Send "CONNECT LIGHTLEVEL 1 TO IN 1”
Send "CONNECT DHT 1 TO IN 2”
Send "CONNECT MOISTURE  1 TO IN 3”
Send "RANGE LIGHTLEVEL 1 0 100”
Send "RANGE MOISTURE 1 0 100”

Disp "Warming Up Sensors!”
temperature:=-273
While temperature<-270
Send "READ DHT 1 TEMPERATURE”
Get temperature
Wait 1
EndWhile
```

```
For n,1,50
  Send "READ LIGHTLEVEL 1”
  Get light
  Send "READ  MOISTURE 1”
  Get moisture
  Send "READ  DHT 1 TEMPERATURE”
  Get temperature
  Send "READ  DHT 1 HUMIDITY”
  Get humidity
  Disp "T=",temperature,"H=",humidity,"L=",light,"M=",moisture
  Wait 1
  If light<50 and moisture<5 Then
   Disp ”Good condition to pump”
  Else
   Disp “Not a good condition to pump”
  EndIf
EndFor

EndPrgm
```

(Continued on next page)

Example program *(cont.)*

Some possible challenges/tasks

- Have the students create the code to determine if the DHT sensor is warmed up or not.
- Have the students try simple If-Then-Else statements for determining the best conditions to pump.
- Have the students create decision statements that incorporate multiple sensors using *and* and *or*.
- Incorporate light level, moisture, temperature and humidity into the decision to set the pump or not.
- Discuss and try different threshold values for the best conditions to pump or not.
- Can you restructure the program to use a While loop with a way for you to stop the program instead of a For loop?

Sessions 5 and 6 Background

Example program

```
Prgm

Send "CONNECT ANALOG.OUT 1 TO OUT 1"
© ANALOG.OUT 1 is the object that sets
the power level of the MOSFET and the
speed of the pump.

For n,1,5
 Send "SET ANALOG.OUT 1 eval(45*n)"
 Wait 3
 Send "SET ANALOG.OUT 1 0"
 Wait 3
EndFor

EndPrgm
```

The loop pulses the pump in 3 second increments at increasing power levels from the MOSFET starting at level 45 and growing to 225 in steps of 45. The pump stays on during a 3 second waiting period followed by a command to turn the pump off (power level 0).

Some possible challenges/tasks

» Try the pump at different levels (0-255) (and observe interaction with the soil)

» Try different patterns for pumping the water (trickle, pulse,…)

Session 7 Agenda Example

Given the Science Design Challenge, a graphing calculator, a TI-Innovator™ Hub, sensors and a relay, students will design a solution to the posed challenge using the Engineering Design Cycle. Overall structure of the control program should be part of the design process.

» Discuss the design challenge

» Students sketch the physical design of the system

» Students outline the sections and flow of their control program

» Students share their designs with the group

Design Challenge

Design a system that will:

- Monitor environmental conditions that are relevant to food production agriculture
- Control the flow of water from a collection cistern
- Incorporate an algorithm in a TI-Basic program that optimizes the effective delivery of a limited amount of water based on the present environmental conditions

Science and Engineering Design Challenge

Sketch a control system to monitor and meter water from a rain collection cistern to irrigate a garden in Zimbabwe.

Include the following:

» Sensors (inputs)

» Actuators (outputs)

» Control logic (decisions)

Science and Engineering Design Challenge

Sketch a control system to monitor factors and deliver water to plants.

For additional information: UN Food and Agriculture Organization
http://www.fao.org/docrep/t0231e/t0231e05.htm

Smart Irrigation Program Design

Setup

- What inputs and outputs need to be connected?
- Which ports are they connected to?

Loop to monitor and decide

- What loop will you use to repeatedly check the sensors and decide when to pump? (For, When,…)

Checking the sensor values

- What sensors need to be read?
- What variables will you "Get" the sensor values to?

Deciding to set new output values

- What If-Then or If-Then-Else statements are needed?
- What sensor values should start the pump?

Pumping (starting, output level(s), stopping)

- What output level(s) and patterns do you want to use for pumping?
- What sensor values should cause the pump to stop?

Sessions 8 and 9 Agenda Example

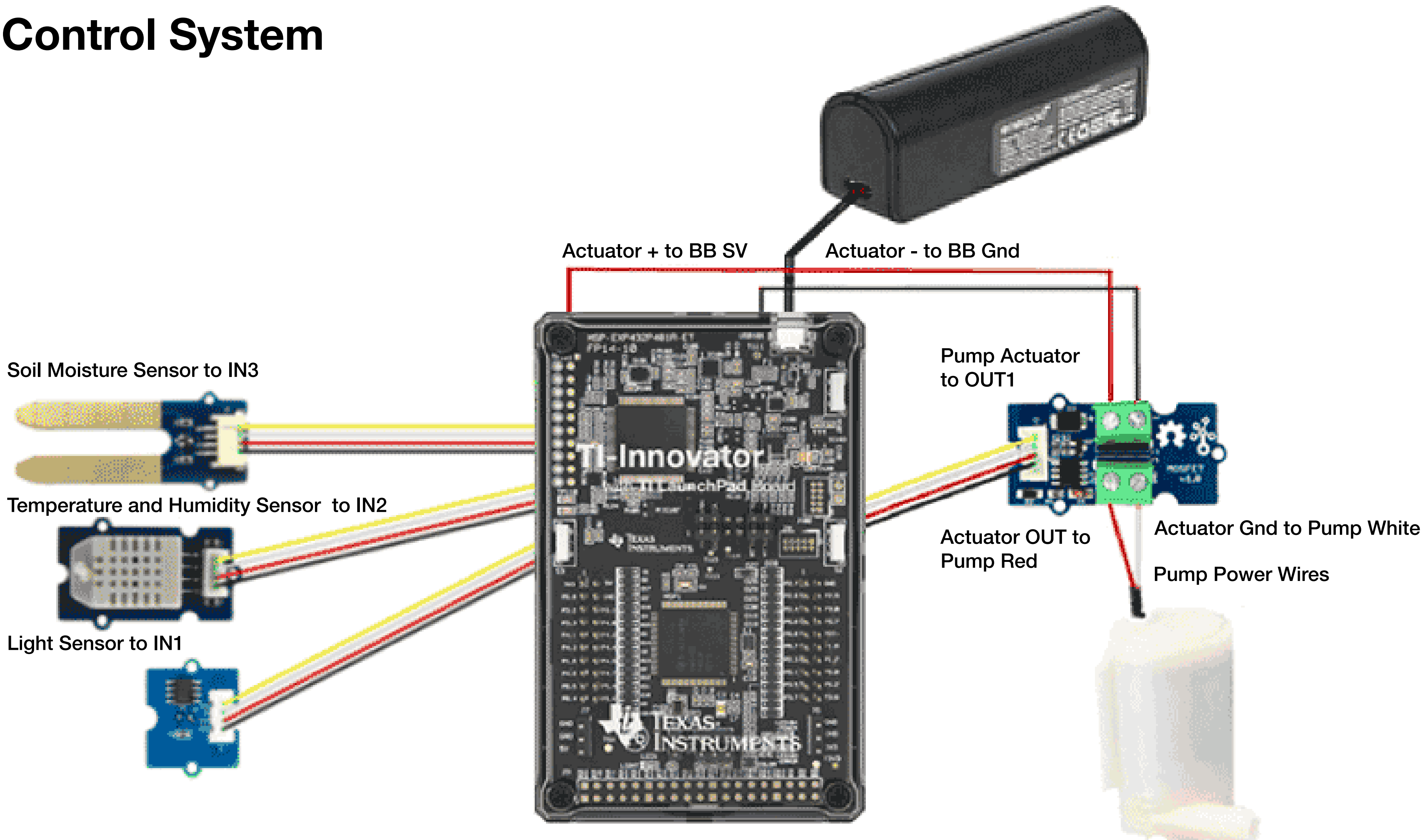
The students will work on the construction of their project. Supplies for building should be made available.

» Teams assemble, program, troubleshoot, etc. their solutions

» Teams share their progress and observations

Sessions 8 and 9 Background

Control System



Example program

```
Prgm

Send "CONNECT ANALOG.OUT 1 TO OUT 1”
Send "CONNECT LIGHTLEVEL 1 TO IN 1”
Send "CONNECT DHT 1 TO IN 2”
Send "CONNECT MOISTURE  1 TO IN 3”
Send "RANGE LIGHTLEVEL 1 0 100”
Send "RANGE MOISTURE 1 0 100”

Disp "Warming Up Sensors!”
temperature:=-273
While temperature<-270
Send "READ DHT 1 TEMPERATURE”
Get temperature
Wait 1
EndWhile
```

```
For n,1,50
  Send "READ LIGHTLEVEL 1”
  Get light
  Send "READ  MOISTURE 1”
  Get moisture
  Send "READ  DHT 1 TEMPERATURE”
  Get temperature
  Send "READ  DHT 1 HUMIDITY”
  Get humidity
  Disp "T=",temperature,"H=",humidity,"L=",light,"M=",moisture
  Wait 1

  If light<50 and moisture<5 Then
    Send "SET ANALOG.OUT 1 120”
  Else
    Send "SET ANALOG.OUT 1 0”
  EndIf
EndFor

Send "SET ANALOG.OUT 1 0”
EndPrgm
```

(Continued on next page)

Example program *(cont.)*

Some possible challenges/tasks

- Set the pump to a pulse pattern that gives time for the water to sink into the soil
- Incorporate humidity and temperature into the decision to pump
- Update the main loop of the program to be a While loop (instead of a For loop) that can run continuously until you give an input to stop the program

Session 10 Agenda Example

Gallery-walk and presentations with peer review of solutions, certificates, etc.

- Teams share their final solutions
- Reflection discussion on the project
- Recognition ceremonies

Session 10 Background

Computational Thinking

NGSS Science and Engineering Practices

Using Mathematics and Computational Thinking

http://ngss.nsta.org/Practices.aspx?id=5