Research Experience for Science and Math Teachers in Internet of Things (IoT)

A BioMath Approach to Introduce Basic Programming

Fabiola Guillaume

Orange County Public Schools—Edgewater High School, Orlando, FL 32804

UCF College of Engineering & Computer Science

NSF

Summary

Over the course of six weeks, UCF in support to STEM education in K-12 classrooms initiated a second cohort of 11 science and math teachers intended to equip educators of the region with an explorative background in IoT and quality experiences in engineering design and technology. Through a series of lectures and lab practices the following schedule served as guideline for exploration of each research project module within a one to two–weeks timeline:

Weeks 1 and 2: *Sensor Device Module.*

Option 1—Working principles: *Design and Fabrication of Environmental Sensors.*

- Material Synthesis
- Photomask design
- Cleanroom Microfabrication
- Device packaging

Weeks 3 and 4: *Interface and Testing Module*

Working principles: Analog and digital circuit basics

- Number system & Boolean Algebra
- Introduction to C-programming language.

Week 5: *Software and Networking Module.*

Working principles: System Software and Networking

- Introduction to Java programming language.
- Basics of computer networking—Layered model
- Raspberry Pi Java programming.

Week 6: *Mobile Programming*

Working principles: Mobile operating systems

- Introduction to Android
- Writing a Web-based Android program

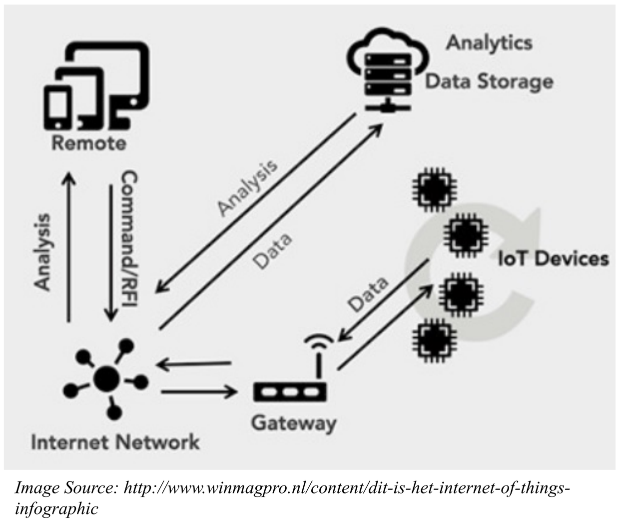


Image Source: http://www.winmagpro.nl/content/dit-is-het-internet-of-things-infographic

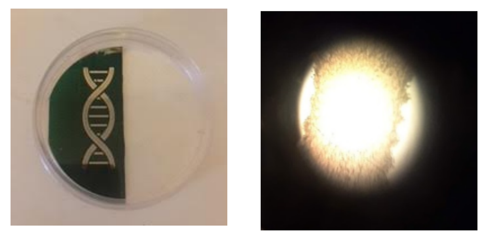
Research Activities

Much like an ecosystem, IoT is an environment or network where everything and everyone is connected. The following research activities were completed under the supervision and assistance of UCF faculty professors and PhD resident students in material, electrical and computer engineering.

Bottom-Up IoT Architectural Design: Design and Fabrication of Environmental Sensors.

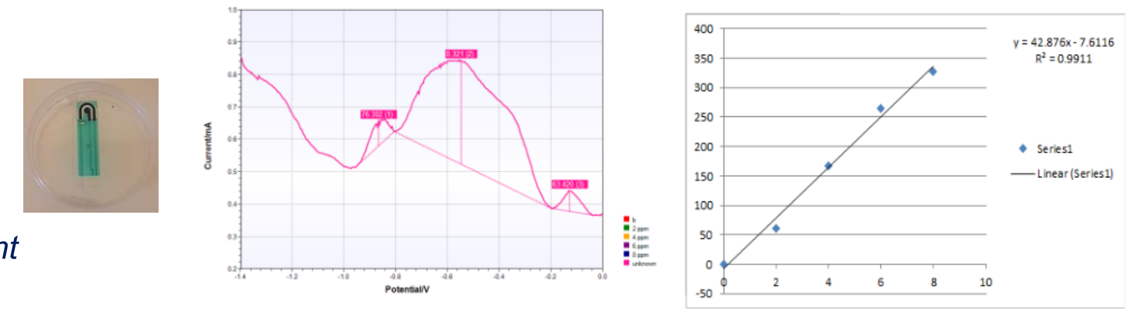
- *Photolithography Fabrication Method: Silicon Circuit Chip*

  Procedures:
  1. *AutoCAD template design*
  2. *Draft sight*
  3. *Plating with metal (ex: Titanium)*
  4. *Cleaning with acetone and nitrogen air*
  5. *Spin-coating with photoresist*
  6. *Soft Bake*
  7. *Apply Mask*
  8. *UV exposure*
  9. *Hard baking*
  10. *Developing with base*
  11. *Etching (with acid)*



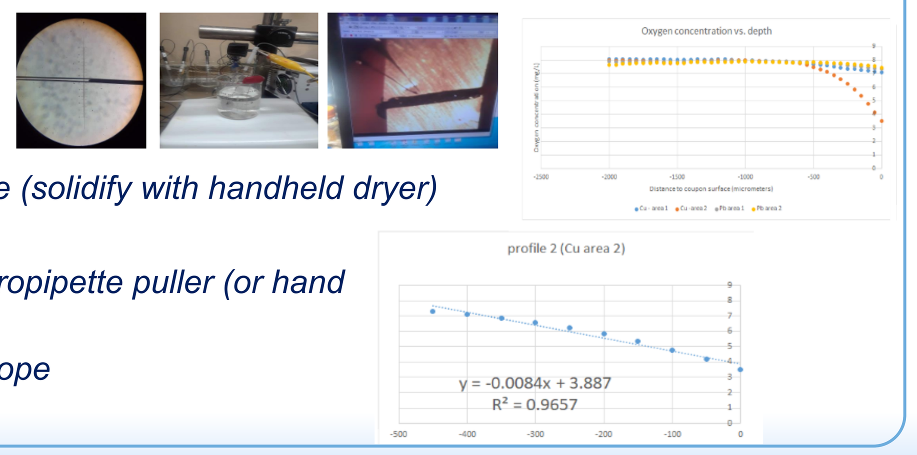
- *Screen Printing Fabrication Method: Bi/Chitosan Electrochemical sensor*

  Procedures:
  1. *Silver paste printing*
  2. *Dry time*
  3. *Carbon molding*
  4. *Drying time*
  5. *UV pasting*
  6. *Curing under UV light*
  7. *Aluminum cleaning*
  8. *Edging and cutting*



- *Microsensor Fabrication and Testing*

  Procedures:
  1. *Stripping off Cu2+ wire coating*
  2. *Connecting Cu/Ti wires using Ag paste (solidify with handheld dryer)*
  3. *Casting wire with capillary glass*
  4. *Pulling Ti wire into needle using a micropipette puller (or hand pulling technique)*
  5. *Measure needle length under microscope*
  6. *Etch Ti with Cyanic acid if needed*



Lesson Plan

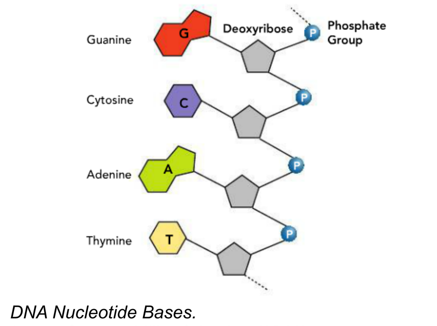
The following lesson activities have been developed from content materials in interface modules 2 and 3. Basic programming concepts will serve as framework for the mental model of DNA processing genetic information to synthesize proteins. A goal of this research lesson is to provide a concise language of science and programming to describe conceptual understanding that may unfamiliar to students.

Day 1—Lesson Plan: *DNA is Binary!*

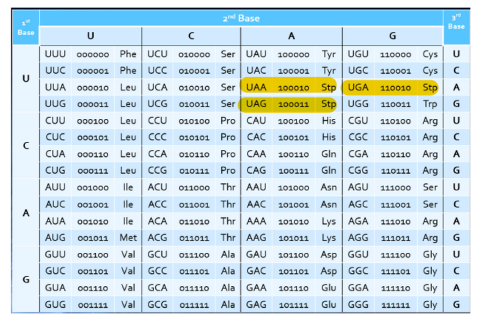
Lesson Activity: *Writing Binary Numbers. (20 minutes)*

Students will be able to represent DNA data using binary and hexadecimal numeration systems. Building from prior knowledge that while representing quantities, numbers are not material things, students will assign values to DNA bases and practice conversion between the two different numeration systems. The two-bit identifiers used to represent the four nucleotides is drawn from the same principle employed by computer engineers using a two-digit numeration system to represent the condition of an electrical switch (0 = "off"; 1 = "on").

DNA Bases	Radix-10	Radix-2
T	0	00
C	1	01
A	2	10
G	3	11



DNA Nucleotide Bases.
Image Source: Miller & Levine (2010)

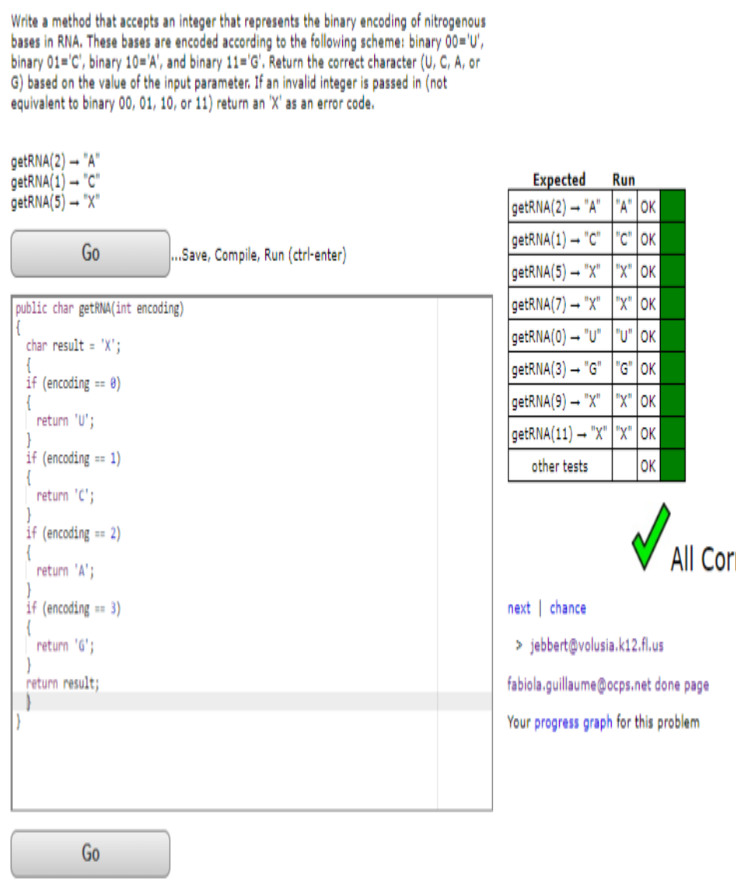


6-bit RNA Codon Chart. Image Source: Nemzer (2017)

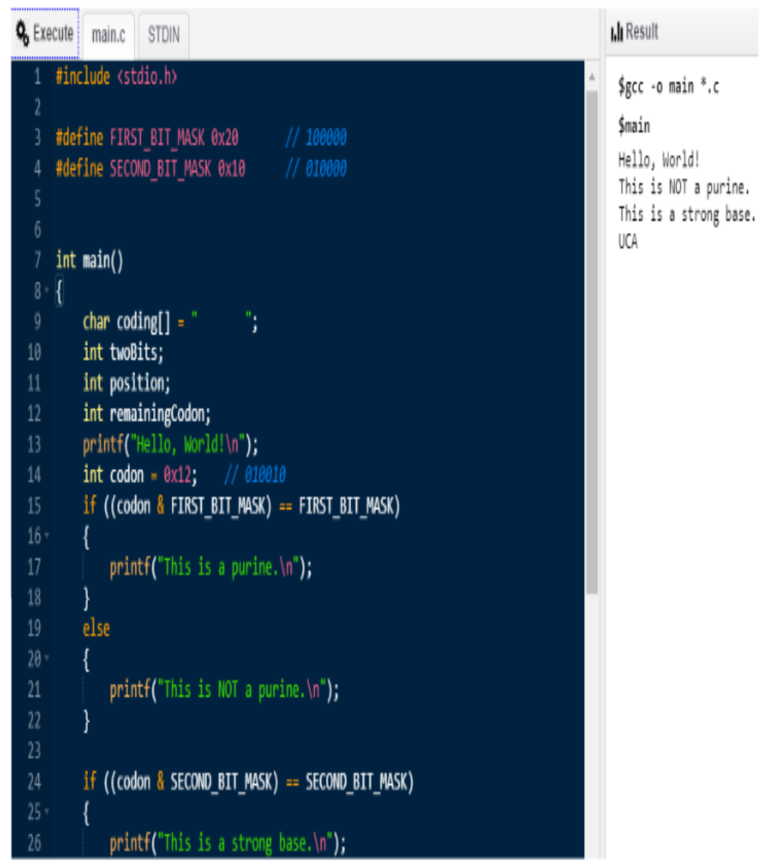
Day 2—Lesson Plan: *Basics of Computer Programming Languages (C and Java Programming)*

This lesson will provide students with key understanding in program language structure while reinforcing taught biology concepts of DNA and RNA base pairing rules, replication and protein synthesis. Students will practice coding using free online websites.

Example of Codingbat Practice Problem



Example of Coding Exercise on Compile C

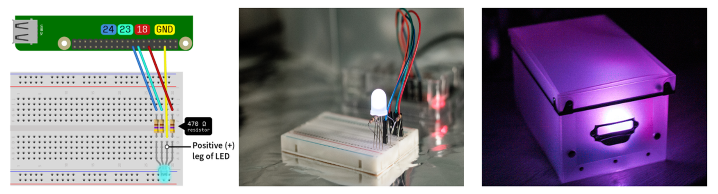


Day 3—Lesson Plan:

Lesson Activity: DNA Flashing LED Box

Student will use basic coding language to program flashing LED lights on Raspberry Pi. Each base is assigned a specific color. Students will play the role of RNA, making a copy of the DNA message (sequence of flashing colors) to synthesize a short polypeptide chain.

- Adenine (A) = Green;
- Thymine (T) = Yellow;
- Cytosine (C) = Blue;
- Guanine (G) = Red.



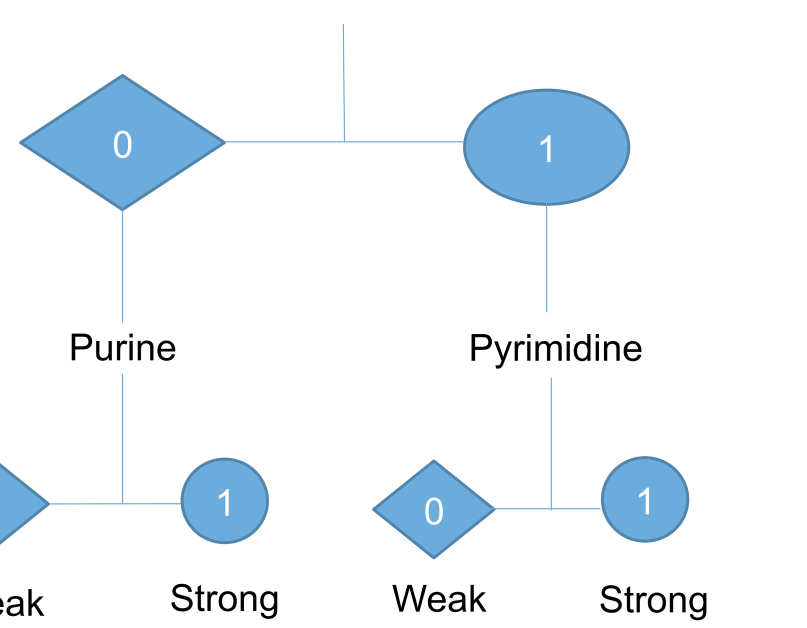
Central Dogma of Life

DNA ⟶ RNA ⟶ Proteins

List of Materials:
- Raspberry Pi
- Breadboard with jumper wires
- RGB LED
- 3x 470 ohms resistors
- Transparent and frosted storage box

Lesson Learned and Assumptions

- IoT assisted instruction helps bring relevance to daily teaching.
- Terminology used to describe scientific concepts may be unfamiliar to teachers and students alike.
- A multidisciplinary approach to teaching conceptual understanding of complex biology topics might help bridge the gap in student learning and achievement.
- Practicing algorithm may help improve critical thinking skills in students.



DNA Bases Algorithm: Is it a purine (0) or pyrimidine (1)? Is it a strong base/acid (1) or a weak base/acid (0)?

Implementation Strategy

The strategies below represent a suggestive list of implementation techniques that will guide content delivery in biology, provide opportunities in programming and help develop critical thinking skills in students. These have been selected for their high effect size in keeping students engaged as well as their impact on achievement and learning (Hattie, 2012). However, it must be understood that a progressive evaluation of these methods might indicate a need for modification if proven ineffective and insufficient at meeting the students' needs.

- Guided Inquiry:
  - Teacher guides students through procedure (Written and verbal instructions provided)
- Gradual Release:
  - Teacher Model → Whole Group Practice → Collaborative Pair → Independent Practice
- Collaborative Grouping:
  - Heterogeneous grouping of students for peer support and tutoring
- Differentiated Instructions: Days 2 and 3
  - Low and Intermediate Students: Complete missing part of the coding
  - Advanced Students: Write full program
- Formative Assessments:
  - Daily bell work recap from previous lesson
- Immediate Feedback:
  - Day 2: Immediate response check with table of results

Acknowledgments

RET Site: COMET Program, College of Engineering and Computer Science, University of Central Florida. This content was developed under National Science Foundation grant EEC-1611019.

References

Miller, K. R., Levine, J, S. Biology—Elevate Science Florida, Boston, Mass. – Pearson -2010.
Nemzer, L. R. (2017). A binary representation of the genetic code. Biosystems 155, 10-19.
ACT: the Artemis Comparison Tool.