

Chapter 19 Written Problems Worksheet

Use this sheet to record your answers to the following Chapter 19 Challenge Problems.

19-1. The code contains **one mistake**. Do **not** compile this code. Instead, **identify** and then **correct** the mistake.

```
void loop() {
  digitalWrite(0, HIGH);
  delayMicroseconds(1000);
  digitalWrite(0, LOW);

  digitalWrite(1, HIGH);
  delayMicroseconds(2000);
  digitalWrite(1, HIGH);

  delay(20);
}
```

19-2. The code contains **one mistake**. Do **not** compile this code. Instead, **identify** and then **correct** the mistake.

```
void loop() {
  digitalWrite(0, HIGH);
  delayMicroseconds(100);
  digitalWrite(0, LOW);

  digitalWrite(1, HIGH);
  delayMicroseconds(2000);
  digitalWrite(1, LOW);

  delay(20);
}
```

19-3. The code contains **one mistake**. Do **not** compile this code. Instead, **identify** and then **correct** the mistake.

```
void loop() {
  digitalWrite(0, HIGH);
  delayMicroseconds(1000);
  digitalWrite(0, LOW);

  digitalWrite(1, HIGH);
  delayMicroseconds(2000);
  digitalWrite(1, LOW);
}
```

19-4. The code contains **one mistake**. Do **not** compile this code. Instead, **identify** and then **correct** the mistake.

```
void loop() {
  digitalWrite(0, HIGH);
  delayMicroseconds(1000);
  digitalWrite(0, LOW);

  digitalWrite(1, HIGH);
  delayMicroseconds(2000);
  digitalWrite(1, LOW);

  delay(2);
}
```

19-5. The code contains **one mistake**. Do **not** compile this code. Instead, **identify** and then **correct** the mistake.

```
void loop() {
  digitalWrite(0, HIGH);
  delayMicroseconds(1000);
  digitalWrite(0, LOW);

  digitalWrite(1, HIGH);
  delayMicroseconds(2000);
  digitalWrite(1, LOW);

  delayMicroseconds(20);
}
```

19-6. The code below is meant to control the motion of a robot whose wheels are connected to two servomotors. In keeping with the convention used throughout this chapter, the robot's left servo is connected to pin 0 and its right servo to pin 1. You may assume that the servo pins have been properly placed in **OUTPUT** mode. Fill in the blanks of the following code with numeric values that could be used to move the robot **forward** with **high speed**.

```
void loop() {
  digitalWrite(0, HIGH);
  delayMicroseconds( _____ );
  digitalWrite(0, LOW);

  digitalWrite(1, HIGH);
  delayMicroseconds( _____ );
  digitalWrite(1, LOW);

  delay ( _____ );
}
```

19-7. The code below is meant to control the motion of a robot whose wheels are connected to two servomotors. In keeping with the convention used throughout this chapter, the robot's left servo is connected to pin 0 and its right servo to pin 1. You may assume that the servo pins have been properly placed in **OUTPUT** mode. Fill in the blanks of the following code with numeric values that could be used to move the robot in **reverse** with **slow speed**.

```
void loop() {
  digitalWrite(0, HIGH);
  delayMicroseconds( _____ );
  digitalWrite(0, LOW);

  digitalWrite(1, HIGH);
  delayMicroseconds( _____ );
  digitalWrite(1, LOW);

  delay ( _____ );
}
```

- 19-8. The code below is meant to control the motion of a robot whose wheels are connected to two servomotors. In keeping with the convention used throughout this chapter, the robot's left servo is connected to pin 0 and its right servo to pin 1. You may assume that the servo pins have been properly placed in **OUTPUT** mode. Fill in the blanks of the following code with numeric values that could be used to **pivot** the robot about its center to the **left** at **high speed**.

```
void loop() {  
    digitalWrite(0, HIGH);  
    delayMicroseconds( _____ );  
    digitalWrite(0, LOW);  
  
    digitalWrite(1, HIGH);  
    delayMicroseconds( _____ );  
    digitalWrite(1, LOW);  
  
    delay ( _____ );  
}
```

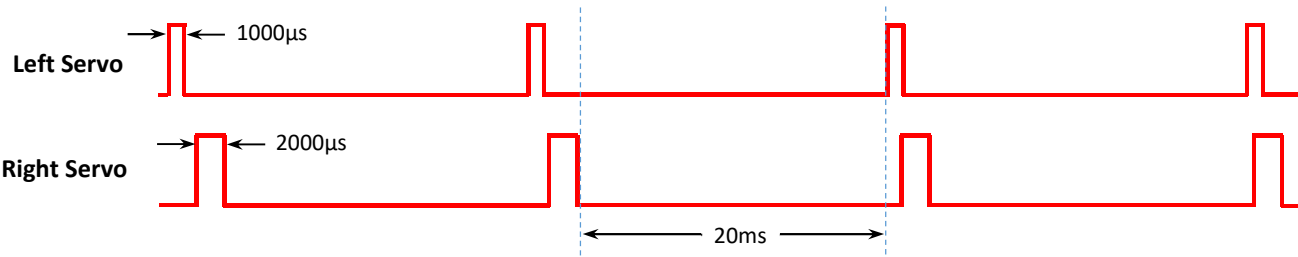
- 19-9. The code below is meant to control the motion of a robot whose wheels are connected to two servomotors. In keeping with the convention used throughout this chapter, the robot's left servo is connected to pin 0 and its right servo to pin 1. You may assume that the servo pins have been properly placed in **OUTPUT** mode. Fill in the blanks of the following code with numeric values that could be used to cause the robot to make a **wide, sweeping turn to the right**.

```
void loop() {  
    digitalWrite(0, HIGH);  
    delayMicroseconds( _____ );  
    digitalWrite(0, LOW);  
  
    digitalWrite(1, HIGH);  
    delayMicroseconds( _____ );  
    digitalWrite(1, LOW);  
  
    delay ( _____ );  
}
```

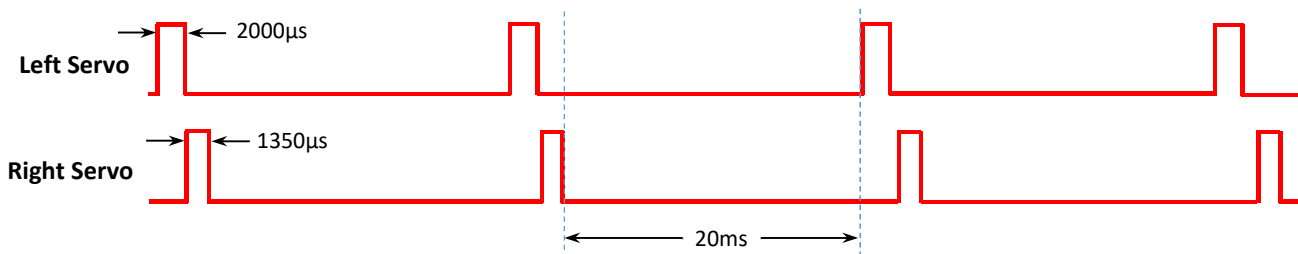
- 19-10. The code below is meant to control the motion of a robot whose wheels are connected to two servomotors. In keeping with the convention used throughout this chapter, the robot's left servo is connected to pin 0 and its right servo to pin 1. You may assume that the servo pins have been properly placed in **OUTPUT** mode. Fill in the blanks of the following code with numeric values that could be used to make the robot's wheels **stop spinning**.

```
void loop() {  
    digitalWrite(0, HIGH);  
    delayMicroseconds( _____ );  
    digitalWrite(0, LOW);  
  
    digitalWrite(1, HIGH);  
    delayMicroseconds( _____ );  
    digitalWrite(1, LOW);  
  
    delay ( _____ );  
}
```

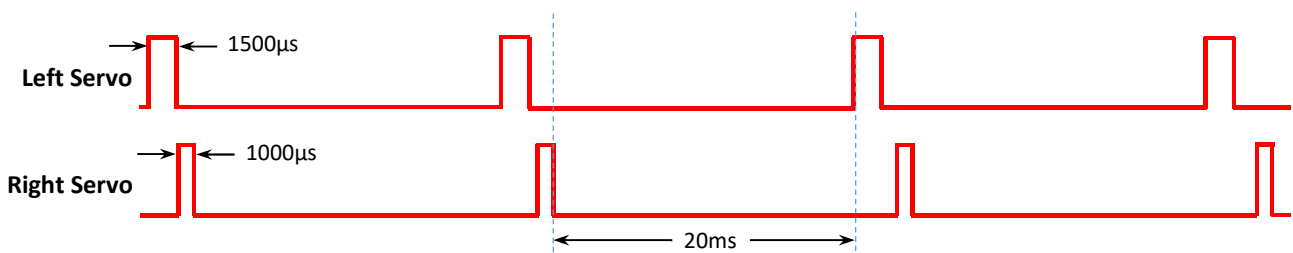
19-11. Imagine that the pulse trains pictured below are sent to *your* robot's left and right wheels. Examine the pulse widths and predict the speed direction would the robot move? Use descriptive words such as "forward", "reverse", "left", "right", "fast", "slow", "pivot", "spin", and "wide turn." Assume the robot is perfectly calibrated. In other words, 1000- μ s pulses rotate the spline just as quickly as 2000- μ s pulses, but in opposite directions, of course. Moreover, 1400- μ s pulses rotate the spline just as slowly as 1600- μ s pulses, again in opposite directions.



19-12. Imagine that the pulse trains pictured below are sent to *your* robot's left and right wheels. Examine the pulse widths and predict the speed direction would the robot move? Use descriptive words such as "forward", "reverse", "left", "right", "fast", "slow", "pivot", "spin", and "wide turn." Assume the robot is perfectly calibrated. In other words, 1000- μ s pulses rotate the spline just as quickly as 2000- μ s pulses, but in opposite directions, of course. Moreover, 1400- μ s pulses rotate the spline just as slowly as 1600- μ s pulses, again in opposite directions.



19-13. Imagine that the pulse trains pictured below are sent to *your* robot's left and right wheels. Examine the pulse widths and predict the speed direction would the robot move? Use descriptive words such as "forward", "reverse", "left", "right", "fast", "slow", "pivot", "spin", and "wide turn." Assume the robot is perfectly calibrated. In other words, 1000- μ s pulses rotate the spline just as quickly as 2000- μ s pulses, but in opposite directions, of course. Moreover, 1400- μ s pulses rotate the spline just as slowly as 1600- μ s pulses, again in opposite directions.



Turn in this sheet to be graded.