Arduino Tutorial 2a

Objectives: Learn basic structure and syntax of Arduino C. Use Serial Monitor to take in and display user input. and as a tool for debugging. Use variables and conditional statements.

Materials and Set Up

- Arduino Uno microcomputer and USB cable
- Code from Tutorial 1
- Servo motor wired as in Tutorial 1

Exercise 1 – Reading in and displaying back data to/from Serial Monitor

1. Use your servo code from Tutorial 1
2. Change the setup() code to ask the user for start and end positions. You will save them in the existing variables, and display them back to the user to confirm.
3. Open the Serial Monitor and set the BAUD rate to 111520
4. In the code, look at the == sign - compare it to the single =
5. Notice the different between print and println (pronounced print line)
6. Notice how some words show up in different colors.

```
/*ServoDB modified 7/9/16
Controls a servo motor.
Takes in data through from the Serial Monitor, sends the motor to the
desired position, and displays it back to the screen
*/

…
void setup()
{
…

  Serial.println("Enter a start position for the servo: ");

// use a while loop to wait for a character in the serial monitor
  while(Serial.available() == 0)
   { }
  startPos = Serial.parseInt(); //read in an integer from serial monitor

  Serial.print(”you entered: “);
  Serial.println(startPos);   // display it to the screen

  Serial.println("Enter an end position for the servo: ");

   … create code as above, for endPos instead of startPos
}
```

Exercise 2 – Conditionals (If statements)

1. Compare the entered values with physical stops on the servo motors and set limits to their travel.
    a. Create messages indicating if the values are out of range
    b. Use If statements to compare and direct program flow
2. Try using If-Else statements to check that the values entered are in the range of the servo motor
    a. Combine logic statements in an OR statement like this: if ((startPos > 180 || startPos < 0))
    b. Or, create an AND statement like this: if ((startPos <= 180 && startPos >= 0))

```
if (startPos > 180)
      {
      Serial.print(“starting position is out of range “);
      }
else if (startPos < 0)
      {
      Serial.print(“starting position is out of range “);
      }
else
      { Serial.println(“Starting position is good”);}

Create checking code for endPos using an AND or OR statement
```

Exercise 3: Conditionals (While loops)

1. Create a Boolean variable called “ready” in the declarations section
2. Create a while loop in the setup() section that waits for startPos < endPos. Your program should continue to ask you for start and end position values until it sets ready to true.
3. Be careful with { }

```
boolean ready = false;

void setup()
{
…
while (!ready)   // wait for values with startPos < endPos
{
Serial.println("Enter a start position for the servo: ");
…
    if (startPos < endPos)
    { ready = true; }

   else
    { Serial.println ("ERROR: enter new positions with start < end");}

} // end of while loop
} // end of setup()
```