Alexander Shah
Project 2 – Temperature with Interrupts
2024-02-18
EN.605.715.81.SP24

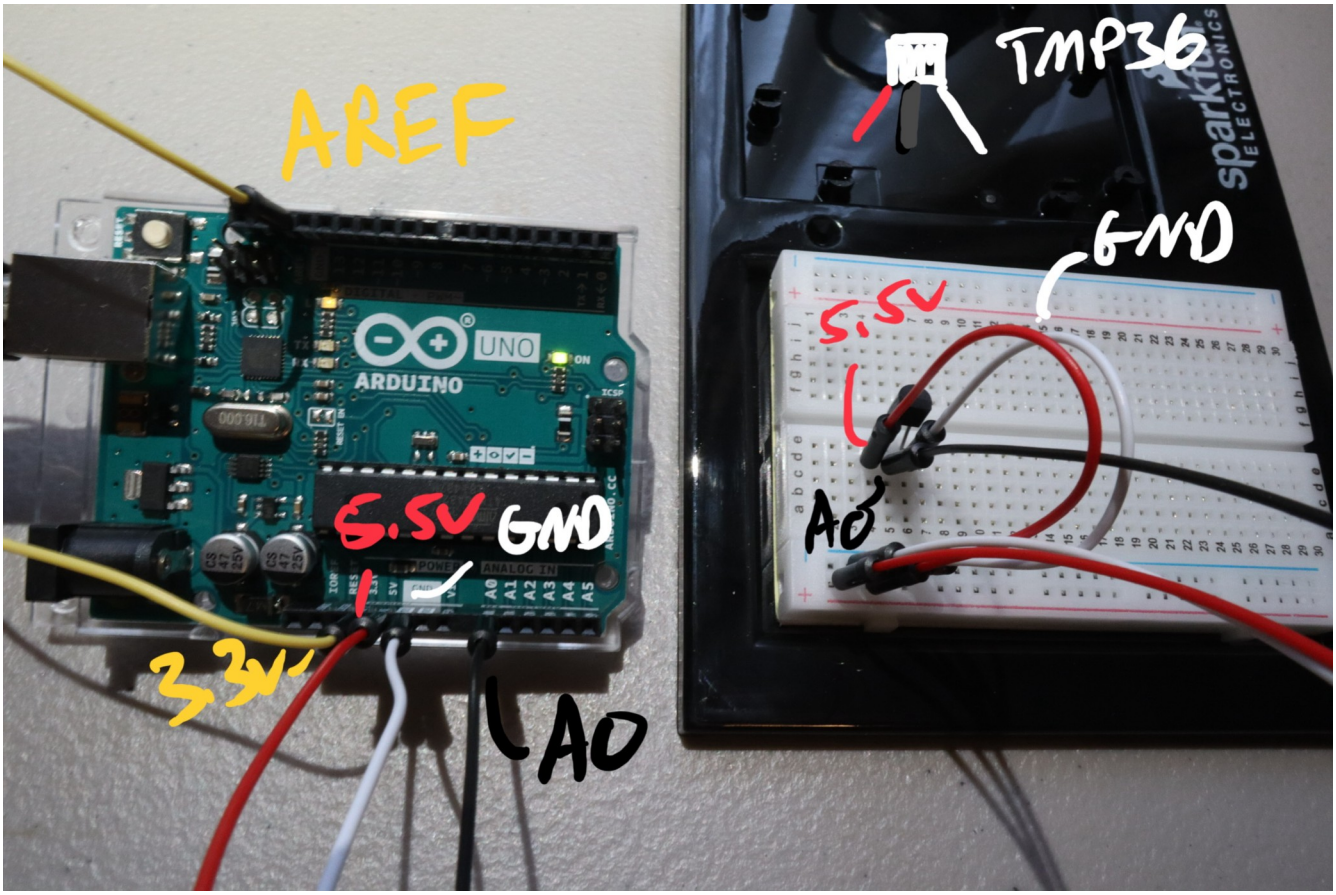
Table of Contents

Requirements

This project uses an Arduino UNO, a tmp36gz temperature sensor, and wiring to read from a temperature sensor in specific intervals, using interrupts to periodically poll the sensor during a round robin loop. This project was written and tested using the Arduino IDE in C to implement the Project 2 requirements as provided.

Design

The Arduino UNO connects via USB for power and serial output to the host. The TMP36 sensor has 3 pins and is inserted into the breadboard where the outer pins are connected to 5.5V and ground, and the middle pin is connected to the analog A0 pin on the arduino. A wire jumps 3.3V to AREF in order to use the reference voltage to improve the accuracy of the temperature sensor.



Implementation

The Project is written in C like code on Arduino IDE.

``` proj2-temp.ino

```
/*
Project 2 - Temperature Sensor with Interrupts
Alex Shah
2/18/24
*/

//using a tmp36gz on pin a0
int iPin = A0;
```
```

```
volatile float temperatureF = 0;
volatile bool newReadingAvailable = false;

void setup() {
  Serial.begin(9600);
  //use AREF port to enhance precision
  analogReference(EXTERNAL);
  attachInterrupt(digitalPinToInterrupt(2), readTemperature,
FALLING);
  cli(); //suspend interrupts
  //clear timers
  TCCR1A = 0;
  TCCR1B = 0;
  //set cs12, cs10 bits for prescalar
  TCCR1B |= (1<<CS12)| (0<<CS11) | (1<<CS10);
  //use compare match by setting OCIE1A to 1
  TIMSK1 |= B00000010;
  //set compare register A ~ 5-7 sec
  OCR1A = 15625;
  sei(); //resume interrupts
}

void loop() {
  if (newReadingAvailable) {
    Serial.println(temperatureF);
    newReadingAvailable = false;
  }
}

void readTemperature() {
```

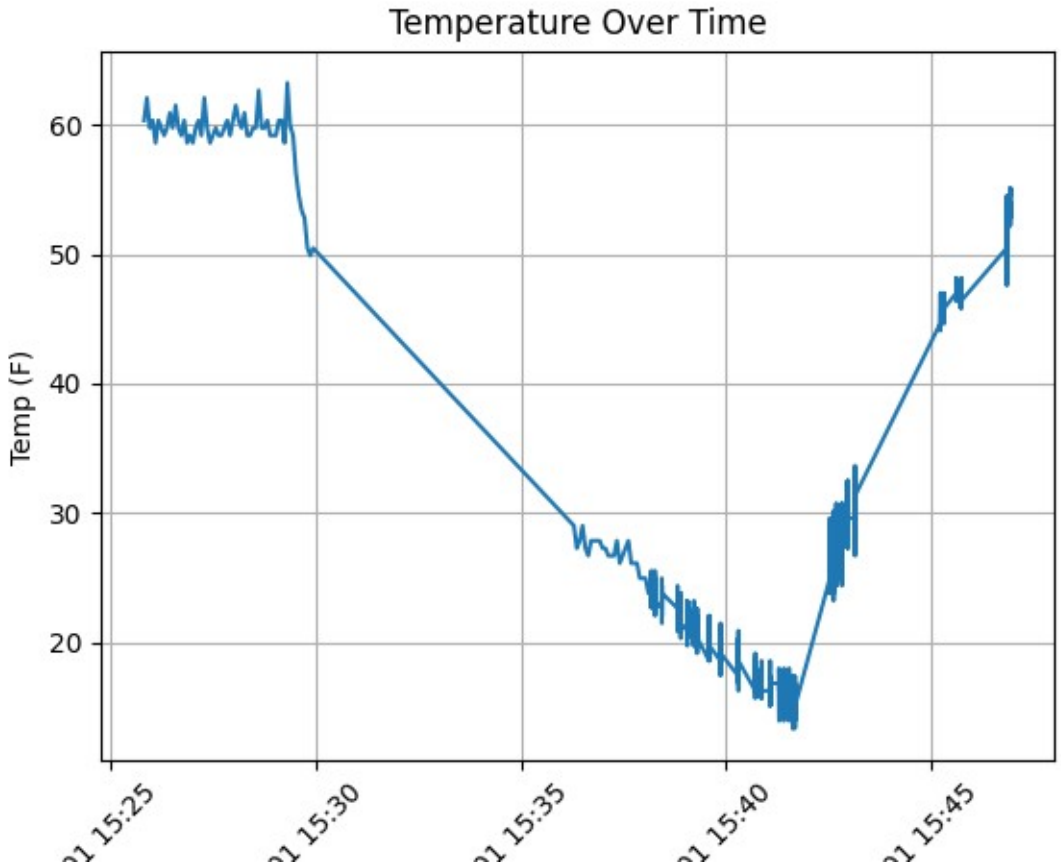
```
  int sensorValue = analogRead(iPin);
  //external reference voltage and resolution
  float fRef = 3.3;
  float fRes = 1024.0;
  // celsius = (sensor value * (voltage in mw / resolution) - offset)
* 100
  float fVolt = sensorValue * (fRef / fRes);
  float fTempC = (fVolt - 0.5) * 100;
  temperatureF = (fTempC * 9 / 5) + 32;
  newReadingAvailable = true;
}

ISR(TIMER1_COMPA_vect) {
  readTemperature();
}
```

Demo

Video showing working assembly:
https://drive.google.com/file/d/1oolzOVZK3Xycx-uyM3KyB1P5aNfiinlL/view?usp=sharing

Figure 1 shows capturing data in a warm room, putting the arduino in the fridge, some time elapses in the fridge (skipped), then sticking the arduino in the freezer, then pulling out the assembly and letting it warm up. (Note, the clock seemed to freeze too and began outputting much shorter intervals than when it was warm, an interesting example of environment affecting data)



References

https://electronoobs.com/eng_arduino_tut140.php