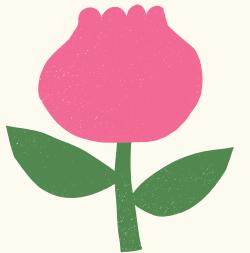


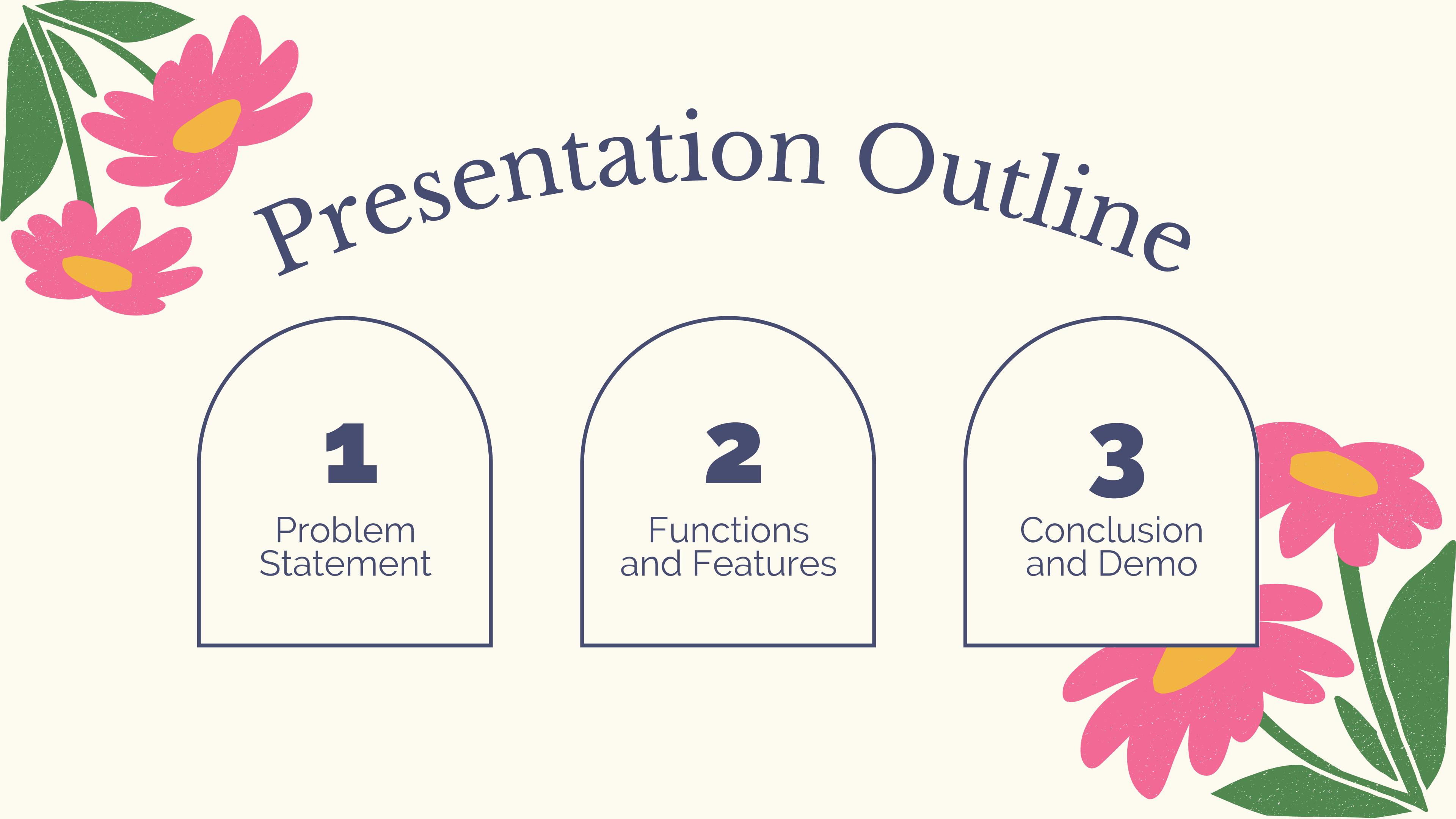
TEAM FRIENDSHIP



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)*





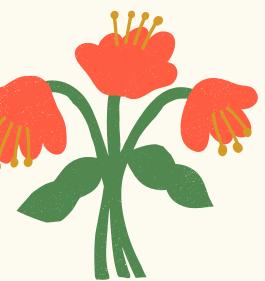
shutterstock.com · 503647327

02 General Sadness

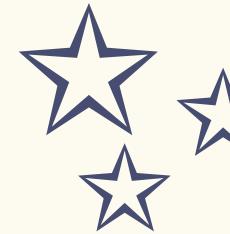
*Have you, or a loved one been
affected by tea-bag related sadness?
Help is available now!
Call 800-273-TALK*



PARTS, FUNCTIONS, AND



Features

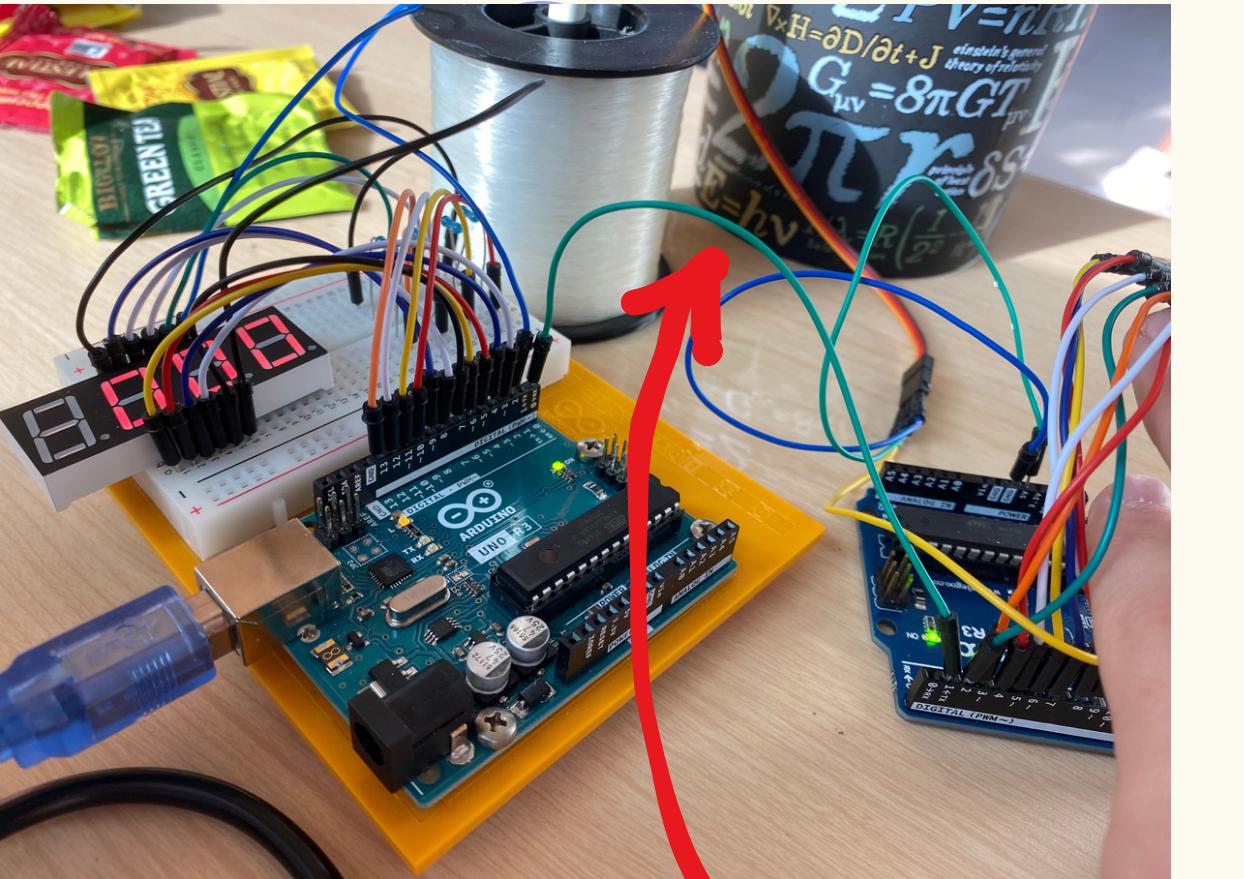




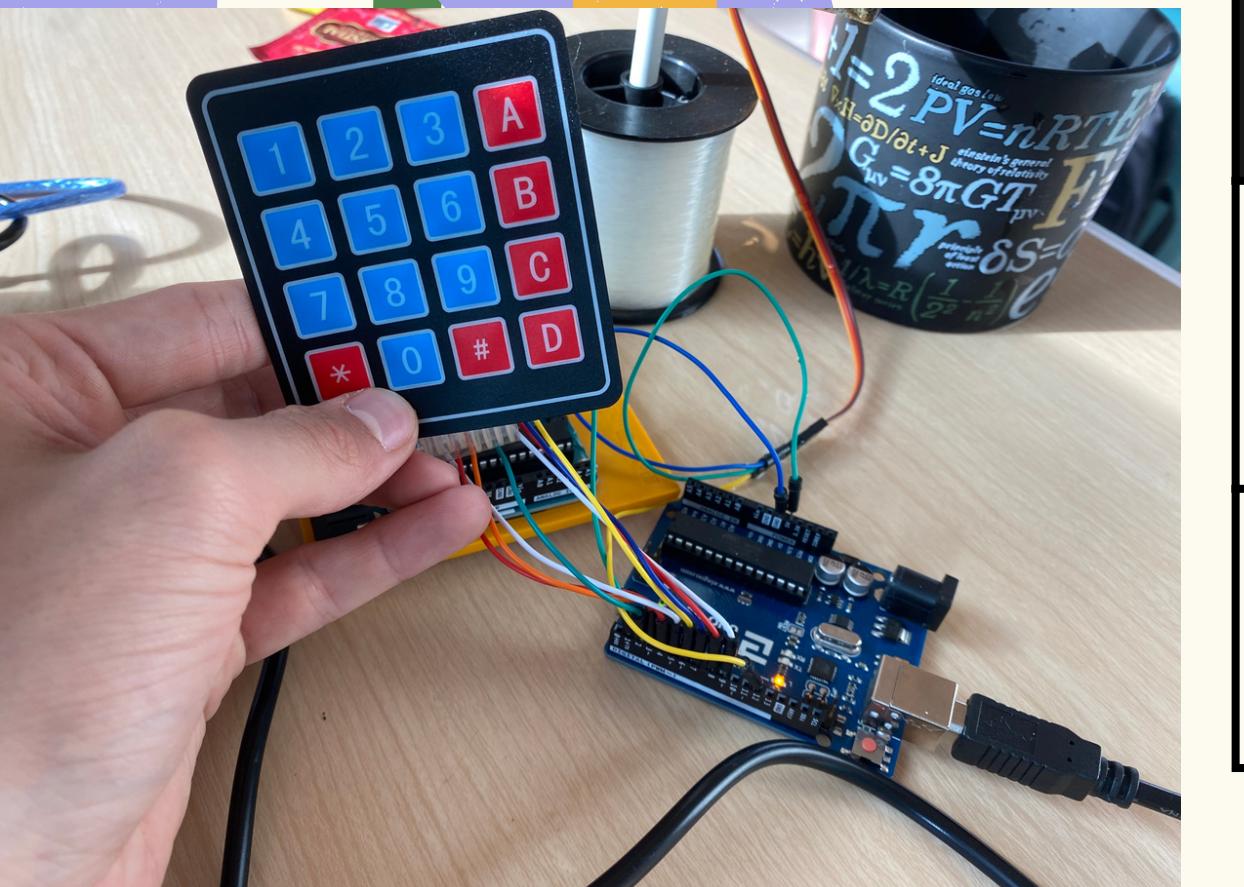
AS SEEN ON
TV

Parts Used

2x Arduino Uno



Rx/TX cable!



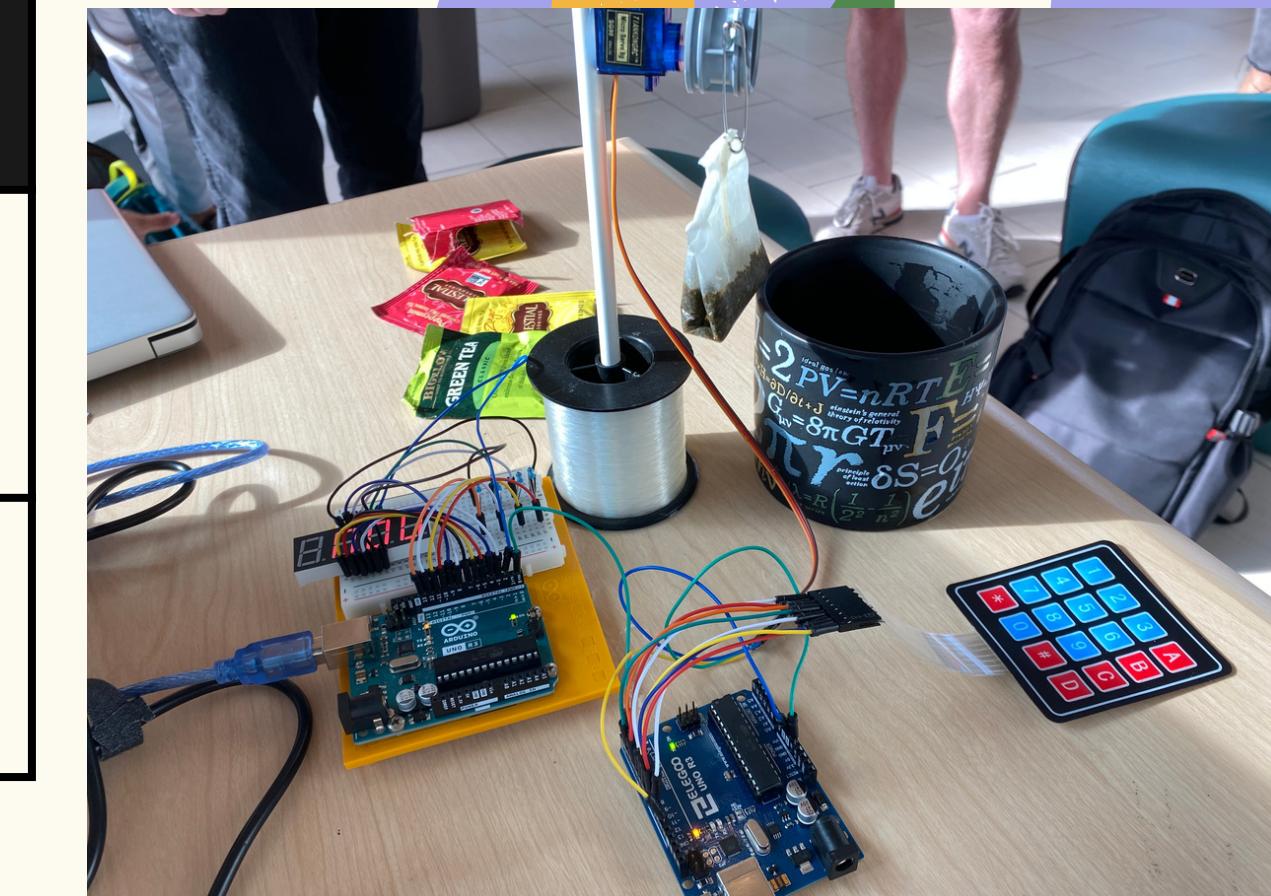
Motor Arduino

4x4 Matrix 16 Key
Membrane Switch
Keypad

Micro Servo Motor

Display Arduino

4 Digit 7-Segment
Display





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

```
TeaBagDipperDisplay.ino
50 void loop(){
51     currMillis = millis();
52
53     if(Serial.available() > 0) {
54         key = Serial.read();
55
56         if (key != '-1') {
57             // Serial.print(" Read key from RX: ");
58             Serial.print(key);
59             // Serial.print(" Display reads: ");
60             // Serial.print(minutes);
61             // Serial.print(seconds);
62             // Serial.print(displaynumber);
63
64             if (key == '*') {
65                 // Serial.println("Timer cleared.");
66                 minutes = 0;
67                 timer_chars[0] = 0;
68                 timer_chars[1] = 0;
69                 timer_chars[2] = 0;
70                 seconds = 0;
71                 timerStarted = false;
72                 displaynumber = 000;
73                 sevseg.setNumber(000, 2);
74             } else if (key == '#') {
75                 // Serial.println("Timer started.");
76                 timerStarted = true;
77             } else {
78
79                 if (timerStarted) {
80                     // Serial.println("Timer already started. Clear and start again.");
81                     return;
82                 }
83
84                 if(timer_chars[2] == 0) {
85                     timer_chars[2] = key - '0';
86                 } else if (timer_chars[1] == 0) {
87                     timer_chars[1] = timer_chars[2];
88                     timer_chars[2] = key - '0';
89                 } else if (timer_chars[0] == 0) {
90                     timer_chars[0] = timer_chars[1];
91                     timer_chars[1] = timer_chars[2];
92                     timer_chars[2] = key - '0';
93                 }
94
95                 seconds = timer_chars[1] * 10 + timer_chars[2];
96                 minutes = timer_chars[0];
97                 //Serial.print("Time set to: ");
98                 //Serial.print(minutes);
99                 //Serial.print(":");
100                //Serial.println(seconds);
101
102                displaynumber = minutes * 100 + seconds;
103                sevseg.setNumber(displaynumber, 2);
104            }
105        }
106    }
107
108    // Every 1 second, execute the code to decrease the timer and display on the timer
109    if(currMillis >= prevMillis + 1000) {
110        prevMillis = currMillis;
111
112        if (timerStarted) {
113            if (seconds == 0 && minutes == 0) {
114                // Serial.println("Timer finished.");
115            }
116        }
117    }
118}
```

```
TeaBagDipperMotor.ino
52
53 void loop() {
54     char key = keypad.getKey();
55     currMillis = millis();
56
57     if (key != NO_KEY) {
58         if (key == 'A' || key == 'B' || key == 'C' || key == 'D') {
59             // ignore non-timer keys
60             return;
61         }
62
63         // Serial.print(" Sending key on Tx: ");
64         Serial.write(key);
65
66         if (key == '*') {
67             // Serial.println("Timer cleared.");
68             minutes = 0;
69             timer_chars[0] = 0;
70             timer_chars[1] = 0;
71             timer_chars[2] = 0;
72             seconds = 0;
73             timerStarted = false;
74         } else if (key == '#') {
75             // Serial.println("Timer started.");
76             timerStarted = true;
77         } else {
78
79             if (timerStarted) {
80                 // Serial.println("Timer already started. Clear and start again.");
81                 return;
82             }
83
84             if(timer_chars[2] == 0) {
85                 timer_chars[2] = key - '0';
86             } else if (timer_chars[1] == 0) {
87                 timer_chars[1] = timer_chars[2];
88                 timer_chars[2] = key - '0';
89             } else if (timer_chars[0] == 0) {
90                 timer_chars[0] = timer_chars[1];
91                 timer_chars[1] = timer_chars[2];
92                 timer_chars[2] = key - '0';
93             }
94
95             seconds = timer_chars[1] * 10 + timer_chars[2];
96             minutes = timer_chars[0];
97             //Serial.print("Time set to: ");
98             //Serial.print(minutes);
99             //Serial.print(":");
100            //Serial.println(seconds);
101
102        }
103
104        if (timerStarted) {
105            if (seconds == 0 && minutes == 0) {
106                // Serial.println("Timer finished.");
107                timerStarted = false;
108                return;
109            }
110
111            if (seconds <= 0) {
112                minutes--;
113                seconds = 59 + seconds;
114            } else {
115                seconds--;
116            }
117        }
118    }
119}
```

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

```
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
125             if (pos < 180) {
126                 for (pos = 0; pos <= 180; pos += 1) { // Move the servo from 0 to 180 degrees
127                     myservo.write(pos); // Set the servo position
128                     delay(5); // Delay to allow the servo to move smoothly
129                 }
130             } else if (pos > 0) {
131                 for (pos = 180; pos >= 0; pos -= 1) { // Move the servo from 180 to 0 degrees
132                     myservo.write(pos); // Set the servo position
133                     delay(5); // Delay to allow the servo to move smoothly
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 3, Col 1 X No board selected

Ln 2, Col 1 X 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:
 $1 \text{ second} / 180 \text{ degrees} = 5.56 \text{ milliseconds/degree}$

But because the Arduino millis() function is only working in integer milliseconds, 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

