Physical Programming

Contact Information

Dan Mack: d_mack@sau9.org

Course Description

The course introduction to Physical Computing with Arduino curriculum allows students to learn about the knowledge and basic programming concepts (control structures, variables, functions, etc.) in order to control a physical device. Students will perform basic physical tasks using LEDs, motors, and sensors to see how computer programming gives physical devices the ability to interact with their environment.

Students will work on small scale devices and virtual circuits to learn the ins and outs of physical computing. Students will also get a chance to explore how they can use these skills on real world projects like creating a hydroponic garden monitor station, weather station or motion detection devices.

Student Learning Outcomes

PC. 1: Students will have **programming knowledge** skills that allow students to develop solutions to novel problems.

- EU 1.1 Programs can be developed for creative expression, to satisfy personal curiosity, to create new knowledge, or to solve problems (to help people, organizations, or society).
- EU 1.2 People write programs to execute algorithms.
- EU 1.3 Programming is facilitated by appropriate abstractions.
- EU 1.4 Programs are developed, maintained, and used by people for different purposes.
- EU 1.5 A variety of abstractions built on binary sequences can be used to represent all digital data.
- EU 1.6 Multiple levels of abstraction are used to write programs or create other computational artifacts.
- EU 1.7 Computing facilitates exploration and the discovery of connections in information.
- EU 1.8 There are trade-offs when representing information as digital data.

PC. 2: Students will critically **computational skills** technical documentation and code.

- EU 2.1 Programming uses mathematical and logical concepts.
- EU 2.2 Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages.
- EU 2.3 Algorithms can solve many, but not all, computational problems.
- EU 2.4 Models and simulations use abstraction to generate new understanding and knowledge.
- EU 2.5 People use computer programs to process information to gain insight and knowledge.

PC. 3: Students will effectively **design and communicate** to be contributing team members

- EU 3.1 Creative development can be an essential process for creating computational artifacts.
- EU 3.2 Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
- EU 3.3 Computing can extend traditional forms of human expression and experience.
- EU 3.4 Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.
- EU 3.5 Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.

PC. 4: Students will be efficient self learners that understand **personal success skills** are critical to career success.

- EU 4.1 Computing enhances communication, interaction, and cognition.
- EU 4.2 Computing enables innovation in nearly every field.
- EU 4.3 Communication comes in a variety of forms and is necessary
- EU 4.5 Knowing your career path allows you to set goals and plan long term

Course Requirements

This course is designed to teach and reinforce understanding of computer science concepts by applying them to physical devices. It is not required but strongly recommend that students have completed algebra I prior to taking this course.

Student Evaluation

Work Ethics	25%
Projects	35%
Challenges	25%
Exercises	15%

LED Strip Bike Device

Students will go through the basics of the Arduino device, such as how to light up and change the brightness of LEDs, and learn how variables can be used to write more versatile programs. Students will build circuits to control LEDs and motors with the Arduino and explore how pseudocode can be used to structure programs from the start. They will create a program and a device which operates LED strips which can be added to a bike's wheel to create a colorful display.

Competency assessed: PC.1

Hydroponic Garden Monitoring Station

Students will learn how to apply control structures, such as if/else statements and loops to create programs that will react to the outside world. They will build programs that use sensors to detect temperature, light, and distance and make decisions based on the information collected. Students will create a monitoring station for a hydroponic garden which will look at soil moisture, water level and temperature of the garden.

Competency assessed: PC.1, PC.2

Final Project

Students will have a chance to explore all of the capabilities of the Arduino on their own! They will research, explore, and teach their peers about new sensors, follow directions to build an advanced device, and have a chance to create their very own Arduino machine.

Competency assessed: PC.3, PC.4

Ethical Behavior

This course's philosophy on academic honesty is best stated as "be reasonable." This course recognizes that interactions with classmates and others can facilitate mastery of the course's material. However, there remains a line between enlisting the help of another and submitting the work of another.

The essence of all work that you submit to the course must be your own. Collaboration on problems is not permitted (unless explicitly stated otherwise) except to the extent that you may ask classmates and others for help so long as that help does not reduce to another doing your work for you. Generally speaking, when asking for help, you may show your code or writing to others, but you may not view theirs, so long as you and they respect this policy's other constraints.

Late work

Assignments are due on the announced due date. It is your responsibility to keep up with class activities and assignments and request missing assignments due to absence. Upon returning from an excused absence, students will be given two days for each day absence to make up missed assignments. All students work at a different pace and will be graded primarily on their quality of work and productivity level during class. As long as the students are highly productive each day and producing work of high quality, they will receive excellent grades. Extensions will also be available upon request.

Food/Drink

Food and drinks are not permitted near the computer stations. There will be dedicated stations for water bottles or snacking.

Electronic Devices

Phones and other electronic devices are allowed in class if they do not become a distraction (texting, playing games, checking social media, web browsing, etc.). Students who are regularly off task or behind in their work, will have their phone privileges revoked. However, during instructional time, tests and quizzes, electronic devices are not to be used at all (unless directed to by the teacher). The CTC has a general no cellphones during instruction time policy. This means your phone should not be visible during lecture, going over examples, group work. You will get one warning to put the phone away. If we are working on individual work you are welcome to ask to use your phone, if you don't ask permission you will get a

warning. After the warning your phone will go into one of the phone jails and you can collect it at the end of the class.

Computer Use

The classroom computers and related devices are to be used for classwork only. Do not download any files or programs not related to your classwork. Do not change the Login screen background. Do not install any program without permission of the teacher. Do not run any unapproved programs (Minecraft, Call of Duty, Halo, etc.), even from a network, external drive or remote device. Do not view or download any images, videos, or sound files that are offensive, racist, promote violence or drug use, etc.

Outline

- **LED Strip Bike Device**
  - Intro to physical computing
  - Goal Setting
  - Comments
  - Pseudocode
  - Analog vs. digital
  - Variables
  - Breadboards
  - Potentiometers
  - Debugging

- **Hydroponic Garden Monitoring Station**
  - For loops
  - While loops
  - Variables
  - If statements
  - If/else statements
  - Using buttons
  - Using motors
  - Operators (arithmetic, comparison, and logical)
  - Using sensors (ultrasonic range finder, light sensor, temperature sensor)
  - Functions and parameters

- **Final Project**
  - Challenges
  - Explore a new sensor
  - Build a step-by-step project