



## Talking Points

### 1. Smart Systems

- A. What makes a device a **computer**?
  - a. Input, Output, Processing, Storage/Memory
  - b. Examples: Raspberry Pi, Arduino (**microcontroller**, computer that is more specialized)
- B. Computers comprise **hardware** and **software**
  - a. Hardware = physical
    - i. Input via **sensors**: gets information (e.g. microphone, button)
    - ii. Output via **actuators**: produces change (e.g. printer, speaker, LED)
  - b. Software = instructions as **programs** (or **applications**, “apps”)
    - i. An **Operating System** (OS) is a program that allows other programs to co-exist, sharing computer resources effectively & securely (e.g. Windows, macOS, Linux)
    - ii. A **shell** (or **terminal**) is a program to executes OS commands
- C. Computers are circuits, and so really only understand voltages
  - a. 0 = LOW voltage ( $\leq 0.9V$ , including **ground**)
  - b. 1 = HIGH voltage ( $\geq 3.0V$ )
- D. Since humans aren’t great at 0/1’s, we write instructions in a **programming language**
  - a. Example languages: C/C++ (Arduino), Python (RPi), Java (CS1), ...
  - b. Typical sequence: write **source code** in a programming language, **compile** into 0/1’s (may reveal errors), run program
  - c. We don’t always have to write code from scratch – a **library** is someone’s code that has been made useful for other situations
- E. The Arduino uses programmable **pins** that make it easy to talk to hardware via software
  - a. Converts sensor voltages to numbers (e.g. if button is pressed...)
  - b. Converts numbers in code to voltage for actuators (e.g. turn on/off the motor/LED)
- F. A **network**: two or more computers that are connected and exchange information
  - a. Need to agree to a **protocol**, a “language” by which to communicate effectively
  - b. Typically either **peer-to-peer** (communication amongst equals) or **client/server**, where one computer makes requests (client) and the other provides a service (server)
- G. The most commonly known network is the... **Internet**
  - a. Each computer has an **IP Address** (e.g. 69.43.65.41)
    - i. Devices can have a shortcut **domain name** (or **host name**) which gets converted to an IP address via **DNS** (e.g. call “grandma” instead of 617-...)
  - b. So that a computer can talk to more than one computer, potentially using more than one protocol, it has many **ports** for communication (e.g. 80 for web, 22 for SSH)
  - c. The **Internet of Things (IoT)** refers to connecting lots of everyday sensors/actuators to the Internet (e.g. light bulbs, cameras, security systems, refrigerators...)

**Project Overview.** We are building an Internet-enabled electronic lock (your first IoT device!). The Arduino, programmed in C++, will store a 4-digit passcode and has a simple protocol: “move” locks/unlocks via motor, given a correct passcode; “change” changes the code, given a correct old code. The Raspberry Pi (RPi) will communicate via USB with the Arduino to provide these commands. You will be able to connect wirelessly to it via a hostname (using your computer or phone), and it will serve a website, in Python, at the 8080 port. When you submit the web form, you will control the lock via communication with the Arduino.

## 2. The Basics of Programming

- A. There are many different programming languages, each with strengths/weaknesses, but they all have common components
- B. Sequential execution
  - a. In most languages, the computer executes one line of code at a time, in the order it is written – start somewhere, execute the next line, then the next line, ...
  - b. Your job as a programmer is to provide a sequence of commands that gets the computer to do what you want
  - c. A **comment** is one-or-more lines that you tell the computer to ignore
    - i. Arduino: anything after //
- C. Information storage
  - a. Most of the time you will need to store information in your program
  - b. A **variable** is a named piece of memory where you can keep information
  - c. A variable usually has a **type**, such as number, character, word(s)
  - d. Unlike variables in algebra, this is re-usable space – you can get the current value and change it as you wish
- D. Code grouping
  - a. It is often useful to group together code into **functions** – since programs get big, this helps find pieces, and also lets you re-use the same parts later (e.g. write a “blink” function on the Arduino and use it over and over for red/green/blue lights)
  - b. When an Arduino powers on, it first executes code found in the “setup” function
  - c. Any function can call any other function (including one in someone else’s library!) – by building on previously written code, your program becomes more powerful!
- E. Code repetition
  - a. Computers are very efficient at doing simple tasks over and over – this is a **loop**
  - b. Most languages have ways of telling the computer to execute code a certain number of times, or until a certain condition does [not] hold
    - i. Example: loop over each of the 4 characters in the passcode and compare them to the 4 characters that were submitted from the website
  - c. After the Arduino runs “setup”, it then calls the “loop” function over and over
- F. Conditional execution
  - a. Very commonly you want the computer to do something **if** a condition is true
    - i. Example: IF the button is pressed, reset the passcode to “1234”
  - b. This naturally combines with loops
    - i. Example: loop over each of the 4 characters in the passcode and compare them to the 4 characters that were submitted from the website – if one is different, bad passcode; if none were different, good passcode!
- G. References and resources
  - a. To be a good programmer, you need to know some basics, but then it is quite common to use Google to find reference documentation on a particular function/library/language, or ask a particular question of other programmers

**Arduino Code.** All the C++ code is found in unit3/server/htmaa/htmaa.ino – see if you can understand what is going on! It uses some functions built into the Arduino (Google if you aren’t sure!), such as using pins. It also uses the CmdArduino to make it easy to send commands like “move” over USB.

**RPi Code.** All the Python code is found in unit3/client/web.py – see if you can understand it! It is mostly using the Flask library to host a website, as well as a Serial library to communicate via USB.