TEAM FRIENDSHIP

Sebastian Barry, Kevin Hawekotte, Matthew Teets

Tea Bag Dipper 9000™

Presentation Outline

1

Problem Statement

2

Functions and Features

3

Conclusion and Demo

Problem Statement

01 Arthritis / Carpal Tunnel

Carpal tunnel affects 4-10 million Americans. (https://rheumatology.org)

Its estimated that 69% of carpal tunnel syndrome related injuries are tied to dipping tea-bags (made up statistic)

02 General Sadness

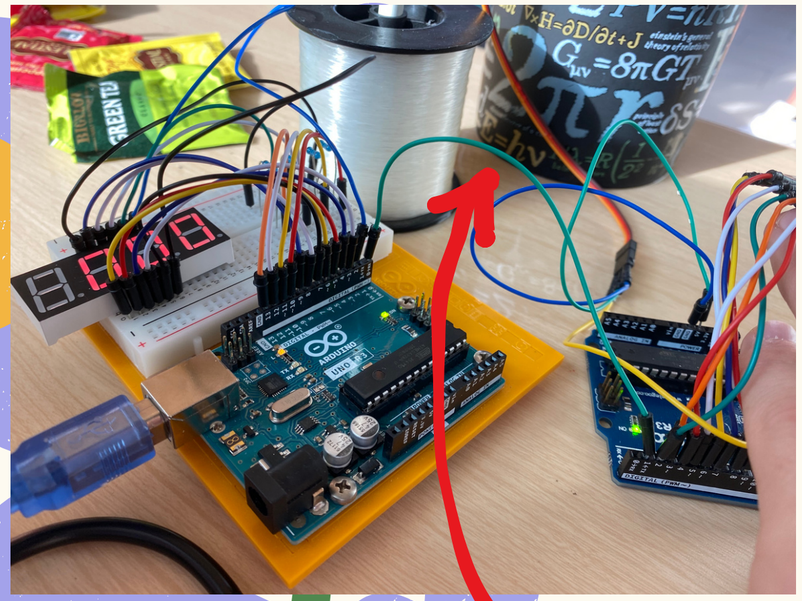
Have you, or a loved one been affected by tea-bag related sadness? Help is available now! Call 800-273-TALK

PARTS, FUNCTIONS, AND *Features*


Arduino | TEA BAG DIPPER 9000 □ | (NOT CLICKBAIT) | HD 1080p
Copy link
Watch on YouTube

AS SEEN ON TV

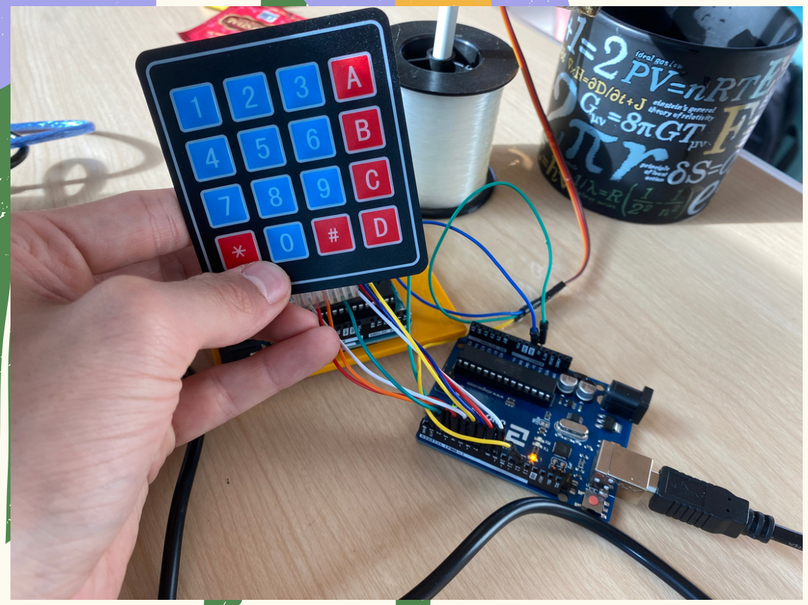
Parts Used



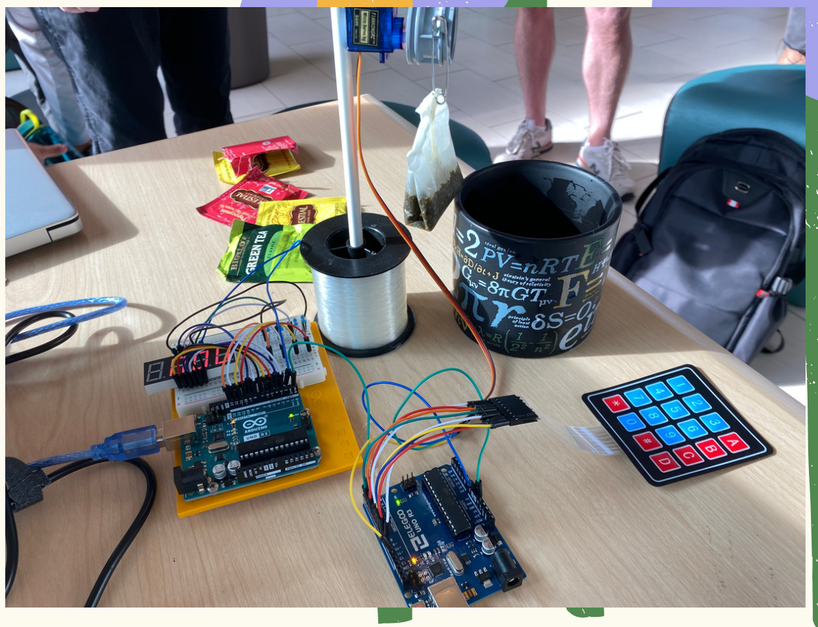
Rx/TX cable!

2x Arduino Uno

All parts from this kit!



Motor Arduino	Display Arduino
4x4 Matrix 16 Key Membrane Switch Keypad	4 Digit 7-Segment Display
Micro Servo Motor	



Arduino Code

How does it work?

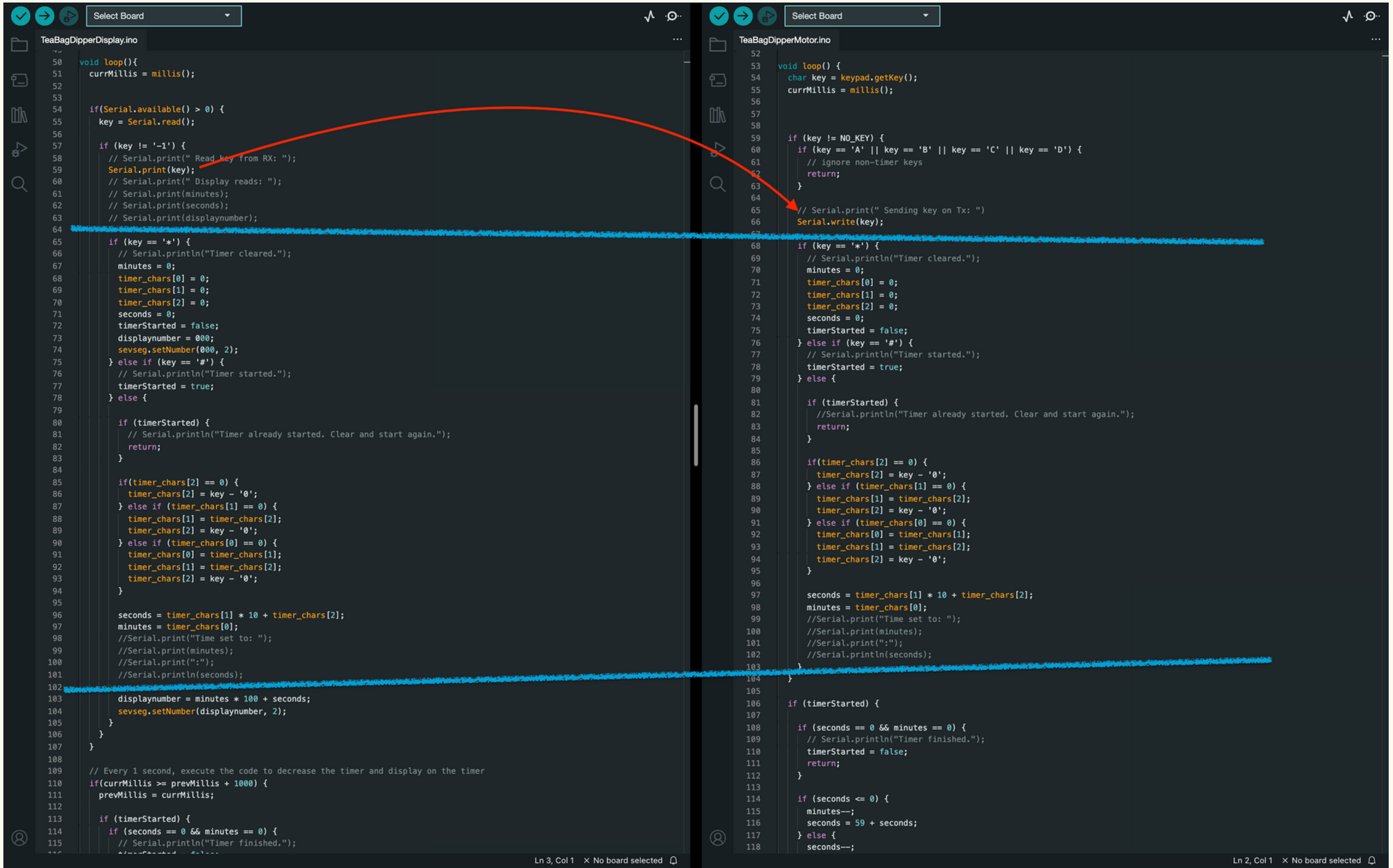
The Motor Arduino accepts the input and sends the character of the input to the Display Arduino using TX/RX

They both contain the same code to process the timer based on the inputs

Because they are communicating on the Serial port, any time we write/print to the Serial monitor, the message gets transferred to the other Arduino

Same code on both Arduinos

Inputted character is sent between both Arduinos



Same code on both Arduinos

One Arduino displays after the timer, one Arduino moves the motor after the timer

Notice the millis values for later (one runs every 1000 millis, the other runs every 5 millis for 180 times)

```
98          //Serial.print("Time set to: ");
99          //Serial.print(minutes);
100         //Serial.print(":");
101         //Serial.println(seconds);
102
103         displaynumber = minutes * 100 + seconds;
104         sevseg.setNumber(displaynumber, 2);
105       }
106     }
107   }
108
109   // Every 1 second, execute the code to decrease the timer and display on the timer
110   if(currMillis >= prevMillis + 1000) {
111     prevMillis = currMillis;
112
113     if (timerStarted) {
114       if (seconds == 0 && minutes == 0) {
115         // Serial.println("Timer finished.");
116         timerStarted = false;
117         return;
118       }
119
120       if (seconds <= 0) {
121         minutes--;
122         seconds = 59 + seconds;
123       } else {
124         seconds--;
125       }
126
127       //Serial.print("Time remaining: ");
128       //Serial.print(minutes);
129       //Serial.print(":");
130       //Serial.println(seconds);
131
132       displaynumber = minutes * 100 + seconds;
133       sevseg.setNumber(displaynumber, 2);
134     }
135   }
136   sevseg.refreshDisplay();
137 }
```

Display

Ln 3, Col 1 × No board selected

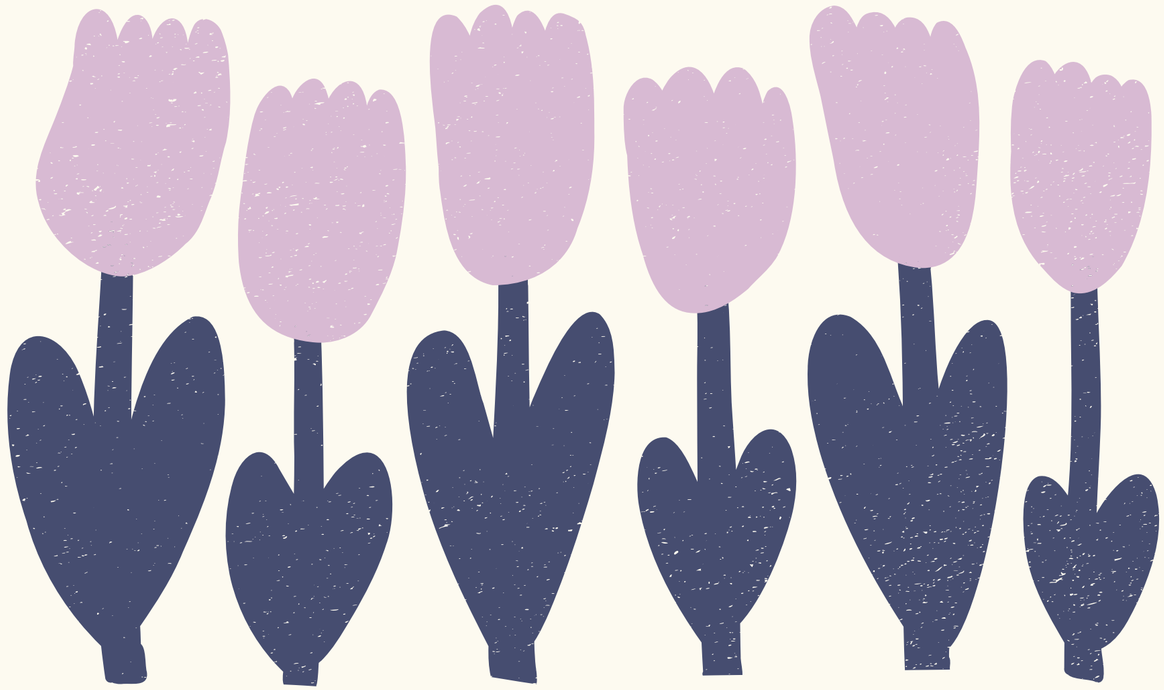
```
104   }
105
106   if (timerStarted) {
107
108     if (seconds == 0 && minutes == 0) {
109       // Serial.println("Timer finished.");
110       timerStarted = false;
111       return;
112     }
113
114     if (seconds <= 0) {
115       minutes--;
116       seconds = 59 + seconds;
117     } else {
118       seconds--;
119     }
120
121     //Move the servo every 5 milliseconds by 1 degree out of 180 degrees
122     if (currMillis >= prevMillis + 5) {
123       prevMillis = currMillis;
124       if(pos < 180) {
125         for (pos = 0; pos <= 180; pos += 1) { // Move the servo from 0 to 180 degrees
126           myservo.write(pos); // Set the servo position
127           delay(5); // Delay to allow the servo to move smoothly
128         }
129       } else if (pos > 0) {
130         for (pos = 180; pos >= 0; pos -= 1) { // Move the servo from 180 to 0 degrees
131           myservo.write(pos); // Set the servo position
132           delay(5); // Delay to allow the servo to move smoothly
133         }
134       }
135     }
136
137     //Serial.print("Time remaining: ");
138     //Serial.print(minutes);
139     //Serial.print(":");
140     //Serial.println(seconds);
141   }
142
143 }
```

Move the motor

Ln 2, Col 1 × No board selected

Challenges

What challenges did we face?



Timer is 100ms behind for every 1 second the timer runs.

Because we need to rotate the motor 180 degrees every second:

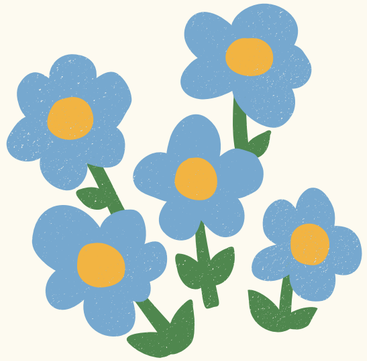
1 second / 180 degrees = 5.56 milliseconds/degree

But because the Arduino millis() function is only working in integer miliseconds, we have to use the value 5 milliseconds instead of 5.56 milliseconds, so the timer becomes more and more incorrect the longer the timer runs for.

The second time you run the Timer, it spins indefinitely. You must press the reset button on both Arduinos before resetting the timer.

We are not sure why this is happening, so for now it is a feature!

Conclusion and Demo



DEMO

Free Resource Page

Free Resource Page