AT THE END OF THIS LESSON, YOU WILL BE ABLE TO ANSWER THE FOLLOWING QUESTIONS.

- Know how to identify specific processes that a Control System must accomplish and understand how to implement that in a system
- Know how to map identified processes to their desired input and output and understand how to implement that in logic
- Know how to create simple functions that relate the input, process, and output identified in Control Systems

WHAT IS A FUNCTION?

Being able to segment your code into manageable chunks allows a programmer to create module pieces of code that perform different tasks and then return to the area of code from when the function was “called”.

WHAT IS A FUNCTION?

The typical cases for creating functions are:

- You have a lot of code in the main the loop and you need to move it elsewhere
- You are repeating the same code over and over again.

WE'VE ALREADY SEEN FUNCTIONS

- `void setup(){}`
- `void loop(){}`
- `pinMode(pin, INPUT|OUTPUT);`
- `digitalWrite(pin, HIGH|LOW);`
- `digitalRead(pin);`

THERE IS A LOT OF SIMILARITY BETWEEN FUNCTIONS AND VARIABLES

- Both have
-
    - A data type; and
    - A name (or a label)

```
int myMultiplyFunction(int a, int b){
    return a * b;
}
```

MORE ABOUT FUNCTIONS

However, functions also have input that they can manipulate and turn into a range of different outputs.

```
int myMultiplyFunction(int a, int b){
    return a * b;
}

void loop() {
    int num = myMultiplyFunction(int a, int b);
}
```

Speaker notes