**HW1 Calibrated Temperature**

Johns Hopkins University

Real Time Software for Embedded Systems

Fall 2014

Tony Florida

2014-09-16

**Requirements**

Hardware

* The temperature sensor shall be connected to an Arduino microcontroller circuit

Software

* The software running on the Arduino shall use a Round Robin design
* The software shall capture the temperature to 1/10th of a degree
* The software shall wait to record temperature until the temperature has stabilized
* The software shall record temperature at a rate of 10 seconds

Test

* The test should begin at room temperature for 5 minutes, then record in a refrigerator for 5 minutes, then record room temperature for 5 minutes
* The deliverable should be a plot of temperature vs time

Parts List

* (1) Arduino Uno
* (2) 2.2k resistors
* (1) DS18B20 Temperature Sensor
* (5) hobby wires of length 3” or greater
* (1) USB 2.0 A/B cable
* (1) breadboard

Required Software

* Arduino Sketch v1.0
* Microsoft Excel 2010

**Architecture**

Hardware

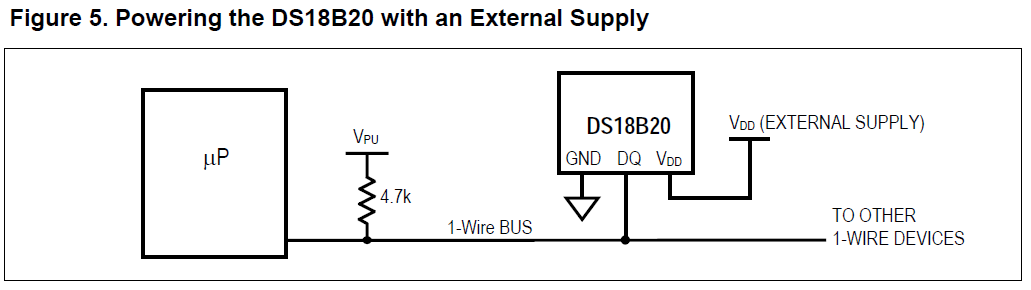


Figure - Circuit Schematic [1]

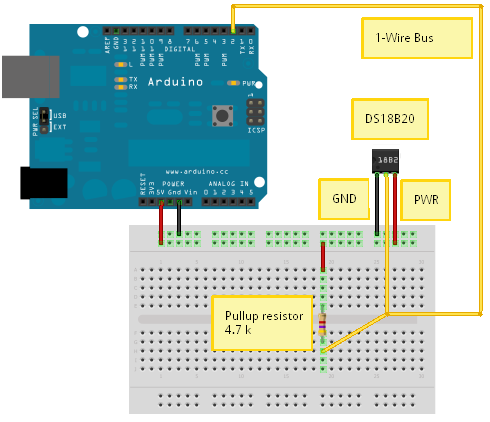


Figure – Physical Schematic[1]

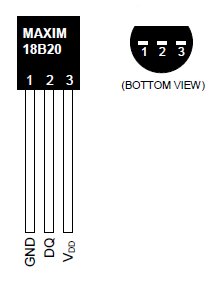


Figure - Temperature Sensor

Software



Figure - Software Architecture Diagram

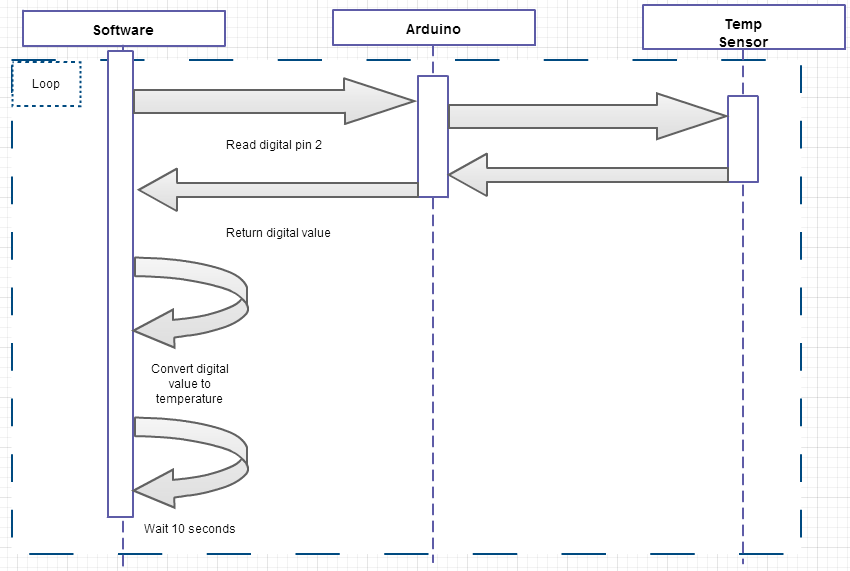
****

Figure - Hardware/Software Sequence Diagram

**Design**

Software



Photo of the Hardware

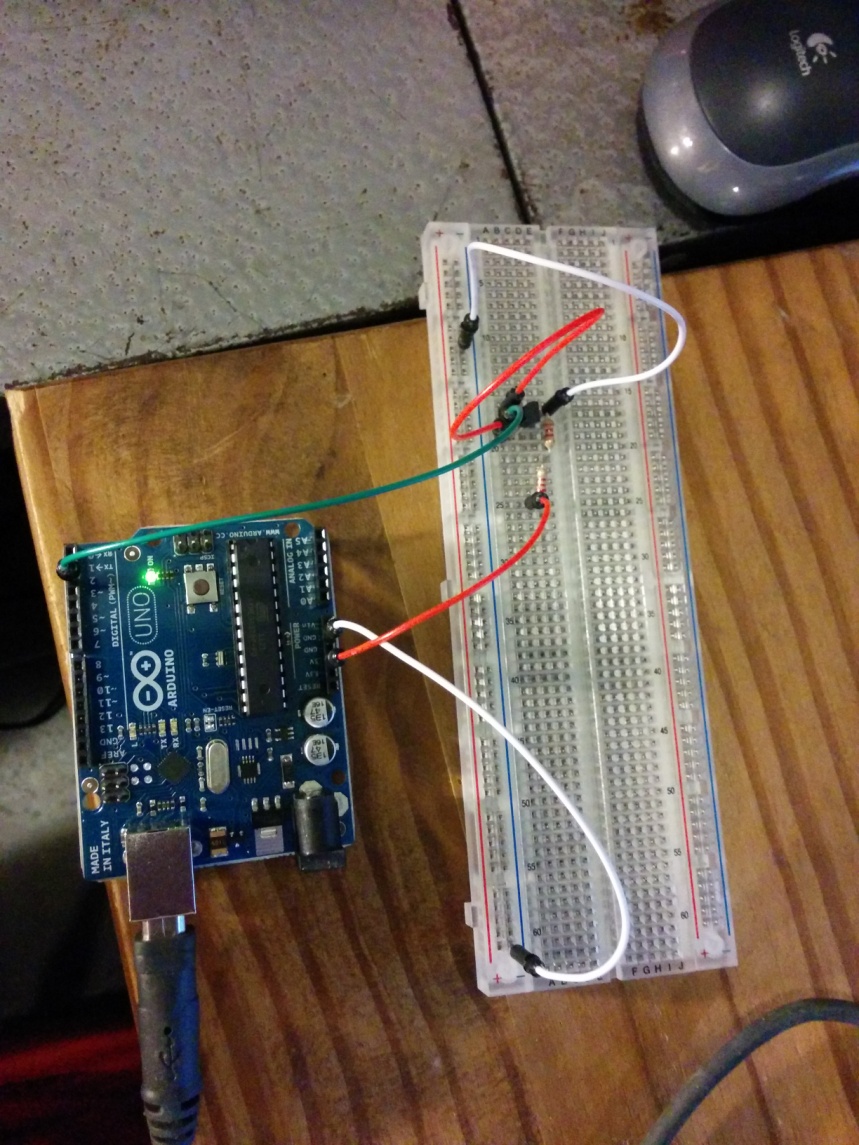
****

Figure - Photo of the hardware in action

**Implementation**

**#include <OneWire.h>**

**#include <DallasTemperature.h>**

**//JHU RTSW HW 1 - Calibrated Temperature**

**//Tony Florida**

**//2014-09-16**

**//References: http://www.hobbytronics.co.uk/ds18b20-arduino**

**// Data wire is plugged into pin 2 on the Arduino**

**#define ONE\_WIRE\_BUS 2**

**// Setup a oneWire instance to communicate with any OneWire devices**

**// (not just Maxim/Dallas temperature ICs)**

**OneWire oneWire(ONE\_WIRE\_BUS);**

**// Pass our oneWire reference to Dallas Temperature.**

**DallasTemperature sensors(&oneWire);**

**void setup(void)**

**{**

**// start serial port**

**Serial.begin(9600);**

**Serial.println("JHU RTSW HW1");**

**Serial.println("Tony Florida");**

**Serial.println("2014-09-16");**

**// Start up the library**

**sensors.begin();**

**}**

**//remember the most recent temperatures**

**int ARRAY\_LEN = 5;**

**int recent\_temperatures[] = {0.0, 0.0, 0.0, 0.0, 0.0};**

**int index = 0; //index into temperature array**

**boolean temp\_stabilized = false; //remember once the temp has stablilize**

**//function to check if the temperature has stabilized**

**boolean stabilized()**

**{**

**//get the first**

**double temp = recent\_temperatures[0];**

**for(int i = 1; i < ARRAY\_LEN; i++)**

**{**

**if(recent\_temperatures[i] != temp)**

**{**

**return false;**

**}**

**}**

**temp\_stabilized = true;**

**return true;**

**}**

**//main round robin loop**

**void loop(void)**

**{**

**// call sensors.requestTemperatures() to issue a global temperature**

**sensors.requestTemperatures();**

**double temp = sensors.getTempCByIndex(0);**

**//wait until the temperature stabilizes**

**if(!temp\_stabilized)**

**{**

**//keep index between 0 and 5**

**index++;**

**if(index > ARRAY\_LEN)**

**{**

**index = 0;**

**}**

**//save the temperature into the array**

**recent\_temperatures[index] = temp;**

**//check stabilization**

**stabilized();**

**Serial.println("Waiting for temperature to stabilize...");**

**}**

**else**

**{**

**Serial.print("Temperature is: ");**

**// 0 refers to the first IC on the wire**

**Serial.println(temp);**

**delay(10000); //delay 10 seconds**

**}**

**}**

**Results**

Log

JHU RTSW HW1

Tony Florida

2014-09-16

Waiting for temperature to stabilize...

Waiting for temperature to stabilize...

Waiting for temperature to stabilize...

Waiting for temperature to stabilize...

Waiting for temperature to stabilize...

Waiting for temperature to stabilize...

Temperature is: 22.12

Temperature is: 22.06

Temperature is: 22.06

Temperature is: 22.06

Temperature is: 22.06

Temperature is: 22.00

Temperature is: 22.06

Temperature is: 22.12

Temperature is: 22.19

Temperature is: 22.19

Temperature is: 22.19

Temperature is: 22.19

Temperature is: 22.19

Temperature is: 22.19

Temperature is: 22.19

Temperature is: 22.19

Temperature is: 22.19

Temperature is: 22.25

Temperature is: 22.25

Temperature is: 22.25

Temperature is: 22.25

Temperature is: 22.31

Temperature is: 22.25

Temperature is: 22.25

Temperature is: 22.25

Temperature is: 22.25

Temperature is: 22.19

Temperature is: 22.25

Temperature is: 22.25

Temperature is: 21.56

Temperature is: 20.44

Temperature is: 19.44

Temperature is: 18.31

Temperature is: 17.56

Temperature is: 17.12

Temperature is: 16.56

Temperature is: 16.00

Temperature is: 15.63

Temperature is: 15.31

Temperature is: 15.06

Temperature is: 14.81

Temperature is: 14.50

Temperature is: 14.13

Temperature is: 14.00

Temperature is: 13.81

Temperature is: 13.75

Temperature is: 13.50

Temperature is: 13.31

Temperature is: 13.13

Temperature is: 12.94

Temperature is: 12.75

Temperature is: 12.63

Temperature is: 12.50

Temperature is: 12.38

Temperature is: 12.25

Temperature is: 12.25

Temperature is: 12.06

Temperature is: 12.06

Temperature is: 12.00

Temperature is: 13.00

Temperature is: 14.25

Temperature is: 14.94

Temperature is: 15.56

Temperature is: 16.06

Temperature is: 16.50

Temperature is: 16.75

Temperature is: 17.00

Temperature is: 17.31

Temperature is: 17.69

Temperature is: 17.81

Temperature is: 17.94

Temperature is: 18.19

Temperature is: 18.25

Temperature is: 18.44

Temperature is: 18.44

Temperature is: 18.62

Temperature is: 18.62

Temperature is: 18.69

Temperature is: 18.75

Temperature is: 18.87

Temperature is: 19.00

Temperature is: 19.00

Temperature is: 19.12

Temperature is: 19.19

Temperature is: 19.31

Temperature is: 19.37

Temperature is: 19.37

Temperature is: 19.44

Plot

**References**

[1] <http://tushev.org/articles/arduino/item/52-how-it-works-ds18b20-and-arduino>

[2] <http://www.hobbytronics.co.uk/ds18b20-arduino>