AUTOMATED THERMAL SAMPLE ACQUISITION WITH APPLICATIONS.

K. A. Kooshesh[1], D. H. Lineberger[2]. [1]Durham Academy 12kooshesh@da.org

Introduction: An automated system of data collection is very useful to the researcher, as it would be able to perform the exact same experiment more efficiently, consistently, and in a fashion that gathers many more data points than a human researcher could. We have created an Arduino®-based robot to detect the presence of samples subject to an experiment, perform certain measurements once each sample is located, and store the results for further analysis. More specifically, we designed the robot to perform an experiment on thermal inertia – measuring the changes in temperature over time of soil samples of various grain sizes – as thermal inertia has already been used to predict the size of materials on Mars [1]. Thus, we built the robot to remotely find samples arrayed in a circular path, measure the samples' surface temperature, then store that data in an organized fashion on a computer.

Methods: The robot consists of a vertical base, built from PVC pipe and bolted to a wooden platform, and a rotating arm, powered by a 1.8˚ precision step motor and attached to the top of the PVC pipe at a ninety-degree angle (Figure 1). The arm is able to rotate 270˚ (user-defined) on a horizontal plane parallel to the wooden platform. A Vex® ultrasonic range finder and a Mylexis® infrared temperature sensor attached to the free end of the rotating arm turn with the arm. The distance from these two sensors to the wooden platform was set at 25 cm.

The motion of the arm (facilitated by the motor) is controlled by an Arduino Duemilanove® (Figure 2), which is programmed to move the arm from its initial (rest) position, to detect samples, to take the temperature of a sample at its center, and to record that temperature data in the computer. As the arm works its way through this process and travels 270˚ from its original position, the microcontroller changes the direction of the rotation and moves the arm back to its original position. At that point, the arm hits a reset switch (a touch sensor) and rests for five minutes before it repeats the data collection process.

The specific algorithm used to move the robotic arm makes the motor step while the ultrasonic sensor continuously pings and monitors the distance from sensor to ground. When the ultrasonic sensor records a distance of approximately 25 cm, the sensor is pinging the wooden platform. But once the sensor detects something closer to it than 25 cm, the Arduino® assumes this is a sample, and the Arduino® starts counting the steps the motor takes until the arm reaches the other side of the sample (namely, when the ultrasonic sensor logs a distance greater than or equal to 25 cm). Knowing the number of steps that the motor took to move the arm from one end of the sample to the other end, the robot retraces half of the steps it took over the sample to position the temperature sensor over the center of the sample. Then, the robot takes temperature data using the Mylexis® probe and stores this in a computer data file under a specific sample number (the robot was programmed in this instance to find and store data on nine samples in a 270˚ arc, which it labels 1-9). Once it stores the temperature of the sample, the robot moves the arm to the end of the sample and resumes its pattern of sample searching. This process continues until the robot reaches the specified 270˚ maximum arc, whereby it changes the direction of the motor, rotates the arm until it reaches its rest position, hits the reset switch, and waits for five minutes before repeating data collection.

Figure 1: Assembled robot with main body, arm, motor, temperature sensor, and ultrasonic range finder

Results: Displayed below is some of the resulting code used to program the robot to search for samples while stepping, to recognize the samples, and to move the robot to the samples' centers, the most difficult parts of the project.

Stepping the Motor While Sample Searching:

```
double findNextSample() {
  int sampleStepCount = 0; // step count post sample
    detection
  int halfSampleStepCount = 0;
  int dist = measureDistance();
  while(dist > 25 && stepCount < MAX_STEPS )   {
     //while distance is greater than 25 cm, keep
       stepping
   myStepper.step(1);
   delay(50);
   dist = measureDistance();
   Serial.print( "Back from measureDistance dist = ");
   Serial.println( dist );
   stepCount++;
   Serial.println(dist);        }
  while( dist <= 25 && stepCount < MAX_STEPS ){
   myStepper.step(-1);
   delay(30);
   sampleStepCount++;
   stepCount++;
   dist = measureDistance();
   Serial.print( "Stepping through the sample: dist=");
    Serial.println( dist );      }
  if( stepCount >= MAX_STEPS )
   return 0; // reset
  delay(1000);
  halfSampleStepCount = sampleStepCount / 2;
  myStepper.step(halfSampleStepCount);
  delay(1000);
  double d = getTemp(); // Stores sample temp at
middle
  myStepper.step(-halfSampleStepCount);
  delay(1000);
  return d;                    }
```

Making the Ultrasonic Range Finder Ping:

```
int measureDistance()     {
  int cm;
  digitalWrite(outputPin, LOW);
  delayMicroseconds(2);
  digitalWrite(outputPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(outputPin, LOW);
  duration = pulseIn(inputPin, HIGH);
  cm = microsecondsToCentimeters(duration);
  delay(30);
  return cm;                 }
long microsecondsToCentimeters(long
microseconds) {
  // The speed of sound is 29 microseconds per
centimeter. Ping travels out and back, so to find the
distance of the object, we take half of the distance
travelled.
  return microseconds / 29 / 2;   }
```

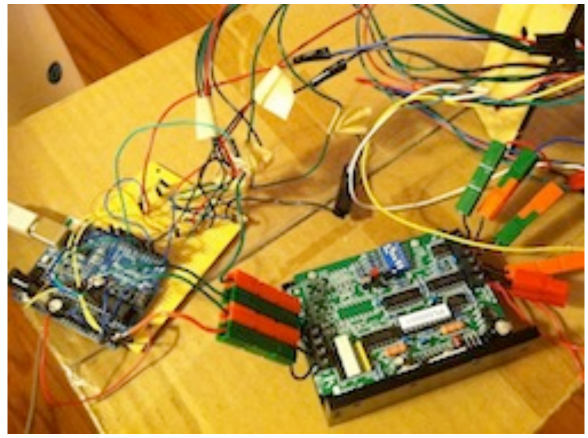

Figure 2: Robot circuit boards, including the Arduino® and Applied Motion Stepper Drive®

Discussion: The greatest challenge in getting the robot to function properly was that the 1.8˚ precision stepper motor demanded more power (12VDC) than the Digi-Key® IC Unit we had originally integrated could handle, thus frying the IC Unit without us knowing it. Thus, the robot would work in sporadic bursts, sometimes stepping back and forth, sometimes performing all the necessary sample-seeking duties. Once the problem was identified, an Applied Motion 2035® Stepper Drive (Figure 2) was installed in place of the IC Unit to boost the signal coming from the Arduino® to the motor.

Future Study: We would like to refine our data collection process with a robot that is more precisely built and can handle as many samples as one would like. For example, a sample-seeking robot similar to this could be built such that it follows a path of samples arrayed in rows and columns and takes its measurements while suspended on a rail framework from above.

Acknowledgements: We would like to thank the Mars Outreach for North Carolina Students (MONS) program for supporting us. Additionally, we would like to thank the Burroughs Wellcome Fund for funding. Finally, thank you to Robert Jackson, Carl Ward, and Fred Ward for the assistance provided in the early stages of the project.

References: [1] Ferguson, R. L and Christensen P. R. *40th Lunar and Planetary Science Conference (2009),* Abstract #1997.