

A curriculum for advancing do-it-yourself (DIY) environmental sensing networks to monitor water quality and quantity in streams and rivers

Dan Bondanza, Steve Hicks, Dave Arscott, Anthony Aufdenkampe, David Bressler, Tara Muenz, Peter Wise, Sara Damiano, Steve Kerlin



Introduction

Here we present a Science, Technology, Engineering, and Math (STEM) curriculum targeted for secondary education that introduces an approach to building, customizing, and deploying real-time data logging environmental sensor kits. The curriculum uses the topic of water quality monitoring to introduce students to a diverse set of STEM topics and enables students to collect advanced water quality data on healthy and impaired waterways throughout the U.S. Students will develop skills in critical thinking, data interpretation, watershed science, and use of technology for scientific research. The technology that serves as the basis for the curriculum is the “EnviroDIY Mayfly Data Logger™”, developed by Stroud Water Research Center™. The Mayfly board can be connected to a vast amount of sensors, which perform many monitoring components such as: water depth, air and water temperature, specific conductivity, turbidity, dissolved oxygen, precipitation, solar/light levels, and/or soil moisture. Students will learn about water quality and quantity and how naturally occurring conditions differ from human impacted conditions.

Science

Students will learn about water quality and quantity of local streams/rivers using the Mayfly board. Numerous labs have been developed as part of the curriculum to teach students to use the Mayfly boards in short-term experiments, as well as long-term deployments, to assess the physical, hydrological, and chemical dynamics of their study systems (e.g., stream, river, lake, or pond). The curriculum will lead students through the details of many of the natural and anthropogenic factors that affect both the quality and quantity of their aquatic ecosystem. Students will gain insight into how these variables respond to environmental conditions, how certain variables influence one another (e.g., water temperature effects on dissolved oxygen concentrations), as well as methods and procedures that can help to improve the quality of their aquatic ecosystem.

Features of the EnviroDIY Mayfly Data Logger

- A: MicroUSB port
- B: Power switch
- C: MicroSD connector
- D: Push button
- E: MicroSD card Socket
- F: 20-pin header for analog pins
- G: Auxiliary 16-bit Analog-to-Digital Connectors
- H: Grove Digital ports
- I: I²C port
- J: 5V boost converter
- K: 20-pin header for digital pins
- L: Battery backup for real time clock
- M: LiPo battery connectors
- N: Red & Green LEDs
- O: Solar panel connector
- P: FTDI programming port
- Q: Bee Module Socket
- R: Processor
- S: Real time clock

Technology

Students will begin exploration of their EnviroDIY Mayfly boards and the associated Arduino programming language before setting up long-term deployments. Students will gain insight into the basic programming that enables the board to communicate with sensors/probes and to transmit data to off-site databases. Simple code instructions will be included in the curriculum for students to get hands on experience into programming as well as learning basic programming syntax. Students will learn about loops, counters, input/output commands, using built in libraries, manipulating timers and many other basic programming skills and approaches. Students will also gain an understanding of digital versus analog inputs/outputs, as well as learn to use a solar panel/rechargeable battery system to power their Mayfly boards and the associated sensors.

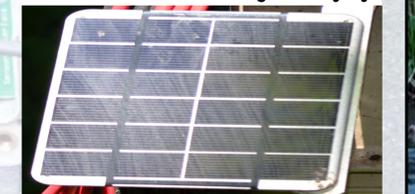
Code for Mayfly Data Logger

```

#include <Arduino.h>
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include <RTClib.h>
#include <Adafruit_BMP280.h>
#include <Adafruit_SHT1x.h>
#include <Adafruit_Grove_I2CGain.h>
#include <Adafruit_Grove_Serial.h>
#include <Adafruit_Grove_Temperature.h>
#include <Adafruit_Grove_Humidity.h>
#include <Adafruit_Grove_AirQuality.h>
#include <Adafruit_Grove_Power.h>
#include <Adafruit_Grove_Solar.h>
#include <Adafruit_Grove_Battery.h>
#include <Adafruit_Grove_LEDs.h>
#include <Adafruit_Grove_FT232RL.h>
#include <Adafruit_BeeModule.h>

// Define pins
const int LED_RED = 13;
const int LED_GREEN = 12;
const int BEEP = 11;
const int SW = 2;
const int A0 = 0;
const int A1 = 1;
const int A2 = 2;
const int A3 = 3;
const int A4 = 4;
const int A5 = 5;
const int A6 = 6;
const int A7 = 7;
const int A8 = 8;
const int A9 = 9;
const int A10 = 10;
const int D0 = 23;
const int D1 = 24;
const int D2 = 25;
const int D3 = 26;
const int D4 = 27;
const int D5 = 28;
const int D6 = 29;
const int D7 = 30;
const int D8 = 31;
const int D9 = 32;
const int D10 = 33;
const int D11 = 34;
const int D12 = 35;
const int D13 = 36;
const int D14 = 37;
const int D15 = 38;
const int D16 = 39;
const int D17 = 40;
const int D18 = 41;
const int D19 = 42;
const int D20 = 43;
const int D21 = 44;
const int D22 = 45;
const int D23 = 46;
const int D24 = 47;
const int D25 = 48;
const int D26 = 49;
const int D27 = 50;
const int D28 = 51;
const int D29 = 52;
const int D30 = 53;
const int D31 = 54;
const int D32 = 55;
const int D33 = 56;
const int D34 = 57;
const int D35 = 58;
const int D36 = 59;
const int D37 = 60;
const int D38 = 61;
const int D39 = 62;
const int D40 = 63;
const int D41 = 64;
const int D42 = 65;
const int D43 = 66;
const int D44 = 67;
const int D45 = 68;
const int D46 = 69;
const int D47 = 70;
const int D48 = 71;
const int D49 = 72;
const int D50 = 73;
const int D51 = 74;
const int D52 = 75;
const int D53 = 76;
const int D54 = 77;
const int D55 = 78;
const int D56 = 79;
const int D57 = 80;
const int D58 = 81;
const int D59 = 82;
const int D60 = 83;
const int D61 = 84;
const int D62 = 85;
const int D63 = 86;
const int D64 = 87;
const int D65 = 88;
const int D66 = 89;
const int D67 = 90;
const int D68 = 91;
const int D69 = 92;
const int D70 = 93;
const int D71 = 94;
const int D72 = 95;
const int D73 = 96;
const int D74 = 97;
const int D75 = 98;
const int D76 = 99;
const int D77 = 100;
const int D78 = 101;
const int D79 = 102;
const int D80 = 103;
const int D81 = 104;
const int D82 = 105;
const int D83 = 106;
const int D84 = 107;
const int D85 = 108;
const int D86 = 109;
const int D87 = 110;
const int D88 = 111;
const int D89 = 112;
const int D90 = 113;
const int D91 = 114;
const int D92 = 115;
const int D93 = 116;
const int D94 = 117;
const int D95 = 118;
const int D96 = 119;
const int D97 = 120;
const int D98 = 121;
const int D99 = 122;
const int D100 = 123;
const int D101 = 124;
const int D102 = 125;
const int D103 = 126;
const int D104 = 127;
const int D105 = 128;
const int D106 = 129;
const int D107 = 130;
const int D108 = 131;
const int D109 = 132;
const int D110 = 133;
const int D111 = 134;
const int D112 = 135;
const int D113 = 136;
const int D114 = 137;
const int D115 = 138;
const int D116 = 139;
const int D117 = 140;
const int D118 = 141;
const int D119 = 142;
const int D120 = 143;
const int D121 = 144;
const int D122 = 145;
const int D123 = 146;
const int D124 = 147;
const int D125 = 148;
const int D126 = 149;
const int D127 = 150;
const int D128 = 151;
const int D129 = 152;
const int D130 = 153;
const int D131 = 154;
const int D132 = 155;
const int D133 = 156;
const int D134 = 157;
const int D135 = 158;
const int D136 = 159;
const int D137 = 160;
const int D138 = 161;
const int D139 = 162;
const int D140 = 163;
const int D141 = 164;
const int D142 = 165;
const int D143 = 166;
const int D144 = 167;
const int D145 = 168;
const int D146 = 169;
const int D147 = 170;
const int D148 = 171;
const int D149 = 172;
const int D150 = 173;
const int D151 = 174;
const int D152 = 175;
const int D153 = 176;
const int D154 = 177;
const int D155 = 178;
const int D156 = 179;
const int D157 = 180;
const int D158 = 181;
const int D159 = 182;
const int D160 = 183;
const int D161 = 184;
const int D162 = 185;
const int D163 = 186;
const int D164 = 187;
const int D165 = 188;
const int D166 = 189;
const int D167 = 190;
const int D168 = 191;
const int D169 = 192;
const int D170 = 193;
const int D171 = 194;
const int D172 = 195;
const int D173 = 196;
const int D174 = 197;
const int D175 = 198;
const int D176 = 199;
const int D177 = 200;
const int D178 = 201;
const int D179 = 202;
const int D180 = 203;
const int D181 = 204;
const int D182 = 205;
const int D183 = 206;
const int D184 = 207;
const int D185 = 208;
const int D186 = 209;
const int D187 = 210;
const int D188 = 211;
const int D189 = 212;
const int D190 = 213;
const int D191 = 214;
const int D192 = 215;
const int D193 = 216;
const int D194 = 217;
const int D195 = 218;
const int D196 = 219;
const int D197 = 220;
const int D198 = 221;
const int D199 = 222;
const int D200 = 223;
const int D201 = 224;
const int D202 = 225;
const int D203 = 226;
const int D204 = 227;
const int D205 = 228;
const int D206 = 229;
const int D207 = 230;
const int D208 = 231;
const int D209 = 232;
const int D210 = 233;
const int D211 = 234;
const int D212 = 235;
const int D213 = 236;
const int D214 = 237;
const int D215 = 238;
const int D216 = 239;
const int D217 = 240;
const int D218 = 241;
const int D219 = 242;
const int D220 = 243;
const int D221 = 244;
const int D222 = 245;
const int D223 = 246;
const int D224 = 247;
const int D225 = 248;
const int D226 = 249;
const int D227 = 250;
const int D228 = 251;
const int D229 = 252;
const int D230 = 253;
const int D231 = 254;
const int D232 = 255;
    
```

Solar Panel Powering a Mayfly



Engineering

Students will need to fabricate waterproof housing for their Mayfly boards to ensure long-term sustainability of their deployment in the environment. Students will design cases for the Mayfly boards as well as adaptors and mounting hardware for various peripheral devices that are connected to the board. Students will have to find suitable locations for keeping their Mayfly boards dry and safe, setting up solar panels that ensure maximum efficiency and mounting sensors to ensure accurate readings while minimizing the risk for interrupted signals or environmental hazards (e.g., flooding, wild animal and human curiosity). Brackets and various other devices will be constructed and manipulated by the students to help with both short and long-term deployment and upkeep of the Mayfly boards.



Mayfly Data Logger in waterproof housing for long-term water quality monitoring

Mathematics

Math skills are developed during units that introduce techniques for the management and analysis of time-series data recorded from environmental sensors. Using statistics, students will calculate regressions and residuals for their data and try to model predictions over longer periods of time. Students will use multi-degree polynomials as well as sine functions to model their data. Students will be asked to present their data in different formats using various algorithms and graphical analysis. While observing trends over time, students will gain a deeper understanding of the mathematics involved in long-term monitoring. Using these models, students will begin to predict trends and identify possible solutions to stream/river management issues, while avoiding major pitfalls such as over-extrapolation and using short-term models to predict long-term effects.

Turbidity

Water Depth and Conductivity

Dissolved Oxygen

mean depth = $\frac{\text{area}}{\text{top width}}$
hydraulic radius = $\frac{\text{area}}{\text{wetted perimeter}}$

Water Temp: 2.9
Logger Temp: 0.8

White Clay Creek