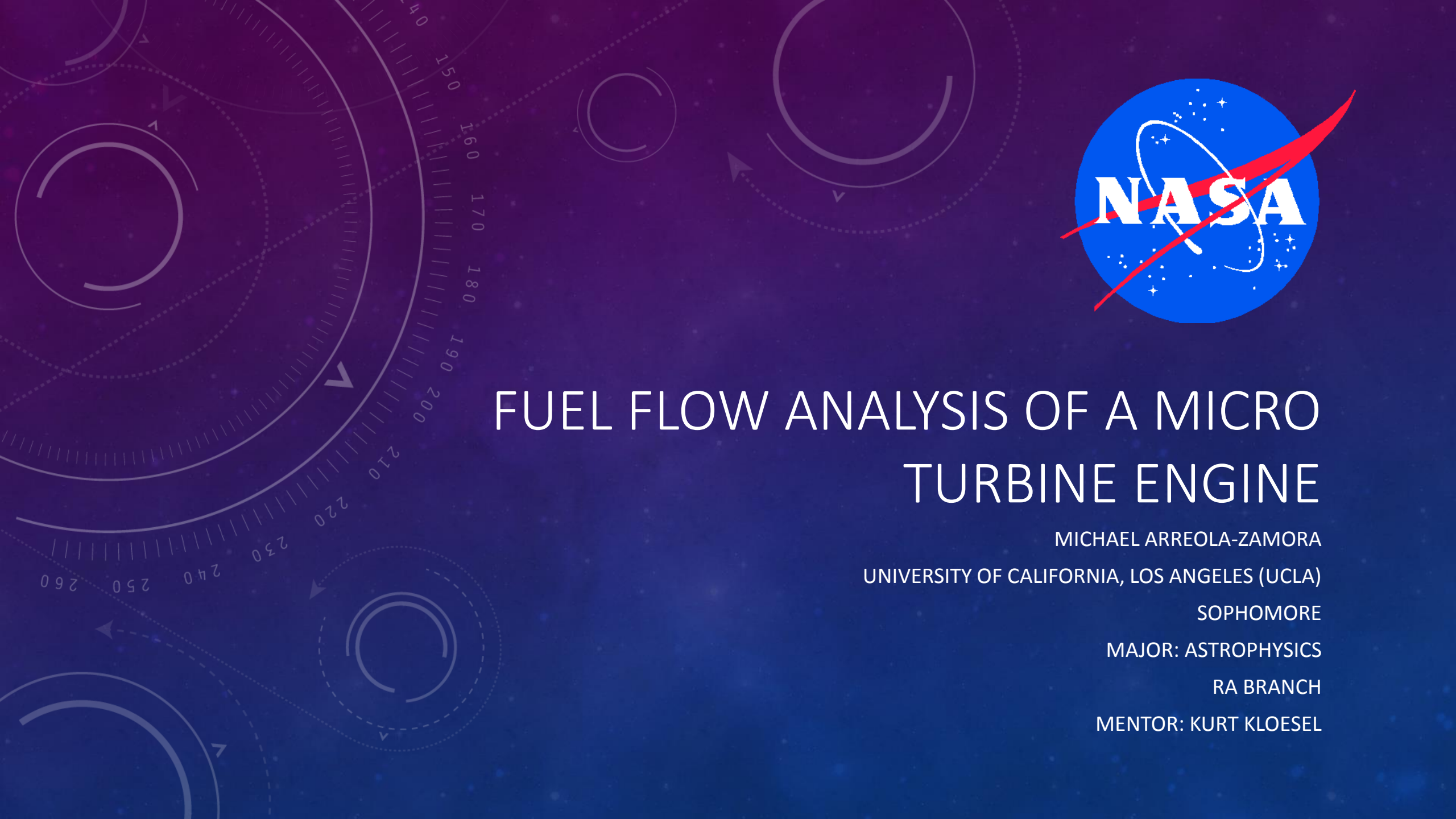


FUEL FLOW ANALYSIS OF A MICRO TURBINE ENGINE

MICHAEL ARREOLA-ZAMORA

UNIVERSITY OF CALIFORNIA, LOS ANGELES (UCLA)

SOPHOMORE

MAJOR: ASTROPHYSICS

RA BRANCH

MENTOR: KURT KLOESEL

OBJECTIVES

- Understand the processes a micro-turbine engine goes through
- Use micro-turbine engines to predict how bigger turbines would function
- Program an Arduino Uno to act as an engine control unit

MICRO TURBINE ENGINE

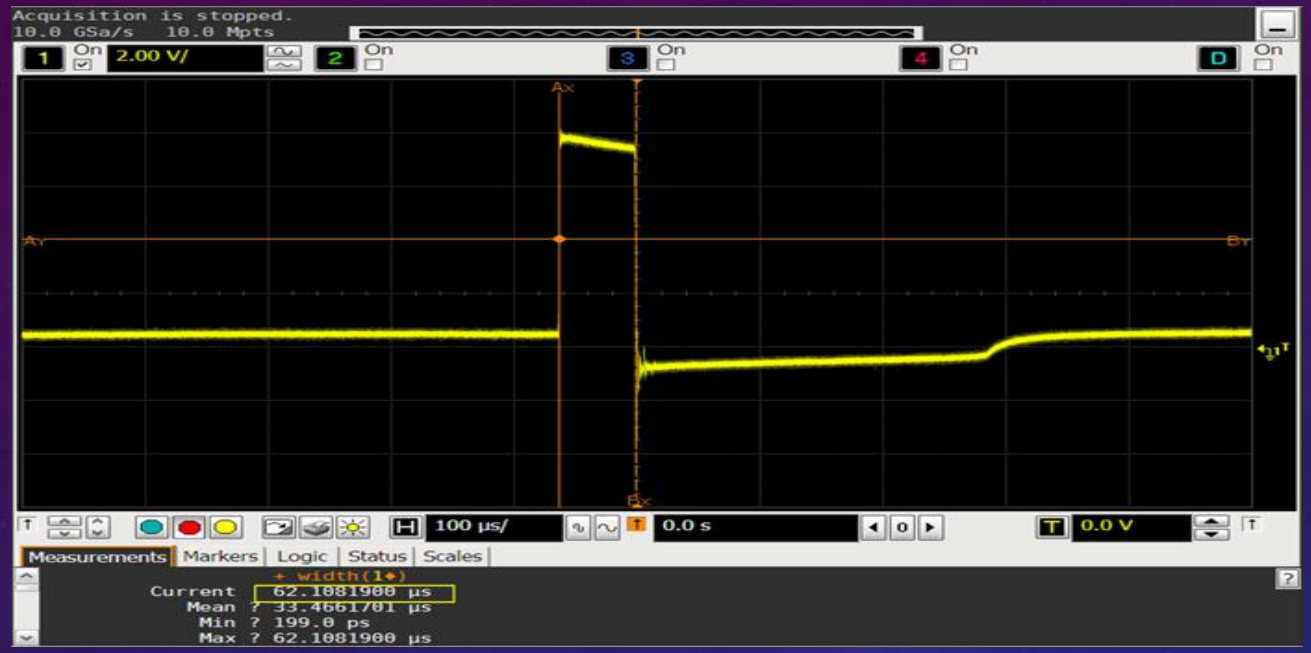
- Produces 15 pounds of thrust using propane and kerosene
- Relies on the fuel pump and engine control unit to begin running

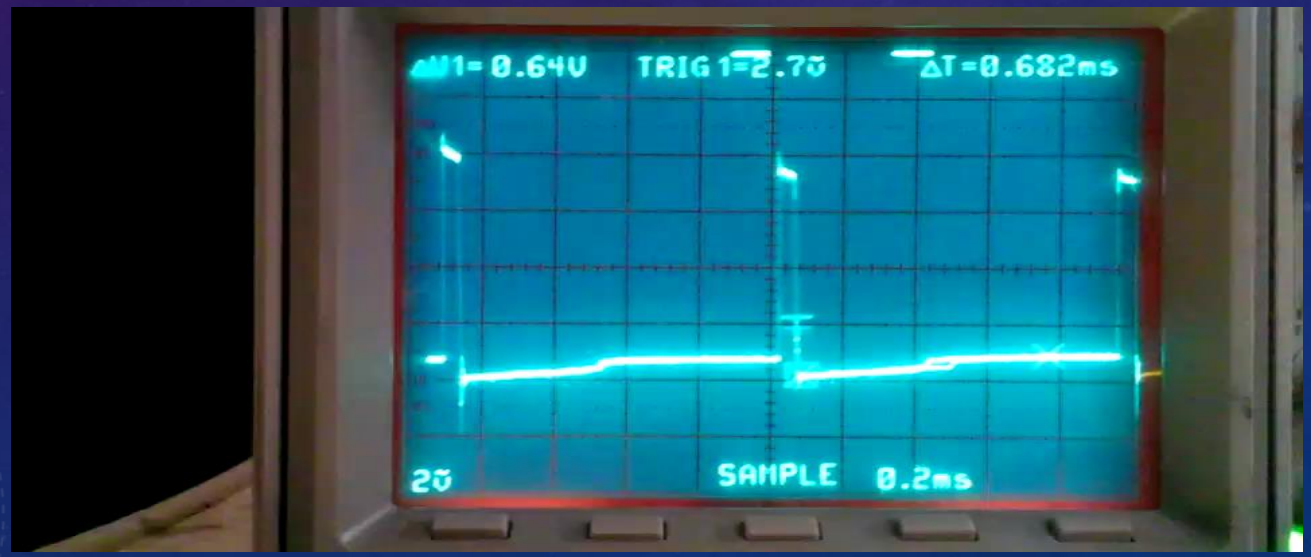
- Used to theorized operations of a larger turbine
- Able to achieve up to 120,000 RPMs

ESSENTIAL COMPONENTS OF THE TURBINE

- The engine control unit (ECU) is necessary for the operation of the turbine and receives inputs: RPM, temperature, and user throttle
- The engine control unit communicates with the fuel pump
- ECU informs the fuel pump and user if any errors occur (Ex. temperature too high)
- Monitors the amount of fuel being poured into the turbine

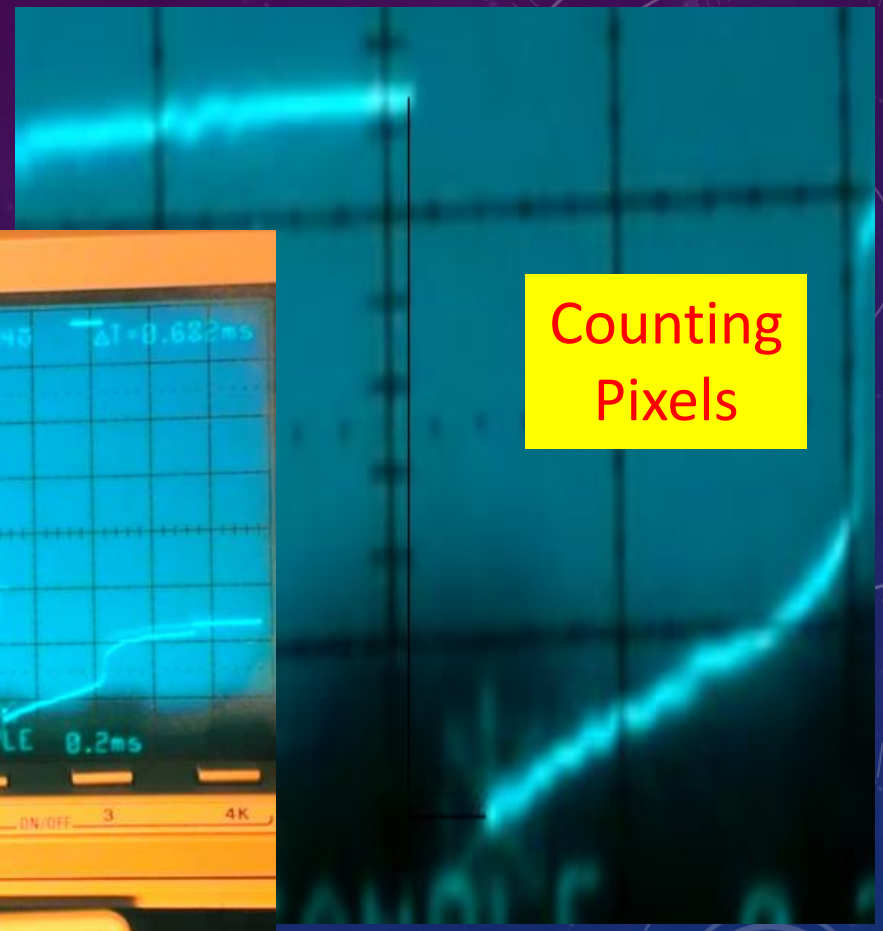
PULSE WIDTH





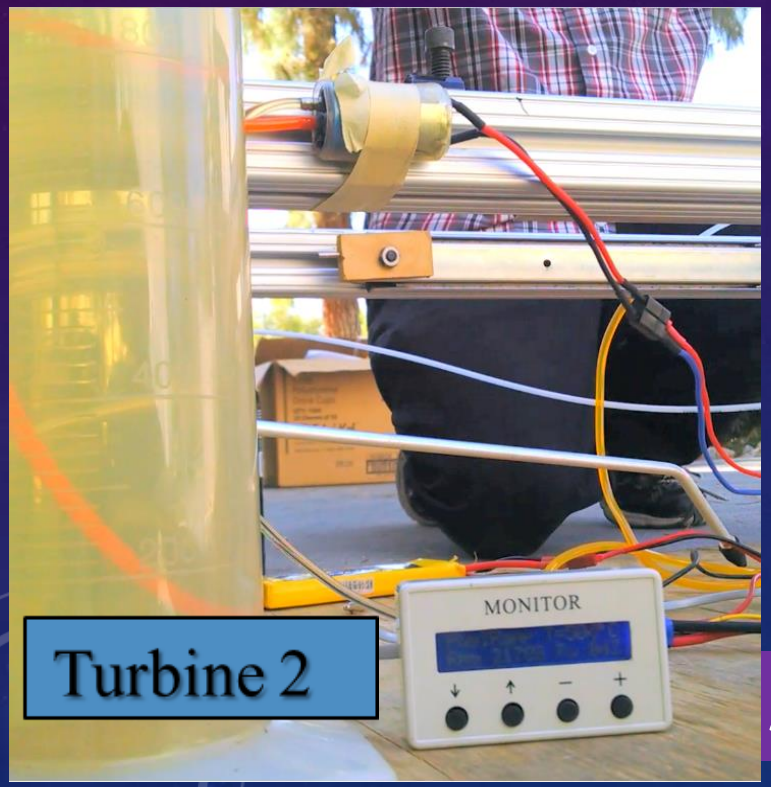
- What is pulse width?
  - ➢ Time (micro seconds) that the object runs at full power
- The highlighted number is the pulse width and will be approximately the same as the pulse width on the display monitor
- This also shows that the turbine will give a desired range of performance up to 8 volts
- The fuel pump has a pulse width measuring the amount of fuel being pumped into the turbine
- Knowing the pulse width is essential to understanding the turbine

Turbine 3



Counting Pixels

Turbine 1

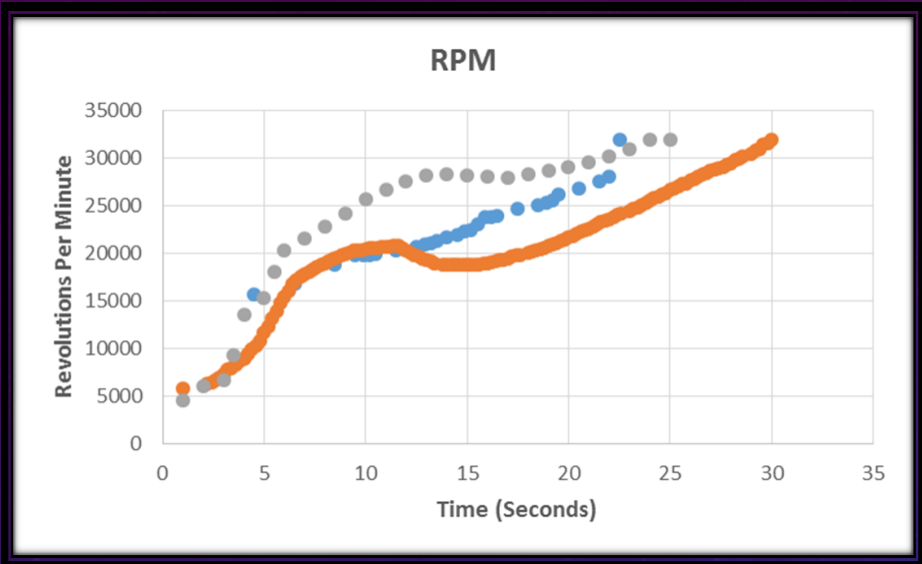


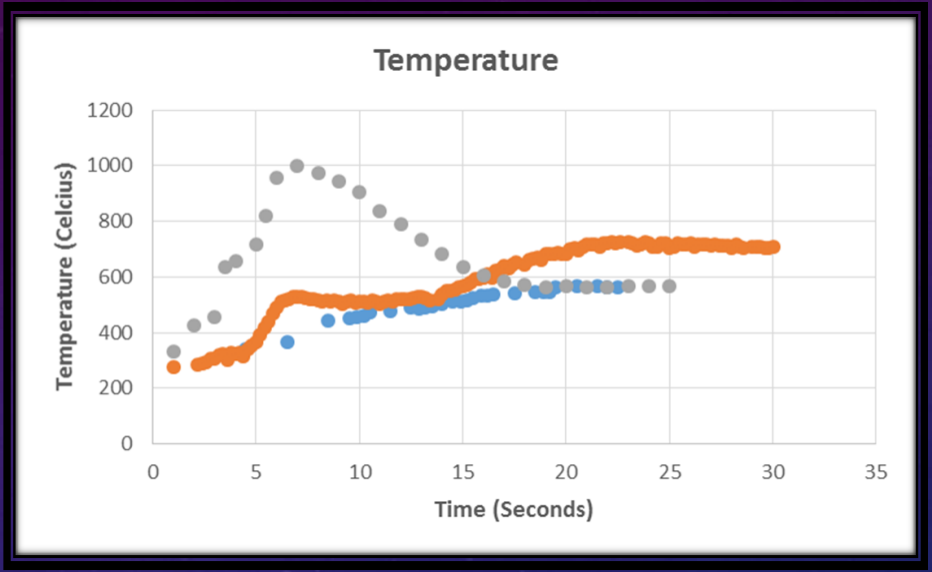
Turbine 2

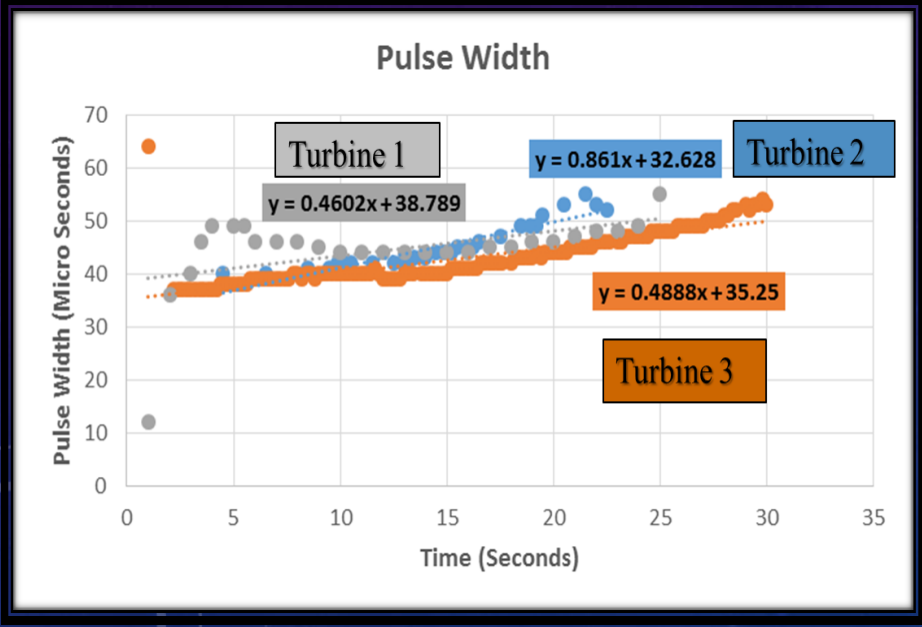
- Used frame-by-frame technology to gather data from each video
- Used Microsoft paint to enlarge images

Analyzing Fuel Pump Pulse Width In Multiple Turbine Runs

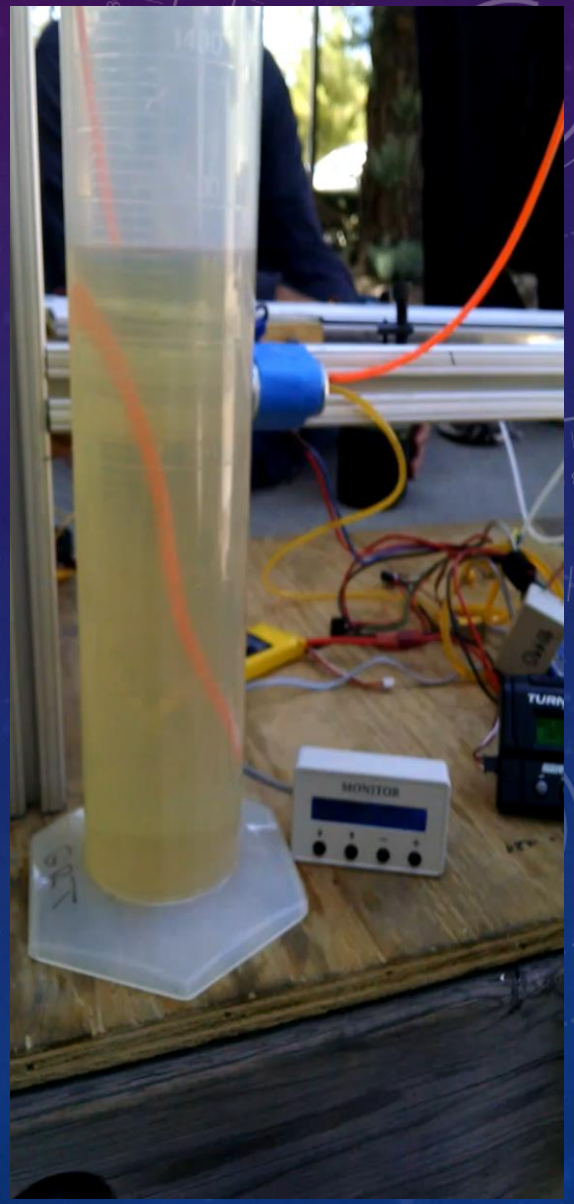
ANALYZED DATA







- Data was captured in three different runs with two different turbines (Same turbine in 2 & 3)
- Temperature becomes constant before the end of the fuel ramp
- In turbine 1, the temperature rose to 999°C which lowered the pulse width in order to stabilize
- Run times varied by a few seconds



Manipulation of a Micro Turbine Engine

RPM through a signal generator

Throttle Control

Temperature through a volt power source

Input

Input

Input

Engine Control Unit

Response

Output

Display Monitor

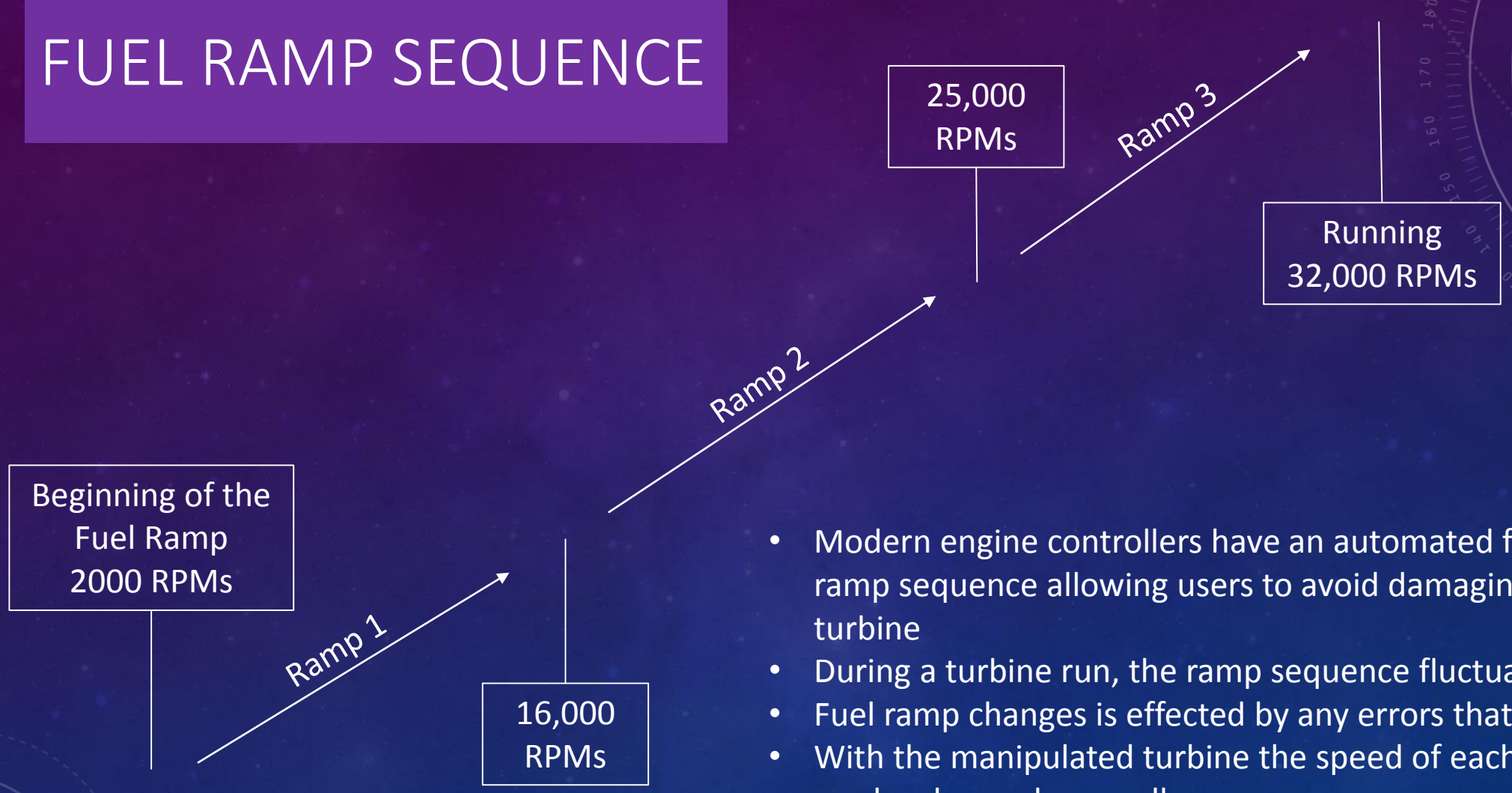
Output

Fuel Pump

Output

Oscilloscope measuring fuel pump pulse width

FUEL RAMP SEQUENCE

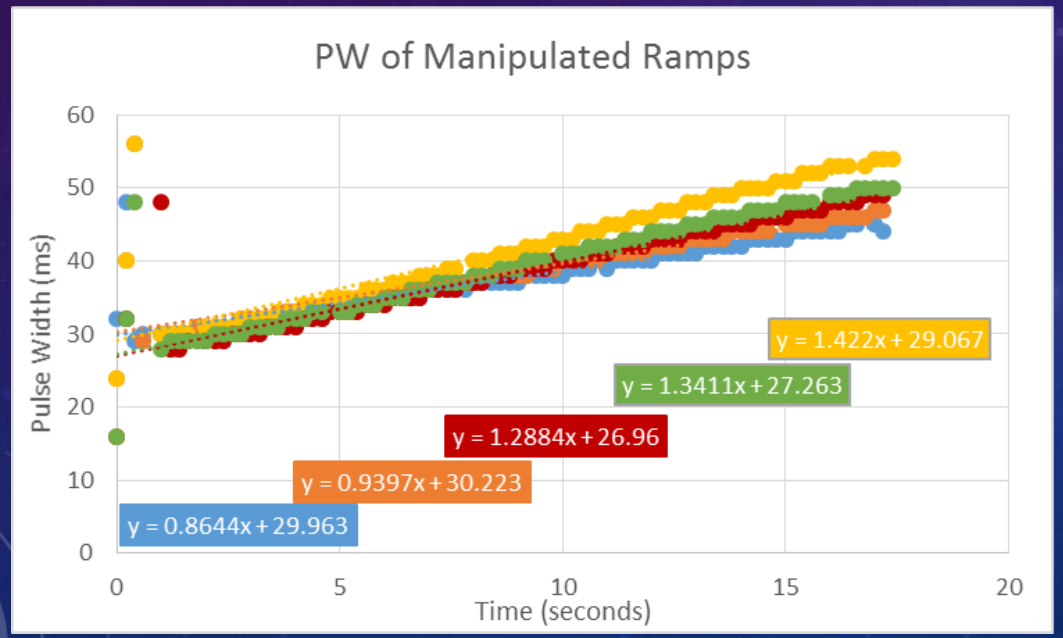


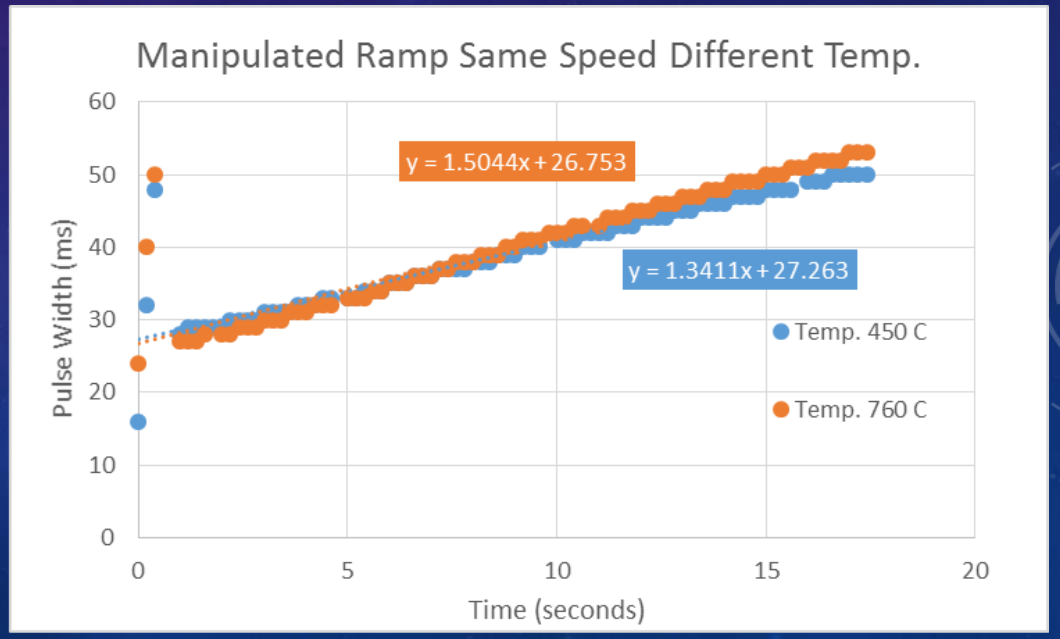
- Modern engine controllers have an automated fuel ramp sequence allowing users to avoid damaging the turbine
- During a turbine run, the ramp sequence fluctuates
- Fuel ramp changes is effected by any errors that occur
- With the manipulated turbine the speed of each ramp can be changed manually

SPEED CHANGE BETWEEN RAMP SEQUENCES

- Kept ramp 2 & 3 at a fixed speed, only changing ramp 1
- Want to know the difference between speeds
- Will reference this data when programing the Arduino

- Used same speed but with different temperatures
- Suggests that the temperature effects the pulse width
- Need more evidence to conclude this statement





ARDUINO UNO PULSE WIDTH

- Use the fading example to produce a square wave
- Generate a pulse width from the Arduino
- Do not directly hook up the Arduino to the motor [It will not work]

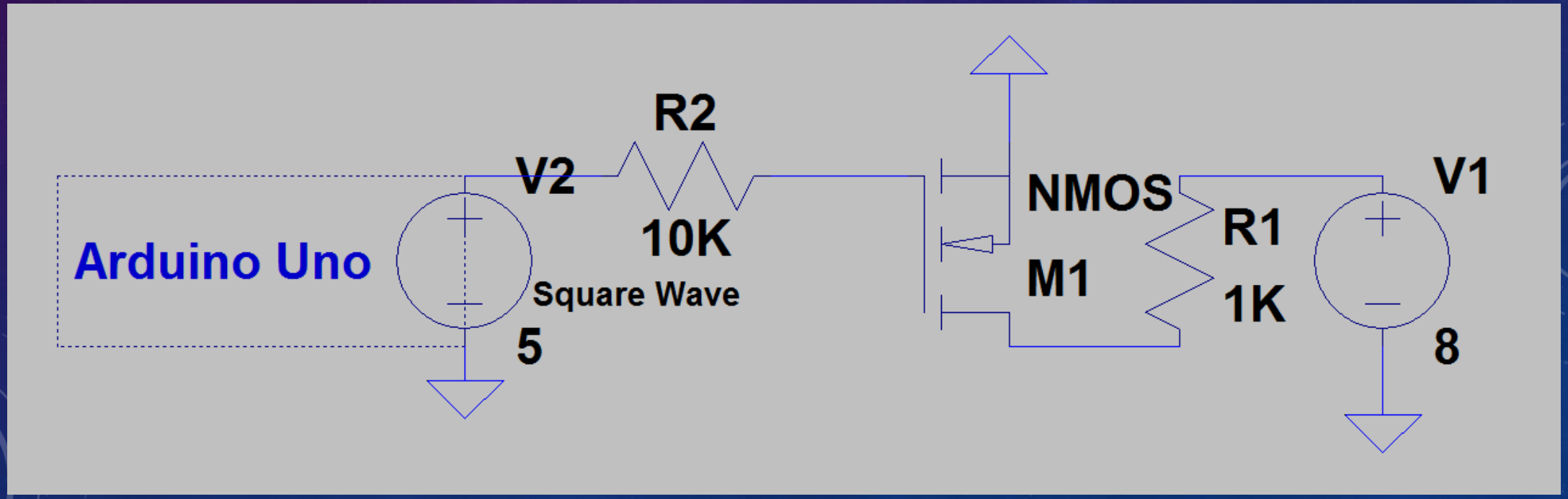
File Edit Sketch Tools Help

Fading

```
int ledPin = 9;    // LED connected to digital pin 9

void setup() {
  // nothing happens in setup
}

void loop() {
  // fade in from min to max in increments of 5 points:
  for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }

  // fade out from max to min in increments of 5 points:
  for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -= 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }
}
```

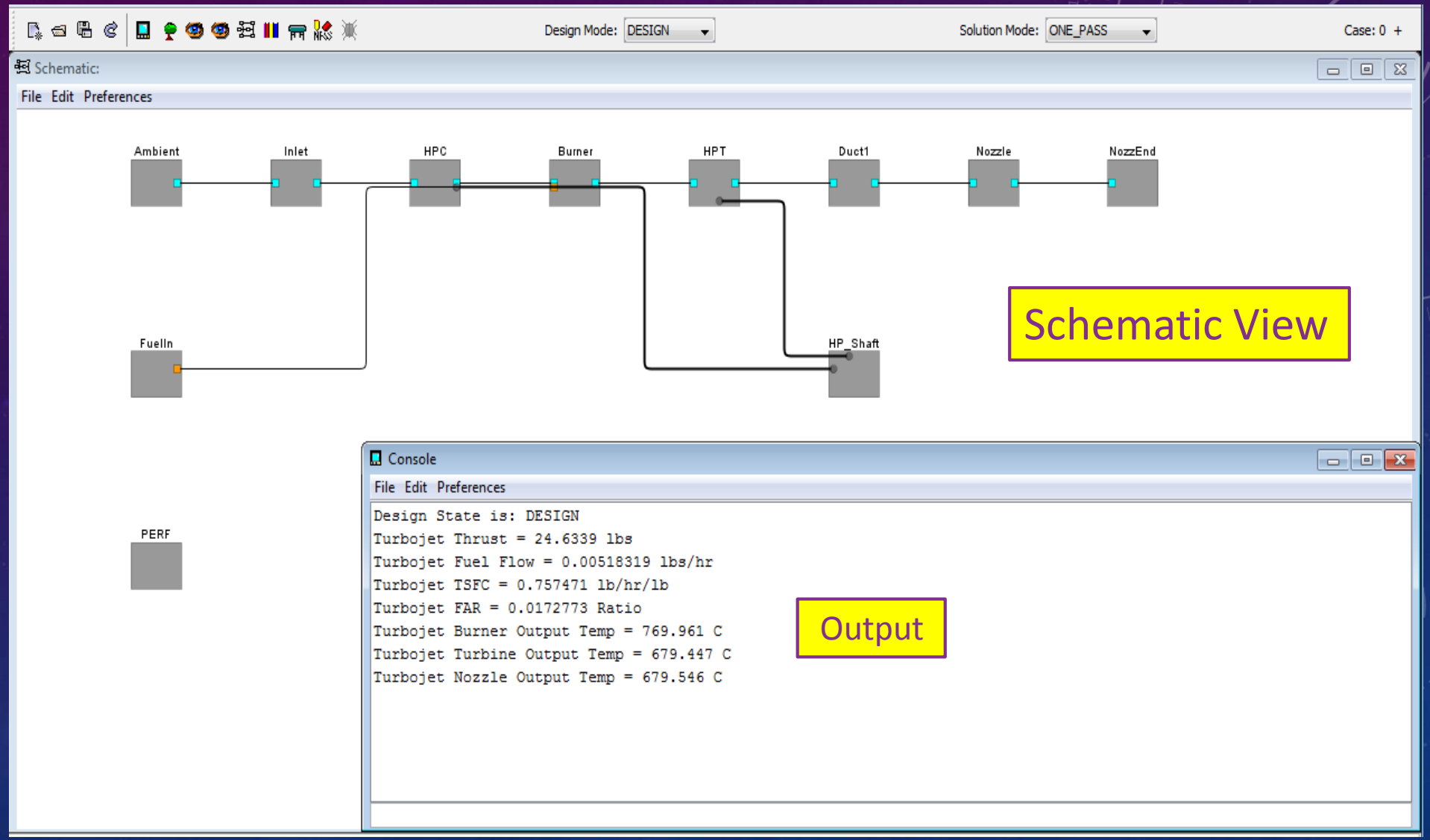
RUN THE MOTOR USING THE ARDUINO

- First step: Understand how the mosfet/transistor works (Use a power supply instead of Arduino)
- Second Step: Use a signal generator instead of Arduino to see how the square wave affects the transistor
- Third Step: Run the motor using the signal generator
- Fourth Step: Plug the Arduino in and see if it works like the signal generator (without motor)
- Final step: Run the motor with the Arduino Uno



NUMERICAL PROPULSION SYSTEM SIMULATION (NPSS)

- NPSS allows the user to model the turbine in depth
- Simulates a turbine to output specific values
- Uses a compressor map to simulate multiple runs at once
- Helps understand which inputs effect which outputs



FUTURE

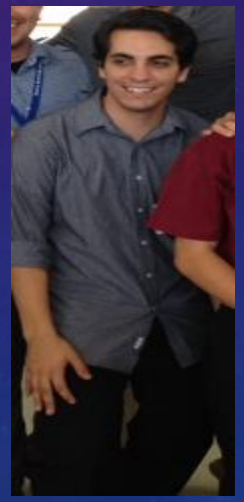
- With further investigation, this model can be used as a basis for future turbine engines in hybrid-electric airplanes
- The next step is to take this information from the micro turbine engines and apply it to a 65kW turbine
- The Arduino will allow for more research on the micro turbine engine at a cheaper cost
- This information may be transferred to aid other NASA projects (Ex. HEIST - Hybrid Electric Integrated System Test-bed)

WHAT I HAVE DONE AND LEARNED FROM THIS INTERNSHIP

If You Ain't Burnin'...
You Ain't Learnin'

ACKNOWLEDGEMENTS

Thanks to Kurt Kloesel, Michael Butros, Troy Kuhns, Thomas Pestolesi, Joseph Martinez, Kevin Collins, Christopher Bryan, Becky Flick, Aero Institute, all of code RA, and all of NASA Armstrong Flight Research Center.



K. A. R. A. T. E. SQUAD

QUESTIONS?