```
/*
Cello Arm Version 1 18 July 2016

*********************************************************
** ALWAYS RESET ARDUINO BEFORE POWERING UP STEPPER MOTOR **
*********************************************************

Does Forward stroke folled by reverse stroke after graphene trigger
Stroke takes 0.833248 seconds in each direction, total time 1.666667 seconds or once a bar at 144 beats / second
tempo
See spreadsheet for exact timing calculations

After forward and reverse stroke checks graphene signal again

    - If closed repeats forward and reverse stroke
    - If open Stops and waits for new ghaphine closure closure
*/

/*

** WIRING DETAILS **

GND - Black wire to microswitch common
GND - Purple wire to Graphene Contact
+5V - Orange wire to stepper controller OPTO
*/
const int direct = 7; // Grey  wire to stepper controller DIR
const int pulse = 4; // White wire to stepper controller PUL
const int led13 = 13;
const int microswitch = 2; // Yellow Wire to microswitch N.O.
const int Graphene_Start = 12; // Blue wire to Graphene Contact

int count = 0;
int ramp = 0;
```

```
int ramp_delay = 0;
int stroke_time = 352; // Stroke time = 0.833248 seconds


/*
Sets up I/O hand uses micro switch to find reference.
*/

void setup() {

  pinMode(led13, OUTPUT);
  pinMode(pulse, OUTPUT);
  pinMode(direct, OUTPUT);
  pinMode(Graphene_Start, INPUT); // Graphene Start Input
  digitalWrite(Graphene_Start, HIGH); // Turn on Pull Up
  pinMode(microswitch, INPUT); // Forward end stop
  digitalWrite(microswitch, HIGH); // Turn on Pull Up
  digitalWrite(direct, LOW); // low = backwards stroke, high  = forward stroke

  count = 1;
   digitalWrite(direct, HIGH);
    do {
        digitalWrite(led13, HIGH);   // turn the LED on (HIGH is the voltage level)
        digitalWrite(pulse, HIGH);  // start step
         delayMicroseconds(2500);            // wait for a 5m seconds
        digitalWrite(led13, LOW);    // turn the LED off by making the voltage LOW
        digitalWrite(pulse, LOW);  // stop step
         delayMicroseconds(2500);           // wait for a 5ms second
        count = digitalRead(microswitch);
    } while (count > 0);
    count = 0;
   digitalWrite(direct, LOW);
    do {
        digitalWrite(led13, HIGH);   // turn the LED on (HIGH is the voltage level)
```

```
      digitalWrite(pulse, HIGH);  // start step
       delayMicroseconds(640);              // wait for a n u seconds 320us needed
       digitalWrite(led13, LOW);    // turn the LED off by making the voltage LOW
       digitalWrite(pulse, LOW);  // stop step
       delayMicroseconds(640);              // wait for a 5ms second
       count ++;
   } while (count <1600);
   delay(2000);
}

// the loop function runs over and over again forever
void loop() {
    do {
       /*
        Wait For Graphene Start
       */
      } while (digitalRead(Graphene_Start) > 0);

   forward_stroke();
   backward_stroke();


}

void forward_stroke() {
  count = 0;
   digitalWrite(direct, HIGH);
   ramp_delay = 1000+ stroke_time;

   for (ramp = 0; ramp <100; ramp ++) {
       digitalWrite(led13, HIGH);   // turn the LED on (HIGH is the voltage level)
       digitalWrite(pulse, HIGH);  // start step
       delayMicroseconds(ramp_delay);             // wait for a n u seconds 320us needed
       digitalWrite(led13, LOW);    // turn the LED off by making the voltage LOW
```

```
      digitalWrite(pulse, LOW);  // stop step
      delayMicroseconds(ramp_delay);              // wait for a 5ms second
      ramp_delay = ramp_delay - 10;
      count ++;
    }

    do {
      digitalWrite(led13, HIGH);   // turn the LED on (HIGH is the voltage level)
      digitalWrite(pulse, HIGH);  // start step
      delayMicroseconds(stroke_time);              // wait for a n u seconds 320us needed
      digitalWrite(led13, LOW);    // turn the LED off by making the voltage LOW
      digitalWrite(pulse, LOW);  // stop step
      delayMicroseconds(stroke_time);              // wait for a 5ms second
      count ++;
    } while (count <1399);

    ramp_delay = stroke_time;
    for (ramp = 0; ramp <100; ramp ++) {
      digitalWrite(led13, HIGH);   // turn the LED on (HIGH is the voltage level)
      digitalWrite(pulse, HIGH);  // start step
      delayMicroseconds(ramp_delay);              // wait for a n u seconds 320us needed
      digitalWrite(led13, LOW);    // turn the LED off by making the voltage LOW
      digitalWrite(pulse, LOW);  // stop step
      delayMicroseconds(ramp_delay);              // wait for a 5ms second
      ramp_delay = ramp_delay + 10;
      count ++;
}
}
void backward_stroke() {


    count = 0;
    digitalWrite(direct, LOW);
    ramp_delay = 1000 + stroke_time;
```

```
  for (ramp = 0; ramp <100; ramp ++) {
     digitalWrite(led13, HIGH);   // turn the LED on (HIGH is the voltage level)
     digitalWrite(pulse, HIGH);  // start step
     delayMicroseconds(ramp_delay);             // wait for a n u seconds 320us needed
     digitalWrite(led13, LOW);    // turn the LED off by making the voltage LOW
     digitalWrite(pulse, LOW);  // stop step
     delayMicroseconds(ramp_delay);             // wait for a 5ms second
     ramp_delay = ramp_delay - 10;
     count ++;
  }

  do {
     digitalWrite(led13, HIGH);   // turn the LED on (HIGH is the voltage level)
     digitalWrite(pulse, HIGH);  // start step
     delayMicroseconds(stroke_time);             // wait for a n u seconds 320us needed
     digitalWrite(led13, LOW);    // turn the LED off by making the voltage LOW
     digitalWrite(pulse, LOW);  // stop step
     delayMicroseconds(stroke_time);             // wait for a 5ms second
     count ++;
  } while (count <1399);

  ramp_delay = stroke_time;
  for (ramp = 0; ramp <100; ramp ++) {
     digitalWrite(led13, HIGH);   // turn the LED on (HIGH is the voltage level)
     digitalWrite(pulse, HIGH);  // start step
     delayMicroseconds(ramp_delay);             // wait for a n u seconds 320us needed
     digitalWrite(led13, LOW);    // turn the LED off by making the voltage LOW
     digitalWrite(pulse, LOW);  // stop step
     delayMicroseconds(ramp_delay);             // wait for a 5ms second
     ramp_delay = ramp_delay + 10;
     count ++;
}
}
```