An Intelligent Air Compressor

Brian Rasnow, Benjamin Ramirez, Alexander Barrick
California State University, Channel Islands

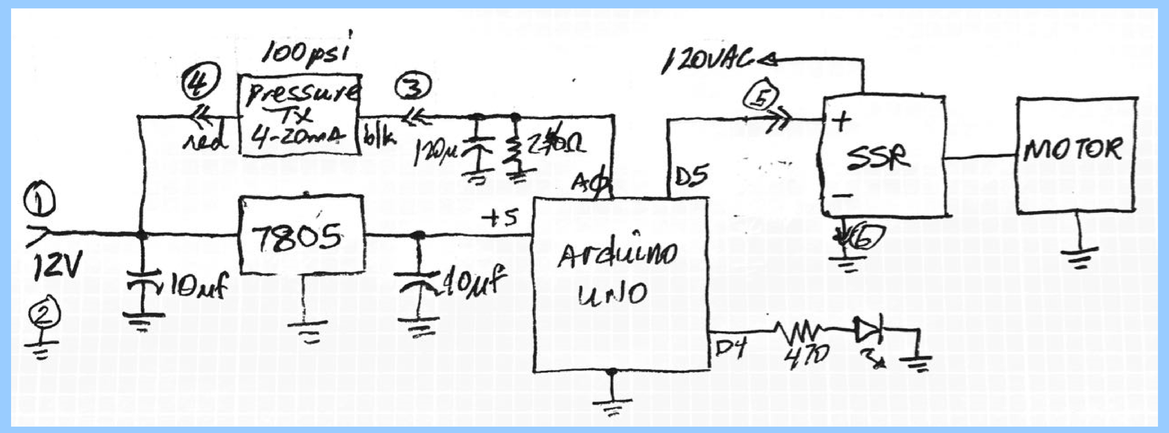
Project Background

Small air compressors are common tools many people infrequently use, and they are thus designed to minimize cost with little regard for their longevity. We present the rationale, design, and modifications to create a robust, intelligent, air compressor which operates with greater efficiency and extended longevity. By reducing the frequency of replacement of such common machines, we can reduce our contribution towards the residue layer of the human technosphere[1]. In this project we explore the use of simple computers to create an embedded system which manages air pressure. The main goal of the project is to increase longevity. Some other focuses of the project are portability, maintainability, and customization of system behavior. Further exploration of inexpensive microcontroller integration in frequently replaced machines can help reduce the waste output of the growing technosphere.

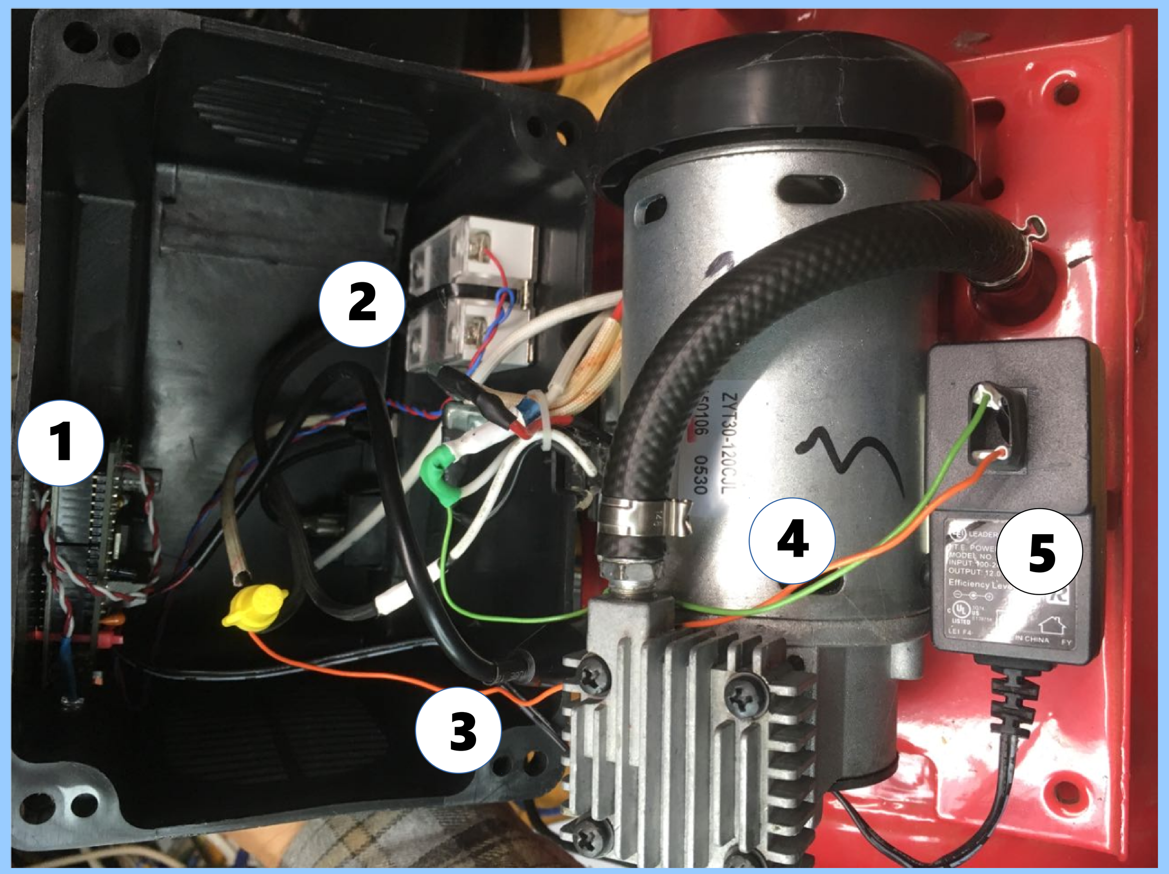
System Design

Our design replaces the digital pressure switch in a commercial air compressor with an analog pressure sensor interfaced to an Arduino programmable microcontroller. The microcontroller is programmed to control the compressor's motor based on both the instantaneous tank pressure and its history. The flexibility of setting on- and off-pressure values by software enable operation at lower pressures, thus saving energy and reducing wear-and-tear. A Watchdog timer ensures the motor will not run continuously (until it self-destructs) when/if the output lines rupture.

System design schematic



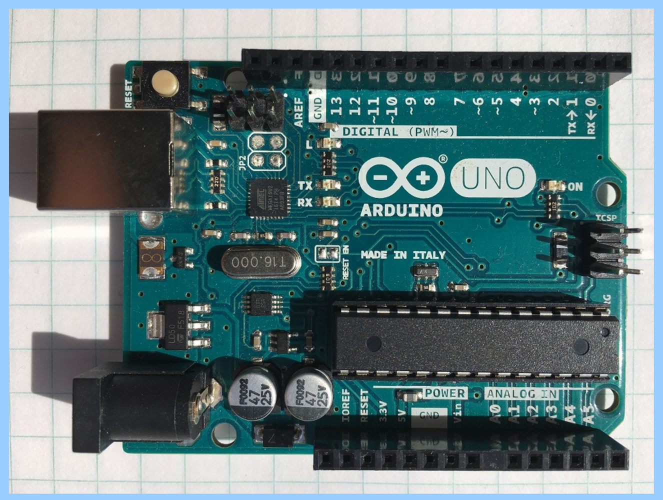
The constructed system



1) Arduino programmable microcontroller
2) Solid-state relay
3) Analog pressure sensor
4) Air compressor motor
5) 12V DC power supply

The Embedded System

We modified a 3 gallon 100 psi air compressor by replacing its 85PSI-on/100PSI off pressure switch with an analog pressure transducer, Arduino microcontroller, 12V DC power supply, solid state relay, heartbeat LED, and interface electronics as shown in the schematic (left).

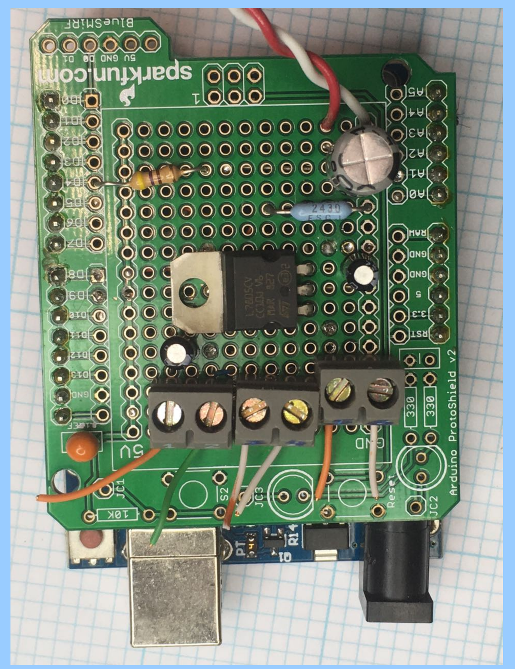


The Microcontroller

The Arduino UNO R3 is a $5 open source microcontroller with many accessible resources and libraries suitable for fast prototyping of embedded systems. With minimal operating and file systems, the Arduino has relatively low overhead and no OS crashes making it ideal for this project.

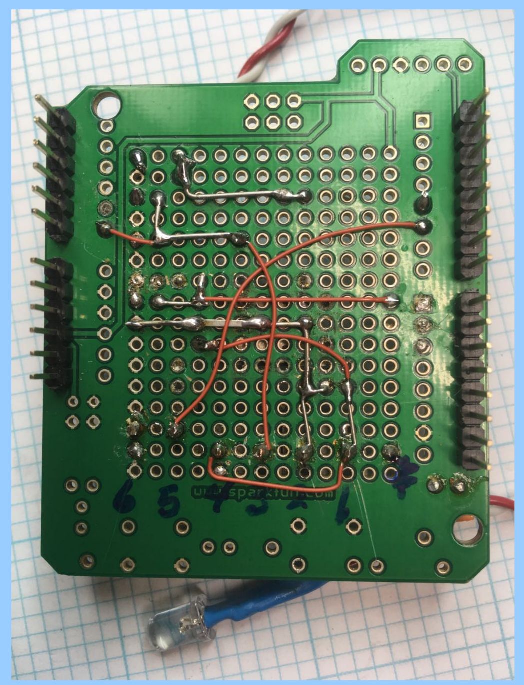
System Input

The analog pressure sensor converts a pressure range of 0-100 PSI to a linearly corresponding current from 4-20mA. A resistor and filter capacitor converts this signal to a low-noise voltage readable by the Arduino's analog input pin.



System Output

We added a solid-state relay (SSR) connected directly to an Arduino digital output pin (D5) which controls the state of the motor. A SSR is an electronic switching device that switches ports on or off when a small external voltage is applied across its control terminals. In this particular application, the SSR allows the 120V AC to power the motor when the digital output of the Arduino is high(5V) and turns the motor off when low(0V). A second digital output (D4) was connected to an LED providing a visual "heartbeat" indicating the state of the system. In normal operation, the LED beat at 1/2Hz, and switched to 2Hz if it detected a failure.



Daughter Board

The Arduino's 5V power supply and interface electronics were mounted on a daughter board or "Shield" constructed by point-to-point soldering.

Software implementation

The software on the microcontroller compares the pressure sensor output with constant setpoints, and updates the compressor motor's state accordingly. The Arduino is programmed with a subset of C/C++ functions[2]. An open-source IDE performs compilation, uploading, and serial monitoring (endnote: www.arduino.cc).

```
void loop() {
  timeSpent = millis() - startTime;
  psi = adu2psi(averageReading(100));

  if (MOTOR != killed) // heart_beat
    digitalWrite(LED_PIN, !digitalRead(LED_PIN));

  if (MOTOR == on && psi >= MAX_P)
    turnMotorOff();
  else if (MOTOR == on && timeSpent > MAX_ON_TIME)   //
    killMotor();
  else if (MOTOR == off && psi <  MIN_P)
    turnMotorOn();
  else if (MOTOR == killed && timeSpent > REBOOT_TIME)
    turnMotorOn();
  checkForSerialCmd();
  delay(READ_INTERVAL);
} // loop
```

The main control loop

Step 1:
Digitally filter and convert the pressure reading to PSI units.

Step 2:
Transition to the next state based on the current state of the motor and elapsed time in the current state.

Step3:
Poll the serial port for a command string, and parse/execute it.

```
int averageReading(int numTimes) {
  float sum = 0.0;
  for (int i = 0; i < numTimes; i++) {
    sum += analogRead(PRESSURE_PIN);
    delayMicroseconds(random(16667)); // average out 60Hz noise and harmonics
  }
  return (sum / numTimes);
} // averageReading
```

The input voltage from the pressure sensor is digitally filtered by averaging 100 successive readings. 60Hz noise can be substantial in proximity of an induction motor. Adding a time delay of a random fraction of 60Hz period ensures 60Hz noise will be at random phases and thus will average to zero.

```
int adu2psi(int reading) { //mapping the adu reading to PSI
  return map(reading, 172, 944, 0, 100);  // empirical params from pressure gauge
}
```

After filtering, the pressure is mapped from ADU to PSI units in order to increase code readability. We improve the signal reading quality by using both averaging as well as reducing noise caused by the 60Hz AC power of the motor.

Validation

In implementing this intelligent air compressor, we seek to discover the effectiveness of integrating a microcontroller and sensor into and existing air compressor with the goal of increasing the system's longevity. This was motivated by the need to reduce our contribution to the growing residue layer of the human technosphere. Finding ways to diverge from the common buy and throw away cycle is becoming more important. Fortunately, countries such as Sweden are beginning to propose tax breaks to people who choose to repair their old appliances instead of throwing them away[3]. Hopefully, other nations will also incentivize their citizens to increase the longevity of their possessions and buy high quality goods instead of short-lived disposable alternatives.

Literature Cited

[1] Zalasiewicz J et al. (2016) Scale and diversity of the physical technosphere: A geological perspective. The Anthropocene Review
[2] Language Reference. (n.d.). Retrieved April 14, 2017, from https://www.arduino.cc/en/Reference/HomePage
[3] Martin, R. (2016). Sweden Proposes Tax Breaks For Repairs. Retrieved April 12, 2017, from http://www.npr.org/2016/10/02/496282845/sweden-proposes-tax-breaks-for-repairs

Acknowledgements

Resources provided by California State University, Channel Islands and Professor Brian Rasnow