

Trial Workday Snake

Kshitij Kapoor

September 6, 2025

Welcome to Trial Workday! Congratulations on making it this far.

Let's play a game!

You have been given:

1. An Arduino
2. A USB-A to USB-B Cable
3. Some code

The computer that the Arduinos can talk to is running a game of snake. But, it needs your inputs!

Figure out a way to get keyboard inputted data from your computer to the Arduino, and then from the Arduino to the snake game!

Note that the snake game expects your input in a very particular format. You must send **2 bytes** of data for every time you want to control your character.

Some really, really ~~annoying~~ thoughtful people made the protocol for sending data.

Player ID	Command ID	Magic Number
2 bits	3 bits	11 bits

In total, this makes 16 bits, or 2 bytes!

You should figure out how to:

1. Get code onto the Arduino
2. Get keyboard input to be read by Arduino
3. Configure that keyboard input to be messages to send to the snake game to control your character!

1 Player ID

If you're player 1, you would send a 1. This is: 0b01

If you're player 2, you would send a 2. This is: 0b10

If you're player 3, you would send a 3. This is: 0b11

2 Command ID

Command: START - 5 - 0b101

Command: UP - 1 - 0b001

Command: DOWN - 2 - 0b010

Command: LEFT - 3 - 0b011

Command: RIGHT - 4 - 0b100

3 Magic Number

The Magic Number is 0x010B. This is - 0b100001011

4 Some Background

4.1 What is binary?

Binary is a number system, just like the number system we use in daily life. The only difference is, instead of being based off of there being 10 individual symbols (0 - 9), it's based off of two individual symbols (0 - 1).

This makes it very, very useful. We can use 1 to represent something having electricity going through it, and 0 otherwise. That's how we get binary data in computers.

Note: the 0b at the front is just a way to note that this is a binary number that is following.

0b000 = 0

0b001 = 1

0b010 = 2

0b011 = 3

0b100 = 4

See a pattern?

4.2 What is hex?

Hex is another number system, but instead of base 2, it's base 16. This makes it useful to us, as programmers, to represent large data. Moreover, each hex digit can translate to 4 binary digits.

```
00 = 0x0 = 0b0
01 = 0x1 = 0b1
02 = 0x2 = 0b10
.
.
.
09 = 0x9 = 0b1001
10 = 0xA = 0b1010
11 = 0xB = 0b1011
12 = 0xC = 0b1100
13 = 0xD = 0b1101
14 = 0xE = 0b1110
15 = 0xF = 0b1111
```

4.3 What is a byte?

Binary, alone, is really cool! But, it's so incredibly granular. We can organize our data into bytes - just eight bits grouped!

This is what is considered a universal unit of data for a computer.

Your message your sending has two bytes. But, we want to encode three different things in it. How can we possibly do that?

4.4 Bit packing

Let's say I have a byte. I want to put two numbers into that byte, 3 and 5. Normally, these would be represented as:

```
3 = 0b00000011
5 = 0b00000101
```

Notice how the non-zero data only takes up the bottom 4 (or really 3) bits?

I'll, for the sake of argument, pack 3 at the higher order, and 5 and the lower order.

I can do that really simply! Just shift 3 up and take the 5.

This, in practice, looks like this:

```
(0b00000011 < 4) | (0b00000101)
```