Spring 2011

Two-wire, low component count soil temperature sensor

Nicholas James Sitter
University of Iowa

Recommended Citation

Sitter, Nicholas James. "Two-wire, low component count soil temperature sensor." MS (Master of Science) thesis, University of Iowa, 2011.
https://doi.org/10.17077/etd.t5hd7xxv

Follow this and additional works at: https://ir.uiowa.edu/etd
Part of the Electrical and Computer Engineering Commons

TWO-WIRE, LOW COMPONENT COUNT

SOIL TEMPERATURE SENSOR

by

Nicholas James Sitter

A thesis submitted in partial fulfillment
of the requirements for the Master of
Science degree in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

May 2011

Thesis Supervisor: Associate Professor Anton Kruger

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

____

MASTER'S THESIS

____

This is to certify that the Master's thesis of

Nicholas James Sitter

has been approved by the Examining Committee
for the thesis requirement for the Master of Science
degree in Electrical and Computer Engineering at the May 2011 graduation.

Thesis Committee: ______________________________________
Anton Kruger, Thesis Supervisor

____
David Andersen

____
Jon Kuhl

To my family

ACKNOWLEDGMENTS

I would like to thank my committee members for serving on my committee. I would also like to thank my colleagues at IIHR and the Iowa Flood Center for their support, especially Dr. Jim Niemeier for discussing the architecture of the current rain gauge and soil moisture networks, as well as helping set up experiments for calibrating and testing the temperature sensors. I would also like to thank the Iowa Flood Center for their financial support throughout this project. Finally, I would like to thank my advisor, Dr. Anton Kruger, for his endless support. If it were not for his guidance, this work would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Temperature measurements are vital in a number of applications which span several areas. A few of these areas include industrial, consumer, and military settings. Temperature is also an important variable in many geophysical applications including weather, soil, and water temperature. As a result, many techniques and devices for measuring temperature exist. However, there are areas where temperature measurements techniques can be significantly improved. Existing techniques will be discussed briefly in this chapter; however, the rest of the thesis is dedicated to the development of a soil temperature sensor for geosciences applications.

Motivation

Recent developments and deployments of environmental sensor networks at The University of Iowa, more specifically IIHR-Hydroscience and Engineering (IIHR) and the Iowa Flood Center (IFC), have motivated the need for a low cost and robust soil temperature sensor. These sensors will be deployed across Iowa along with other sensors to form an environmental sensing network. This network will provide IFC with soil temperature measurements, which are necessary to determine the ground freezing level. The ground freezing level impacts runoff that can significantly contribute to flooding. By providing IFC with soil temperature data, they will be able to more accurately predict flooding across the state of Iowa.

IIHR has had a network of rain gauge sites deployed around the state since 2006. These sites provide researchers at IIHR with accurate rainfall measurements. IIHR has also had a soil moisture network deployed in Ames, Iowa since 2007. This site, described in [14], provides researchers at IIHR with both soil moisture and soil temperature. Recently, IFC has been developing the next generation rain gauge sites, which incorporate both soil moisture and soil temperature sensors. These sites, described in [5], will be deployed across the state and will provide researchers at The University of Iowa with the real-time data necessary to provide flood forecasts to local communities.

As shown in Figure 1, a typical node in this network consists of dual tipping-bucket rain gauges, soil moisture probes, and soil temperature sensors. The soil moisture probes and soil temperature sensors plug into a junction box. This junction box communicates with the data logger over a RS-485 bus. The data logger collects data from all of the sensors, connects to a server located at The University of Iowa using a cell modem, and submits the data to the server. The server stores all of the data in a database, which can be accessed by researchers at IIHR through a web interface.

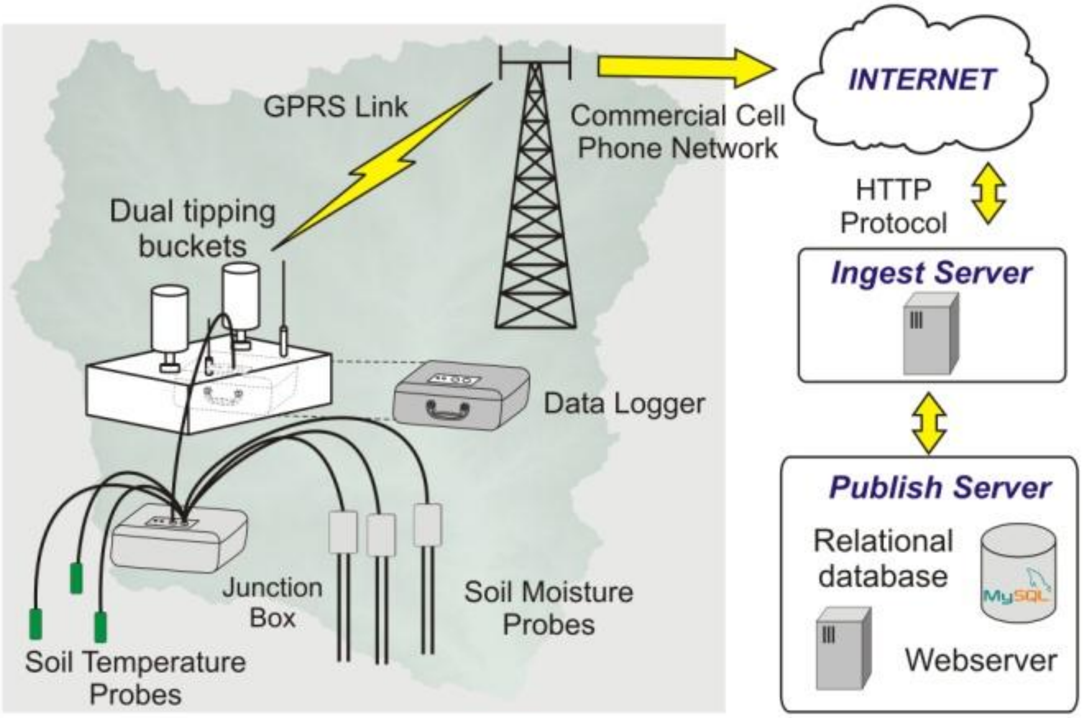


Figure 1. Block diagram of next generation rain gauge site being developed by IFC from [5]. Each site contains soil moisture and soil temperature sensors, as well as dual tipping-bucket rain gauges. A data logger collects all of the sensor data and transmits it back to a server at The University of Iowa.

Redundancy is very important in a system that is deployed in an outdoor environment, as it increases the system's reliability. It takes a significant amount of work to obtain permission from landowners to deploy new sites. Many of these sites are in rural areas and it is very time consuming to service them. Some sensors malfunction for some time, but then recover and function properly. It is difficult to detect these situations without additional redundancy built into each site. This is why two tipping-bucket rain gauges, four soil moisture probes, and eight soil temperature sensors are deployed at each site. The use of many soil moisture and soil temperature sensors provides researchers with a much more distributed measurement, since sensors can be placed at various depths and locations around the site. Thus, the redundant sensor approach can greatly improve data availability and quality.

The Real Cost of Outdoor Sensors

Many factors must be considered when designing sensors that will be deployed outdoors, some more obvious than others. One major factor is the temperature range the sensor must operate in. This restricts what components can be used in the sensors, as the sensor must operate correctly across the entire temperature range. Another factor is the weather conditions the sensor will be exposed to. If the sensor will be exposed to rain or moisture, it needs to be sealed such that it is protected from water infiltration. Electrical or electromagnetic noise may also be a factor depending on where the sensors are deployed, so the sensor must still operate in a noisy environment. A less obvious decision that needs to be made is what type of cable to use to connect the sensor to the data logger. Several factors need to be considered when choosing cable, such as quality and type, which both affect the cost of the wire.

The type of cable used for outdoor applications is extremely important. Engineers at IIHR have significant experience deploying electronics outside and in other harsh environments. They have learned from years of experience that Teflon-coated wire works very well for environmental deployments. Not only does is hold up well to the elements, but unlike other types of wire, animals will not chew on the Teflon-coated wire. Teflon-coated wire is also very rugged and will not wear down in the soil, rain, snow, or sunlight. These advantages to using Teflon-coated wire make it very desirable compared to other types of wire.

The cost of cable makes up a significant portion of the overall cost, and is frequently more expensive than the sensor itself. For example, high quality stranded 22 AWG (American Wire Gauge) Teflon-coated hookup wire costs around $100 for 100 feet. This means a 10 meter twisted-wire pair costs about $70, while a three wire cable of the same length costs $100. Thus, moving from a three-wire interface to a two-wire interface results in a saving of $30. In many locations eight soil temperature sensors will be deployed, so it costs about $240 less to deploy a two wire interface instead of a three wire one.

Requirements for a Soil Temperature Sensor

There are a number of requirements that a temperature sensor must meet to be suitable for this application. The sensor must be robust, since it will be buried for several years, in environments such as Iowa, where large temperature cycles are possible. The cables connecting the sensor to data logging equipment must be critter-resistant. The sensor should cost less than $10, not including the cost of cable. It should utilize a minimum number of components so that it is easy to construct. Therefore, the sensor must not require special order parts with long lead times. With respect to cable length and noise immunity, the sensor must work satisfactorily with cable lengths up to 50 meters. The data logger interface should be simple and insensitive to timing and slew rate. The polling time does not need to be extremely fast, but it should take less than a few seconds. Ideally, one would not calibrate the sensor, but simple one or two point calibration may be acceptable in some instances. Our current application requires the sensor to be accurate over the range $-10^{\circ}\mathrm{C}$ to $+50^{\circ}\mathrm{C}$. It should also have a thermal time constant on the order of a few minutes and an accuracy of $\pm 2^{\circ}\mathrm{C}$ or better. The sensor will measure temperature at a low duty cycle, for example once every 15 minutes, so power consumption during a measurement is less important, as the quiescent power consumption should be very low.

Electronic Temperature Sensing Methods

There are a variety of electronic temperature sensors available in the market today, including resistance temperature detectors, thermistors, thermocouples, and semiconductor-based temperature sensors. Resistance temperature detectors, thermistors, and thermocouples can be thought of as analog sensors, which often require analog signal processing before a data logger digitizes their output signal. Semiconductor-based temperature sensor ICs provide either an analog or digital output. A few of the more common electronic temperature sensing methods are discussed briefly below. See [3], [6], [13], and [23] for more information on these methods.

Resistance temperature detectors, or RTDs, provide very accurate temperature measurements. In fact, RTDs are typically more accurate than thermocouples within the temperature range from $-180^{\circ}\mathrm{C}$ to $+650^{\circ}\mathrm{C}$. RTDs are also used due to their high stability and repeatability. The resistance of metal changes with temperature in a known and predictable way. The resistance of metal increases as the temperature increases and decreases as the temperature decreases. RTDs exploit this characteristic in order to measure temperature. Typically, RTDs are constructed from platinum, copper, or nickel, as these metals have linear temperature coefficients of resistance over different temperature ranges. In order to measure temperature using a RTD, a constant current must be applied through the RTD. The voltage across the RTD is measured, so the resistance can be determined as the temperature changes. This technique is relatively simple and is not very prone to error.

Although RTDs are desirable for their stability, linear temperature coefficients, and wide temperature range when compared to digital temperature sensors, there is one major problem with using them for this application. When used to measure temperature across long cables, the changes in resistance due to the cables can significantly affect their accuracy. In order to compensate for these errors, extra cables can be used along with other analog circuitry to effectively cancel the resistance from the wires running to the RTD. This greatly increases the cost of the sensor, as more cables must run to the sensor in order to make it accurate at long distances. Another approach would be to place this additional analog circuitry close to the RTD itself, but then there still needs to be a way to read the temperature and relay this information back to the master. The number of components required to implement a RTD based temperature sensor is far greater than other approaches.

Thermistors are very similar to RTDs, as their resistance also changes as a function of temperature. However, unlike RTDs, thermistors are constructed from ceramic semiconductors instead of metal. This results in much larger temperature coefficients of resistance for thermistors than RTDs. Thermistors can have either a positive temperature coefficient, known as a PTC thermistor, or a negative temperature coefficient, known as a NTC device. Thermistors are also desirable for their stability, as they are typically more stable than thermocouples and even some RTDs. The same technique used to measure temperature with a RTD is used for a thermistor. A constant current is sent through the thermistor, generating a voltage across it. This voltage changes with temperature due to changes in the thermistor’s resistance.

Although thermistors provide a simple way to measure temperature over a wide range and can measure very small changes in temperature when compared to digital temperature sensors, there is one major disadvantage to using them for this application. Thermistors are difficult to calibrate, as they are nonlinear. In order to provide an accurate temperature measurement, the device must have some way to eliminate the errors from nonlinearities. This could consist of making several measurements at different temperatures over the expected range and find an equation for a curve of best fit. Although some thermistors are designed to be curve matched, these are typically expensive and are only available in limited temperature ranges. Another option is to design a linear thermistor network consisting of resistors in series or parallel with the thermistor. However, this solution requires more components than desired for this application. Either way, a thermistor based temperature sensor is too costly compared to other possible solutions.

Thermocouples are widely used to measure temperature due to both their versatility and low cost. A thermocouple is constructed by joining two electrical conductors that are made of thermoelectrically dissimilar materials at a junction. A voltage is produced due to a difference in temperature between the two junctions, which can then be converted to a temperature. This potential difference resulting from a difference in temperature is a result of the Seebeck effect. See [23] for more details on this effect. There are several different types of thermocouples, which correspond to thermocouples constructed using different materials. Both sensitivity and temperature range vary from type to type.

Despite the low cost and versatility of thermocouples, there are a few major setbacks to using them for this application. Although thermocouples can be used to measure temperature at long distances, heavy gauge wire must be used due to their high resistance, which greatly increases the cost. Compensating cable may also be used, but it must be matched with the thermocouple in order to provide an accurate measurement. Typically, these cables are only matched with the thermocouple over a small temperature range, and do not provide the accuracy that the first option does. Also, in order to provide accurate temperature measurements, thermocouples must be calibrated when installed, and also need to be recalibrated on a regular basis. For these reasons, thermocouples will not work for this application.

Semiconductor-based temperature sensors provide a low cost, quality solution for measuring temperature. They take advantage of the temperature sensitivity of the semiconductor's pn junction being predictable over the range from $-55^{\circ}\mathrm{C}$ to $+150^{\circ}\mathrm{C}$. The majority of semiconductor-based temperature sensors use a bipolar transistor connected like a diode, where the collector-base junction is short-circuited. A voltage is generated across the base and emitter when a constant current is passed through the base-emitter junction. The relationship between the voltage and temperature is linear is approximately equal to $-2\ \mathrm{mV}/^{\circ}\mathrm{C}$. This temperature coefficient is larger than that of a thermocouple or RTD, and also much more linear. Additional circuitry is typically integrated to improve the accuracy and provide different interfaces to a microcontroller. Semiconductor-based temperature sensors are also commonly embedded in other ICs, such as microcontrollers to provide them with additional functionality.

As mentioned above, most semiconductor-based temperature sensors incorporate additional circuitry to improve performance. For analog output temperature sensor ICs, this circuitry can amplify and provide a level shifted output in either degrees Celsius or degrees Fahrenheit. These sensors typically output a voltage or current proportional to the measured temperature. Digital output temperature sensor ICs provide analog to digital conversion, as well as provide temperature measurements using some digital interface. Some of these interfaces are discussed in the next section.

Although semiconductor-based temperature sensors are cheap, linear, and easily integrated into ICs, there is a major issue that needs to be overcome in order provide accurate temperature measurements. Due to variations in the manufacturing process, the semiconductor-based sensor's forward voltage varies significantly from IC to IC. This means the sensors must be calibrated to provide accurate temperature measurements. However, since they are linear, only two temperature points need to be measured to calibrate the sensor. Sometimes only one point is needed to account for the temperature offset, but the sensor is more accurate when the variation in slope is taken into account as well. Although semiconductor-based temperature sensors require calibration, they are the best option for this application, since they are inexpensive, linear, and are easily integrated with a microcontroller.

Three different temperature sensors are used throughout the rest of this thesis for making temperature measurements. These three sensors include the Omega HH-42 handheld thermometer, the DS18B20 digital thermometer IC, and the Atmel AVR ATtiny85V microcontroller. Table 1 provides a comparison of some important criteria for these three sensors. These criteria include the temperature measurement range, sensor accuracy, unit cost, interface, advantages, and disadvantages. It is important to note that although the accuracy of the DS18B20 is specified as ±2°C for the temperature range from −55°C to +125°C in the table below, it is accurate to within ±0.5°C across the temperature range our sensor is required to measure, which is from −10°C to +50°C.

Table 1. Comparison of three sensors used for temperature measurements

	Omega HH-42	**DS18B20**	**ATtiny85V**
Temp Range	−20°C to +130°C	−55°C to +125°C	−40°C to +85°C
Accuracy	±0.015°C	±2°C	±10°C
Cost	$499	$5.99	$2.26
Interface	RS-232	1-Wire	Analog voltage
Advantages	NIST calibrated	3-pin IC	Integrated sensor
Disadvantages	Very expensive	Strict timing	Needs calibration

Each of these temperature sensors serves a very specific purpose. The Omega HH-42 is a very high accuracy thermometer that has been calibrated by the National Institute of Standards and Technology, or NIST. This sensor is used as the reference for all of the experiments discussed below, as well as for calibration of other sensors. See [17] for more detailed information on the HH-42. The DS18B20 is 3-pin thermometer IC that is very easy to interface with using a microcontroller. This sensor is used the temperature sensor for the original prototype, as well as the reference for single point temperature calibrations. See [12] for more details. Finally, the ATtiny85V has an integrated temperature sensor that replaces the DS18B20 in the minimal component count temperature sensor. See [1] for more information on the ATtiny85V.

Digital Communication Interfaces

There are several different communication interfaces used in practice today for communicating between devices. Not every interface will be discussed here, however, a few of the more commonly used ones will be. The majority of digital temperature ICs use 1-Wire, SPI, or I^2C interfaces. There is one major issue with using these interfaces for this application, and that is that they do not operate reliably on long cables across wider temperature ranges. All three of these interfaces were designed to communicate between devices on a single board or across short cables.

The Serial Peripheral Interface, or SPI, bus supports full duplex communication. By using full duplex communication, SPI has a higher throughput than I^2C. Another advantage of SPI is it uses a very simple hardware interface. This typically results in lower power consumption than I^2C since less circuitry is needed. SPI uses a six wire protocol which consists of power, ground, a Serial Clock (SCL) line, a Master Output Slave Input (MOSI) line, a Master Input Slave Output (MISO) line, and a Chip Select (CS) line. SPI supports multiple devices on the same set of wires, although each device needs an individual CS line. However, if there is only one sensor on the bus, the CS line can be hardwired and five wires will suffice. See [8] for more details on SPI.

One major disadvantage to using SPI is that it requires more pins than I^2C or 1-Wire. This makes SPI undesirable for this application since it requires six cables and will be expensive to deploy. Another issue with SPI is that it is prone to noise, which can lead to faulty communication. This could result in erroneous data when used on long wires, and therefore is not suited for this application. SPI communication also occurs at relatively fast speeds, so the long wires could lead to signal integrity issues. Finally, SPI will not work for this application since it was designed for communication between devices on a single PCB. Therefore, it will not work on very long cables up to 50 meters.

Another popular digital interface used for temperature sensors is the Inter-Integrated Circuit, or I^{2}C, bus. The I^{2}C bus uses two bidirectional open-drain lines with pull-up resistors, which decreases the number of wires compared to SPI by one. An in-band address also reduces the number of wires compared to SPI by one, since the slave no longer has to be selected using hardware. Instead, the bus master selects an individual sensor on the bus by transmitting the sensor's unique address. I^{2}C also provides a slave acknowledgement, unlike SPI, so the master knows if a slave is present at the specified address. I^{2}C implements a four wire interface which consists of power, ground, a Serial Clock (SCL) line, and a Serial Data (SDA) line. See [16] for more details on I^{2}C.

Although I^{2}C is more suited for this application than SPI, there are still issues with using it. Since I^{2}C was designed to connect ICs to a motherboard, microcontroller, or cell phone, it does not work on long wires. It works fine on the same circuit board, within the same enclosure, or on a short pigtail outside of an enclosure, but not on cables up to 50 meters long. The data rate used for I^{2}C is also fast enough that it may suffer from signal integrity issues when used across long wires. For these reasons, I^{2}C will not work well for this application.

In the search for fewer wires, Dallas/Maxim Semiconductor developed its 1-Wire communication interface. The 1-Wire interface is very similar to I^{2}C with a few minor differences. 1-Wire operates at a lower data rate than I^{2}C, which makes it better suited for this application. 1-Wire also works on much longer wires than I^{2}C, which is very important for this application. Finally, 1-wire uses one less wire than I^{2}C in its normal configuration, since the data and clock line are combined, resulting in cheaper deployment costs. The 1-Wire interface consists of power, ground, and one additional wire for bidirectional communication. Thus, the 1-Wire refers to only using one wire for communication, although three wires are still required. However, 1-Wire devices can be powered parasitically using the data line, which only requires two wires. This would be ideal for this application, since it minimizes the number of cables required.

Although 1-Wire sensors operate on longer cables than many other digital interfaces, they still have a relatively short limit. In [10] and [11], methods have been developed to extend the range of 1-Wire sensors to hundreds of meters, but require complicated or high component count circuits. Also, 1-Wire has strict timing requirements, and if these requirements are not met, no temperature measurements can be received by the master. Issues arise at extremely cold temperatures, since the sensor's internal clock drifts outside the range necessary to meet the timing requirements. For these reasons, the 1-Wire interface will not work reliably for this application.

There are many other digital interfaces available, some of which represent minor variations of interfaces described here. There are also many ways to transmit analog signals, such as using a voltage-to-frequency converter. However, none of these interfaces suits all of the needs for this application. Many of them provide fast data rates which are not necessary, or even desired for this sensor. Also, many will not operate on long cables, which is required for this application. Most of these interfaces require strict timing, which is difficult to meet across the entire temperature range. Finally, the majority of these interfaces require too many wires or components, which significantly increases the cost of each sensor. This is why a simple pulse width modulation interface was developed. The use of pulse width modulation for communicating temperature data is discussed in more detail below.

Proposed Solution

The proposed two-wire, low component count soil temperature sensor is shown below in Figure 2. The sensor will have one wire for ground and the other wire will be used for both power and communication. Pulse width modulation will be used to send temperature measurements to the master, where the duty cycle is proportional to the temperature. The sensor will parasitically power itself from the bidirectional data line. In order to reduce the number of components necessary, a microcontroller with an internal temperature sensor will be used. Finally, the sensor will be able to receive data from the master on the bidirectional communication line, which will be used for calibrating the sensor.

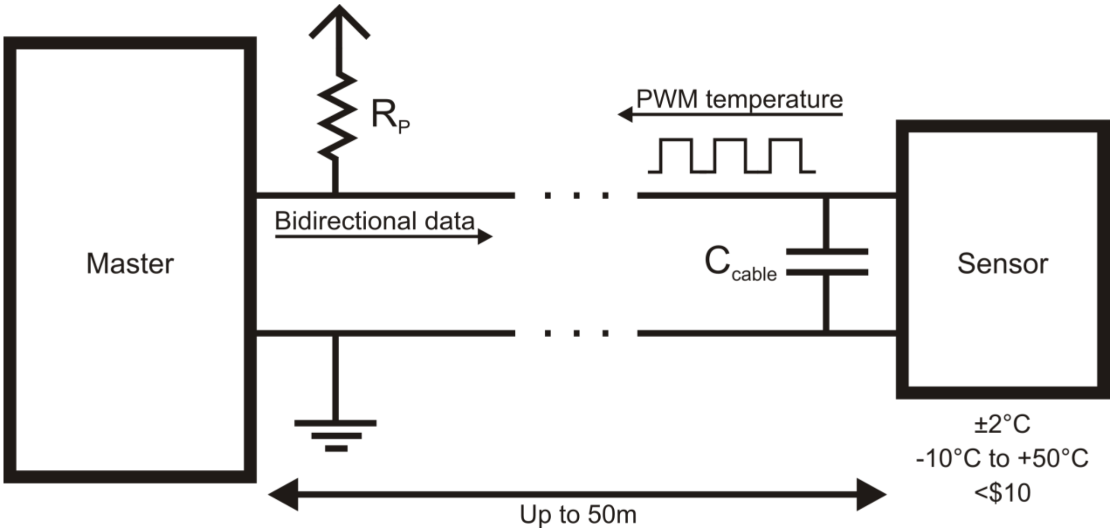


Figure 2. Proposed two-wire, low component count soil temperature sensor. The sensor is connected to a master using a two-wire interface. One wire provides ground and the other wire is used for power and communication. The sensor is powered parasitically from the bidirectional data line. Pulse width modulation is used to send temperature measurements to the master and the bidirectional communication line provides calibration data to the sensor.

CHAPTER II

SYSTEM OVERVIEW

In this chapter, the proposed system for the two wire temperature sensor is discussed in more detail. More specifically, this includes the basic hardware for a low component count sensor. The concept of using parasitic power for powering the sensor from the data line is introduced. The use of pulse width modulation as a method of communication will also be explored. Finally, the use of bidirectional communication for sending calibration information to the sensor will be described in detail.

Low Component Count Hardware

Figure 3 is a schematic of the basic sensor we developed consisting of an 8-pin microcontroller, a digital temperature sensor IC, a capacitor, two diodes, and three resistors. The temperature IC is a Dallas DS18B20 1-Wire digital thermometer, which provides the temperature measurement for the sensor. Other sensors can be used, such as Analog Device's TMP36 ratiometric analog temperature IC or NXP Semiconductor's SE95 I^{2}C sensor. We discuss yet another option later in more detail. The circuit works as follows: the microcontroller interrogates the temperature sensor IC for a temperature measurement. It uses the measured value to adjust the duty cycle of a square wave. The microcontroller outputs this square wave on a microcontroller pin, which is connected to the incoming power supply line via a low value resistor R_2. The result is a square wave current fluctuation in the power supply wires. A data logger or master microcontroller located several meters away can sense the fluctuation in current via the pull-up resistor R_4: when the sensor connects the low value resistor to ground, there is a large voltage drop across the sense resistor.

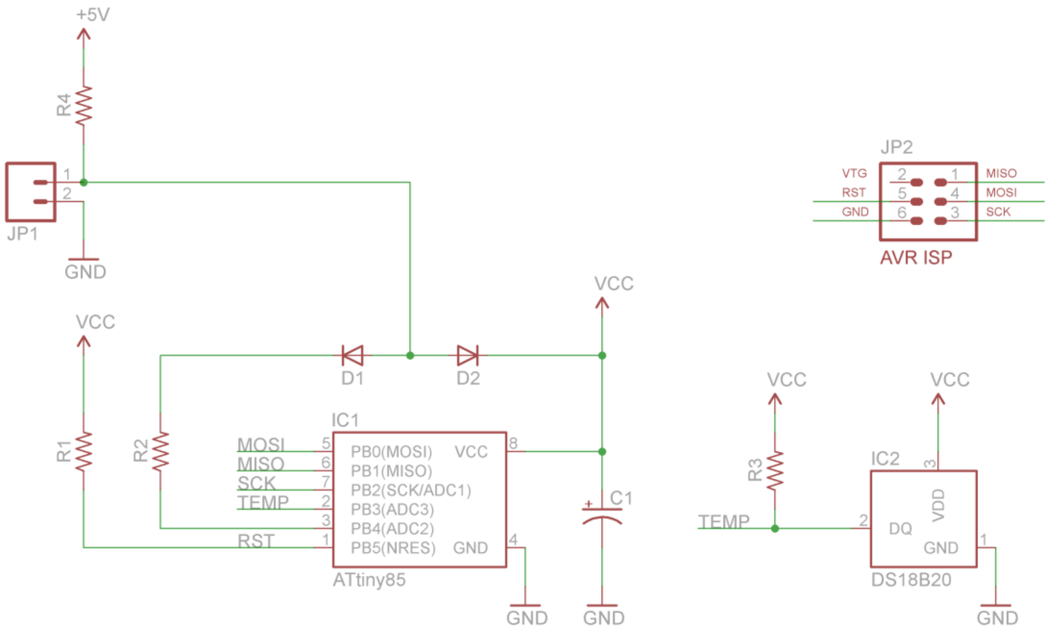


Figure 3. Hardware schematic for prototype temperature sensor with DS18B20. The master supplies power to the microcontroller over the power line. The microcontroller measures the DS18B20's temperature and transmits a square wave with a duty cycle proportional to this temperature. The sensor continues to send temperature measurements until the master removes power.

This signaling scheme implies that the power supply to the sensor microcontroller will fluctuate between a value close to V_{CC} and V_m where V_m depends on the diode D_1, the pull-up resistor R_p located on the master, and the low value resistor R_2. The resistor values were chosen such that V_m is about 500 mV, which means the microcontroller will lose power unless there is some local energy storage. This is where the concept of parasitic power comes into play, which is discussed below. The purpose of D_1 is to ensure that the voltage at the pull-down pin does not exceed the voltage at the microcontroller's V_{cc} pin, and it also provides reverse-polarity protection. Conveniently, there are a number of dual Schottky diodes in a single package on the market. For example, the BAT54A contains two common-anode Schottky diodes in a small, SOT-23 package. This will reduce the overall component count by one.

The microcontroller runs off an internal 8 MHz RC clock, which is internally divided by 8, resulting in a 1 MHz clock. A low clock frequency translates into lower power consumption. Variations in the microcontroller manufacturing process make the exact oscillator frequency uncertain. Furthermore, the clock frequency is a function of supply voltage and temperature, however, since the microcontroller maps measured temperatures to the duty cycle of a square wave, deviations from the nominal RC oscillator frequency are not important. The data logger or master microcontroller simply needs to measure the ratio of the on to off time.

In operation, the data logger or master microcontroller turn on the sensor, wait long enough for C_1 to charge, and then the microcontroller starts execution and makes a measurement. During this time, the microcontroller floats the pull-down resistor R_2. Next, the sensor transmits two or more cycles with duty cycle D, where D is a linear function of the temperature. In noisy environments, the sensor microcontroller can be programmed to transmit more than two cycles, which will allow the receiver to average across several cycles. Alternatively, the data logger or master microcontroller can make several different measurements in rapid succession and average them. In any event, once the master has obtained a measurement, it turns off the power to the sensor. At this time the sensor microcontroller continues to run, powered by C_1. Once the voltage across the capacitor drops below the threshold, the microcontroller turns off. At this point, the sensor is no longer consuming any power.

Parasitic Power

Power is the most important component of any sensor, as the sensor will not operate without it. In order to keep the temperature sensor small, there is not room for a battery, so an external source of power is required. Typically, electronic sensors have at least three wires, where two of them are dedicated to ground and power. Any remaining wires are usually used for communication. There are several different methods for doing this that were discussed earlier. In this case, however, a minimal cable approach is desired since wire is so expensive relative to the sensor itself. For this reason only using two wires would be ideal. One wire would be used as ground and the other cable would provide both power and communication.

The use of parasitic power for providing power over the data line has been around for years. The DS18B20 digital temperature IC uses parasitic power to provide a two-wire interface by stealing power from the data line and charging a small capacitor. This capacitor then powers the sensor for very short durations while the IC communicates over the data line. See [12] for more details. In [22], a bidirectional, single-wire, line powered transceiver is discussed. This transceiver uses a microcontroller to transmit data by modulating the power line. A comparator is used to decode the data modulated on the power line and send it to the microcontroller. The concept of parasitic power has also been used to develop Power over Ethernet, or PoE. PoE was developed to eliminate the need for network devices to be connected to an external AC power supply. This not only provides more flexibility to the network devices, but it also reduces the cost and enhances the safety of these devices since they no longer need to be placed close to an AC outlet. PoE works by providing 48 VDC over the same wire as data. See [9] and [24] for more details on PoE.

The use of parasitic power allows the sensor to communicate with the master while also powering itself temporarily from some local storage reservoir. In this case, a capacitor is used as the storage device and a diode prevents charge from leaking over the power line when it is grounded. When the power line is high, the capacitor charges until the sensor pulls the power line low while communicating. This capacitor then provides power to the sensor until the sensor lets the power line go high again. This process repeats until the master stops providing power to the sensor. The basic hardware configuration used for parasitic power is described below.

The Schottky diode D_2 and the capacitor C_1 in Figure 4 are used for parasitically powering the microcontroller. Once the sensor has made a temperature measurement, the microcontroller switches R_2 to ground and the voltage at the sense node drops to V_m. This results in the diode D_2 being reverse-biased, which blocks current flow from C_1 via the diode. Now the capacitor is providing power to the microcontroller. The pull-up resistor R_3 located at the master senses the voltage changes on the line and converts the square wave to a temperature. When the microcontroller disconnects R_2 from ground, V_m rises to a value close to V_{cc}. This forward-biases D_2 and the master begins to power the microcontroller and charge the capacitor. As soon as the microcontroller pulls the line low again to start sending another temperature measurement, this cycle starts over again.

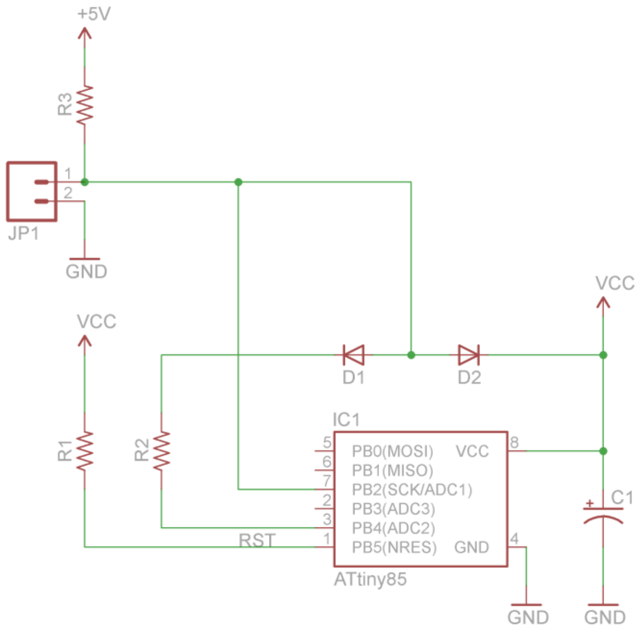


Figure 4. Basic temperature sensor hardware schematic for explaining parasitic power. The master supplies power to the microcontroller which is stored in $C1$. The microcontroller pulls the power line low to send its temperature to the master. When this happens, $D2$ is reverse-biased and $C1$ powers the microcontroller until it releases its power line. This repeats until the master removes power and the capacitor $C1$ discharges enough that it can no longer provide enough power to the microcontroller. An additional pin on the microcontroller is used for bidirectional communication, which is discussed in detail below.

This process repeats until the master removes power from the power line. At this point, the capacitor C_1 continues to power the microcontroller until the voltage drops below the microcontroller's required operating voltage and it turns off. The next chapter provides a detailed analysis of key concepts used for parasitic power.

Pulse Width Modulation

A very simple, yet robust way of transmitting temperature readings across one wire is to use pulse width modulation, or PWM. See [20] for an example of transmitting temperature measurements by varying a pulse's width. Although using PWM requires slower communication rates due to such a large swing in voltages, in this case from 0 V to 5 V, this is acceptable since only small amounts of data need to be sent. Pulse width modulation has advantages over other forms of communication in this environment. PWM does not rely on an accurate clock or clock synchronization that many other communication methods require for a reliable link. The information is contained in the ratio between on time and period, called the duty cycle. Even if the clock drifts due to changes in temperature, the information will still be received correctly, as long as the master can measure the duty cycle with a resolution smaller than the resolution the temperature is transmitted with.

The temperature is transmitted such that one millisecond corresponds to one degree Celsius. This resolution was chosen since speed is not a factor and this allows plenty of margin for the master to measure the duty cycle accurately. As shown in Figure 5, a period of 400 milliseconds is used and the on time is equal to the ADC value from the temperature sensor. The relationship between the temperature and ADC values is shown below in Table 2. This relationship is linear, where each least significant bit of the ADC corresponds to one degree Celsius. A period of 400 milliseconds also provides the sensor with enough power during the on time of each cycle to use parasitic power when the temperature is very low. For example, when the sensor measures its minimum temperature of −40°C the duty cycle is still above 50 percent. This ensures that the capacitor used for parasitic power will have enough time to fully charge.

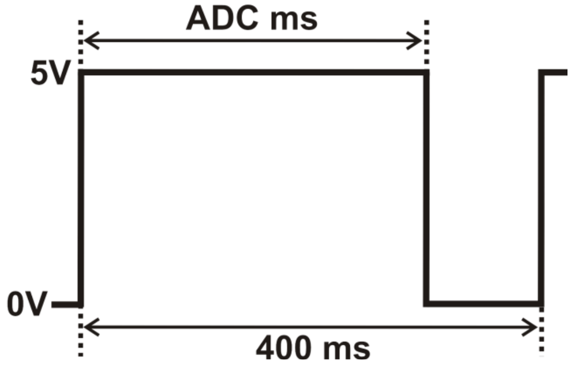


Figure 5. Pulse width modulated square wave output from temperature sensor. The temperature data is contained in the duty cycle of the square wave, with a period of 400 milliseconds. The on time is equal to the ADC value corresponding to the temperature of the sensor.

Table 2. Relationship between ATtiny85V ADC value and temperature

Temperature	−40°C	+25°C	+85°C
ADC	230 LSB	300 LSB	370 LSB

Bidirectional Communication for Calibration

A simple protocol for sending data to the sensor over the power line was developed. This allows the master to send a byte of data at a time, which is then interpreted by the sensor in a predetermined way. Bidirectional communication allows the master to send calibration information to the sensor. However, it also opens up a world of other possibilities to the system. For example, a boot loader could be implemented on the sensor microcontroller and the master could reprogram the sensor over the power line.

The bidirectional communication is used to send two bytes of calibration data to the sensor. Although this complicates the software, it provides a better solution for calibrating the sensor than programming the calibration information onto the microcontroller before soldering it to the PCB. In order to recalibrate that sensor, it would have to be unsoldered from the board and reprogrammed. Unsoldering the microcontroller, reprogramming it, and soldering it back onto the board is a very time consuming and unnecessary process. This functionality greatly increases the flexibility of the temperature sensor.

In order to use bidirectional communication, an additional pin on the ATtiny85V needs to be used in order to sense the actual level of the power line. This is shown above in Figure 4. By sensing this level, the temperature sensor can easily determine if the master is trying to recalibrate it or not. When the microcontroller senses the power line pulled to ground and the microcontroller did not pull the line low itself, it knows the master is sending data to it. The protocol used to communicate and then recover the calibration information from the power line is explained below.

After the temperature sensor is turned on and initialized, it waits 200 milliseconds for a transition from high to low on its power line. If the microcontroller does not sense a transition, it enters the normal mode of operation and begins transmitting temperature data. However, if the microcontroller senses a transition, the master is ready to send calibration information and the sensor immediately enters receive mode. Before the communication protocol is described in detail, the waveforms used to represent binary data are introduced.

Two simple square pulse waveforms are used to transmit either a one or zero. Both of these waveforms have a period of four milliseconds, as well as a duty cycle greater than or equal to 50 percent. The reason the pulse is high for at least half of the period is to insure that the storage capacitor has enough time to charge each cycle, otherwise the parasitic power could not keep the sensor powered. The two waveforms are shown below in Figure 6. The waveform on the left is a binary zero, which has an off time equal to one millisecond. The waveform on the right is a binary one, which has an off time equal to two milliseconds. The sensor can distinguish a one from a zero simply by looking at the off time of each pulse.

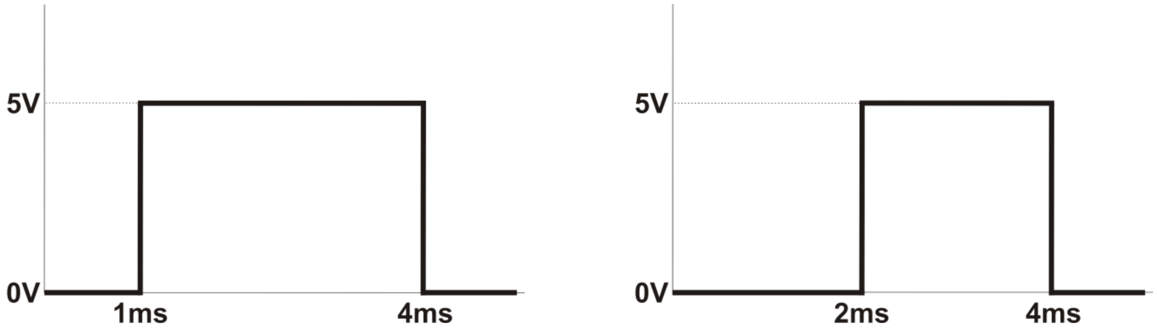


Figure 6. Waveforms used for sending calibration data to the sensor from the master or data logger. A zero is shown on the left, while a one is shown on the right. The period for both of these pulses is 4 ms and the two are distinguished by the amount of time the pulse is low each cycle.

The method for receiving a bit of data from the master is as follows. The microcontroller starts a timer on the falling edge of each pulse. As soon as the power line transitions from low to high, the microcontroller stops the timer. If the timer value corresponds to a time interval less than one and a half milliseconds, then a binary zero was sent. Otherwise, the microcontroller has received a binary one.

Now that the sensor can read a single bit at a time, this functionality can be extended to read a byte of data at a time. The communication protocol used to send a byte of information is very similar to serial communication. Eight bits are sent in series starting with the least significant bit first. No start bit is needed, as the sensor microcontroller starts receiving as soon as it senses a falling edge transition. Also, no parity bit it used, although it would be simple to add this. No stop bit is needed, as the sensor microcontroller knows the byte ends after the eighth bit is received.

The current sensor implementation expects two bytes of data from the master. The first byte corresponds to the slope and the second byte represents the temperature offset. The slope is typically between 0.80 and 1.20 based on experimental data, and a precision of two decimals is acceptable for this application. Therefore, the slope byte is formatted as the slope multiplied by 100. This means the sensor will likely receive a decimal value between 80 and 120, or equivalently a hexadecimal value between 0x50 and 0x78. The temperature offset byte is formatted as an 8-bit signed integer, which is valid for the range -128℃ to $+127$℃. This range is much larger than necessary, since the temperature should be within ±10℃ according to the ATtiny85V's datasheet. A precision of 1℃ is acceptable since the internal temperature sensor can only measure to the nearest degree Celsius.

After the sensor has received these two bytes for calibration, the microcontroller writes them to EEPROM. The method used to store the calibration data in EEPROM is discussed in detail later. After the calibration data is stored, the sensor begins its normal operation by computing its calibrated temperature and sending it to the master. Every time the sensor is powered and there is no data received from the master, it reads its slope and temperature offset from EEPROM. The sensor then computes its calibrated temperature from the slope and offset and sends the pulse width modulated wave corresponding to this temperature.

Summary

An overview of the proposed system for a two wire temperature sensor was discussed in detail. The hardware configuration and operation for a low component count sensor was discussed. The concept of parasitic power for powering the sensor from the data line was explained. The use of pulse width modulation to transmit temperature data was explored. Finally, the use of bidirectional communication for sending calibration information to the sensor was discussed in detail. With an overview of the system and an understanding of the key concepts, analysis and implementation of the basic hardware can be discussed, as well as the problem of calibration. The following chapters will address these topics in more detail.

CHAPTER III

ANALYSIS

This chapter contains an analysis of the detailed hardware design for a low component count temperature sensor. The advantages and disadvantages of different types of capacitors will be discussed, as well as the type of capacitor to use for local energy storage. The size of the storage capacitor necessary will be calculated. The hardware configuration for a minimal component sensor will be discussed in detail. Finally, a mathematical analysis will be performed to determine the impact that cable length has on the sensor output.

Capacitor Type

Several different types of capacitors are available on the market, and each type has advantages over others depending on the application. Some of the more popular types of capacitors include electrolytic, ceramic, and tantalum. Although many other types of capacitors exist, they are not suited for this application due to size, cost, stability, and so on. Some advantages and disadvantages to using electrolytic, ceramic, and tantalum capacitors for this application are discussed below. See [7] for more details on these different types of capacitors.

Aluminum electrolytic capacitors, commonly referred to as electrolytic capacitors, are one of the most popular high capacitance capacitors. They are popular not only because of their low cost, but they also have a high volumetric density. Electrolytic capacitors provide a large capacitance in a relatively small package. However, general purpose aluminum electrolytic capacitors can be problematic at very low temperatures because the electrolyte may freeze. Aluminum electrolytic capacitors also have poor temperature coefficients. This means their capacitance changes significantly with changes in temperature. Even aluminum electrolytic capacitors that are rated from $-40°\mathrm{C}$ to $+105°\mathrm{C}$ can cause problems, since at lower temperatures their capacitance can drop to 60 percent of their nominal value. This decrease in capacitance could cause the sensor to stop operating at colder temperatures.

Ceramic capacitors are available in three different classes. Class I ceramic capacitors are desirable due to their accuracy, as well as temperature-compensation. Although they have the lowest losses out of the three classes, they also have the lowest volumetric efficiency. Class II ceramic capacitors have better volumetric efficiency than class I capacitors, but they also have lower accuracy and stability. Finally, class III ceramic capacitors are desirable because of their high volumetric efficiency. However, they suffer from low accuracy and stability. Since ceramic capacitors trade off volumetric efficiency for stability, they will not work well for this application. The storage capacitor needs to have a high volumetric efficiency, as well as temperature stability in order to operate reliably.

Tantalum electrolytic capacitors, commonly referred to as tantalum capacitors, are an alternative to aluminum electrolytic capacitors. Tantalum capacitors are more stable than aluminum capacitors and have a lower effective series resistance. They also have a very high volumetric ratio, which means a tantalum capacitor that is the same size as an aluminum capacitor can have a higher capacitance. Compared to aluminum electrolytic capacitors, solid tantalum capacitors show much less variation in capacitance with temperature. The biggest disadvantage of using a tantalum capacitor is that they are expensive compared to aluminum capacitors. However, the capacitor is the most important component, as the sensor will not operate without it, so the extra cost is acceptable given the increase in performance.

A good choice for the storage capacitor is a high quality solid tantalum surface mount capacitor. Since there is fierce competition among manufacturers for market share of the 5 V power supply market, there is a large selection of 6.3 V rated solid tantalum capacitors on the market. A tantalum capacitor provides a large capacity in a relatively small package. It also has a low effective series resistance and good temperature stability, which is very important in this application.

Size of Storage Capacitor

Choosing the correct size for the storage capacitor is very important. If it is not the right size, it will not keep the microcontroller powered while the power line is low. In order for the sensor to operate, the parasitic power must work flawlessly. As discussed in the previous section, temperature has a significant effect on the capacity of the capacitor. For a tantalum capacitor, the capacity may vary $\pm 20\%$ from the capacitor's nominal value. This must be accounted for when calculating the size of the capacitor, so that the sensor still operates in the worst case scenario.

When the sensor microcontroller activates the pull-down resistor R_2, capacitor C_1 powers the microcontroller and the voltage across the capacitor drops. It should be noted that the current consumption of the microcontroller decreases as V_{CC} decreases. This means the capacitor will perform better than these simple calculations imply. However, assuming the microcontroller draws a constant current I_{supply} when it discharges C_1, the voltage drop can be estimated during a time interval Δt as follows

$$i(t) = C\frac{dV_C(t)}{dt} \Rightarrow I_{supply} = C_1\frac{\Delta V_C}{\Delta t}$$

so that

$$C_1 = \frac{(I_{supply})\Delta t}{\Delta V_C}$$

As explained before, the period of the output square wave from the sensor is $T = 400\,ms$. The worst case from the standpoint of keeping the microcontroller powered from C_1 is when the temperature is -40°C. This corresponds to an ADC value of 230, which results in an on time of 230 ms. Therefore, the worst case off time $\Delta t = 170$ ms. As we described above, the sensor microcontroller generates at least two, but in general n cycles, so C_1 needs to power the sensor microcontroller for a time nT. In the worst case scenario the capacitor powers the microcontroller for 170 ms, and it charges when the power line is high for 230 ms each cycle. Therefore, the capacitor has time to fully charge each cycle and the number of cycles has no affect on the calculation for capacitor size.

$$C_1 = \frac{(I_{supply})\Delta t}{V_{CC} - V_D - V_{min}}$$

The equation above can be used to determine a proper value for C_1 as follows. Assuming a worst case forward voltage drop of $V_D = 0.4$ V for the Schottky diodes, and a 5 V system where the master powers the sensor with a voltage very close to 5 V, the sensor microcontroller's V_{CC} pin, namely V_C will be at 4.6 V when C_1 is fully charged. The minimum voltage the microcontroller will operate at is $V_{min} = 1.8$ V. Therefore, the maximum allowable voltage drop across C_1 is $\Delta V_C = V_C - V_{min} = 2.8$ V. Let $\Delta t =$ 170 ms, and allow for a voltage drop $\Delta \mathrm{V_C} = 2.8$ V. From the ATtiny85V datasheet it follows that at a clock of 1 MHz, the part consumes about 1.20 mA. Substituting these values into the equation above gives $C_1 = 73$ µF. For safety, use $C_1 = 100\ \mu$F. There

Table 3. Minimum required storage capacitor values

Clock Frequency	I_{supply}	**Period (T)**	C_1
128 kHz	180 µA	200 ms	5.5 µF
128 kHz	180 µA	400 ms	11 µF
1 MHz	1.2 mA	200 ms	36 µF
1 MHz	1.2 mA	400 ms	73 µF
8 MHz	5.0 mA	200 ms	152 µF
8 MHz	5.0 mA	400 ms	304 µF

are several small 6.3 V tantalum capacitors in a 1206 package available on the market. One can reduce the value further by decreasing T or reducing the microcontroller clock frequency. Table 3 summarizes other possible values. The nominal capacitance values listed do not take into account variations in capacitance due to temperature.

In order to verify the basic hardware design used for parasitic power, a simulation was performed in Micro-Cap SPICE. The circuit shown in Figure 7 was used to model the basic parasitic power hardware used on the sensor. The voltage source, V_{CC}, simulates the worst case scenario when the sensor's temperature is -40℃. This results in an on time of 230 ms and an off time of 170 ms. The tantalum capacitor used in the design is modeled using a 100 μF capacitor in series with a 500 mΩ resistor representing the capacitor's effective series resistance, or ESR. Finally, a constant current supply is used to simulate the current drawn by the microcontroller.

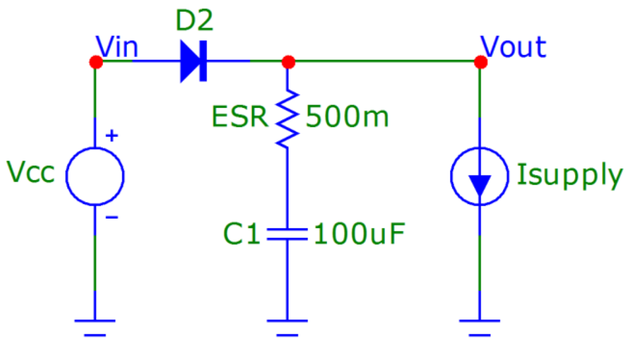


Figure 7. Circuit model used for parasitic power simulations in Micro-Cap SPICE. The voltage source, Vcc, models the case when the sensor's temperature is -40℃, which corresponds to the worst case scenario. The current supply models the constant current the microcontroller draws. The capacitor is a 100 μF tantalum capacitor with 500 mΩ ESR, which has the same specifications as the one used in the temperature sensor.

The results of the Micro-Cap SPICE simulation are shown below in Figure 8. The voltage generated by the voltage source, V_{CC}, is shown in gray and the voltage supplied by the capacitor is shown in black. These two voltages correspond to the nodes labeled V_{in} and V_{out} in the circuit diagram above. At time $t = 0$ ms when the sensor is first powered on, the capacitor begins charging from the power line. As soon as the power line is pulled low at time $t = 230$ ms the capacitor begins powering the microcontroller, which is modeled by the constant current source. The capacitor voltage drops to around 2.7 V before the power line is pulled high again at time $t = 400$ ms. This implies that the capacitor is able to provide power to the microcontroller for a duration of 170 ms, which represents the worst case scenario.

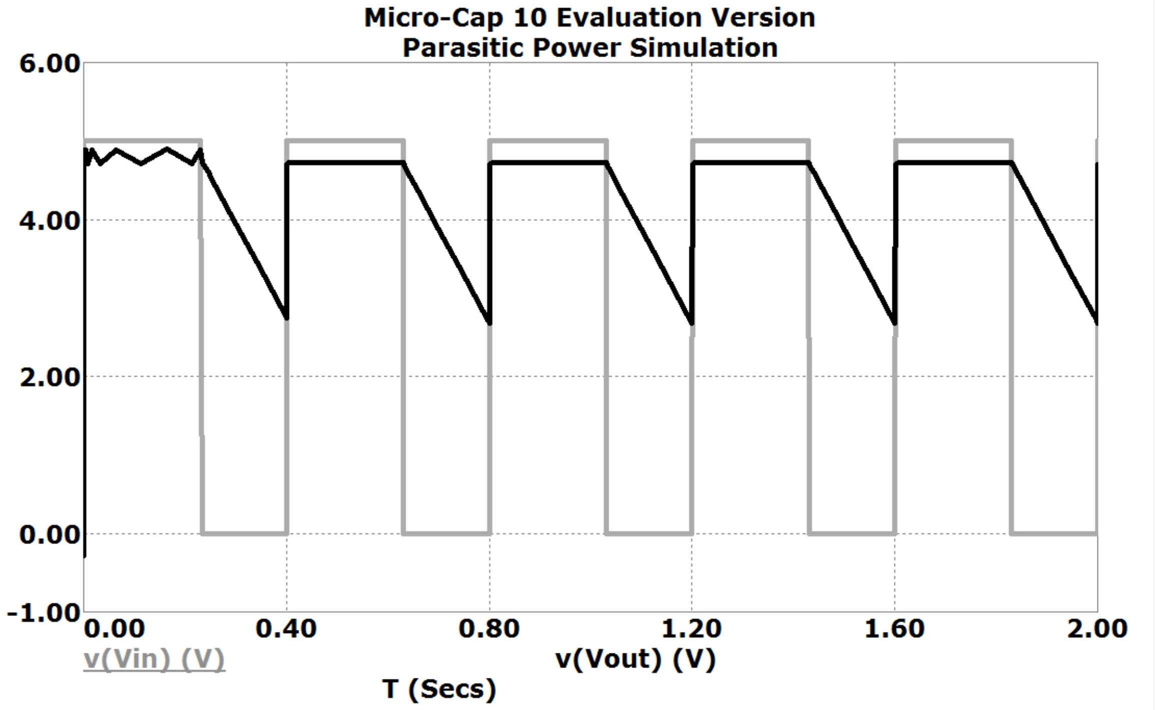


Figure 8. Micro-Cap SPICE simulation for the parasitic power circuit model shown above. The voltage from the voltage source is shown in gray and the voltage supplied by the capacitor is shown in black. The capacitor is able to supply more than the necessary 1.8 V to keep the microcontroller powered when the power line is low.

After deriving an equation to calculate the necessary capacitor value to use for parasitic power and computing what the required capacitance should be, a 100 µF capacitor was selected. These calculations took into account the worst case scenario and also added some margin for safety. The parasitic power circuit was then successfully simulated using this value, verifying that this capacitance is sufficient. Based on these results, a 100 µF tantalum capacitor will be used in the design.

Minimal Component Sensor

The ultimate goal in the development of this sensor is to reduce the number of components, and therefore the overall cost. Also, the fewer the number of parts, the easier it is to build the sensors. In an attempt to minimize the number of components in the design, the ATtiny85V's internal temperature sensor will replace the DS18B20 temperature sensor IC. By removing this IC, the component count is decreased by two, since the pull-up resistor is no longer required. This new minimal component hardware design is discussed in detail below.

The ATtiny85V microcontroller has a built-in 10-bit analog to digital converter, or ADC, as well as an integrated temperature sensor. The microcontroller can be configured to digitize the temperature sensor voltage. It can also be configured so that the built-in ADC converter obtains its voltage reference from an external voltage reference, or one of two internal bandgap references. These on-chip internal voltage references and temperature sensor provide a minimal component sensor, depicted in Figure 9. The circuit is identical to the circuit in Figure 3, but without the external DS18B20 digital temperature IC. Instead, the ATtiny85V uses its internal temperature sensor for making temperature measurements. An additional pin on the microcontroller is used for bidirectional communication, which was discussed previously.

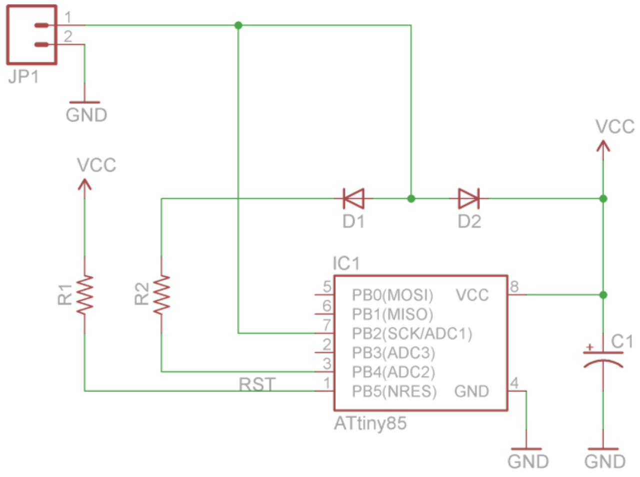


Figure 9. Hardware schematic for minimal component count temperature sensor with only six required components. The master supplies power to the microcontroller over the power line. The microcontroller measures the temperature using its on-chip sensor and transmits a square wave with a duty cycle proportional to this temperature. The sensor continues to send temperature measurements until the master removes power. An additional pin on the microcontroller is used for bidirectional communication.

The microcontroller is programmed to use its internal 1.1 V voltage reference for A/D conversions, and it routes the on-chip temperature sensor to the ADC4 channel. The value from the ADC must then be converted to a corresponding temperature. The following equation is provided in [1] for computing the temperature based on the two bytes returned from the ADC

$$T = k\big((ADCH \ll 8)|ADCL\big) + T_{OS}$$

In this equation, $ADCH$ and $ADCL$ are the high and low ADC channel data registers, respectively. The 10-bit ADC conversion result is simply $(ADCH \ll 8)|ADCL$, where k is a fixed slope coefficient and T_{OS} is the temperature offset. Manufacturing process variations make k and T_{OS} variable across different ATtiny85V chips and the documentation provides little additional information. However, semiconductor–based temperature ICs typically use a forward-biased diode as the basic sensing element, as described previously. This implies that the voltage across the diode changes linearly with changes in temperature.

The ATtiny85V datasheet also states that the slope k is close to one, which is typically consistent with this approach. It is not clear from the datasheet what causes the temperature offset, T_{OS}, however, it does state that the 1.1 V internal bandgap reference voltage is in the range from 1.0 V to 1.2 V. Since this voltage is used as the reference for the ADC when the on-chip temperature sensor is used, the reference's uncertainty directly affects the temperature measurement. A method for determining the actual internal bandgap voltage is discussed later. A few methods for calibrating the temperature sensor to account for these variations in slope and temperature offset are also discussed below.

Cable Analysis

There are many things to consider when analyzing the transmission of data across long wires. A very common approach is to model the wires as a transmission line that has a series resistance and inductance, as well as capacitance between the wires. For this application, only two wires are used, which greatly simplifies the analysis. Since the series resistance of the wire used is only a few Ohms for a distance of 50 meters, its effect on the analysis is negligible compared to the pull-up resistor. Since the data rate is so low and the inductance of the wire used is small, its effect is also negligible. Therefore, only the capacitance between the two wires needs to be taken into account. This capacitance will be modeled as a function of cable length using the following equation from [21], where the units of capacitance are measured in farads per meter.

$$\mathrm{C} = \frac{\pi\varepsilon}{\cosh^{-1}\frac{s}{d}}$$

In this equation, ε is the permittivity and is equal to $\varepsilon_r\varepsilon_0$ where ε_r is the relative permittivity of the material and ε_0 is the vacuum permittivity. The vacuum permittivity is approximately equal to $8.85 \times 10^{-12}\ \mathrm{F/m}$. The relative permittivity of air is very close to one, while the relative permittivity of the Teflon coating around the wire is 2.1. The diameter of the wire is represented by d, and s is the separation distance between the centers of the two wires.

For this application, 22 AWG Teflon coated wire is used, where the overall diameter of the coated wire is approximately 0.050 inches. The thickness of the Teflon insulation is about 0.010 inches. This means the diameter of the wire is approximately 0.030 inches. For a tightly twisted pair of 22 AWG Teflon wire, the separation distance, s, is about 0.050 inches. Using these values in the equation above, the capacitance of the twisted pair is approximately equal to $53\ \mathrm{pF/m}$. This results in a capacitance of around 3 nF for a cable length of 50 meters. This capacitance is negligible compared to the 100 μF storage capacitor in parallel, so it can be ignored for the range of cable lengths used in this application. The cable length would have to be over 18 kilometers for the capacitance to be 1 μF.

The next thing to consider in the analysis of the twisted pair is the time constant, as this affects the accuracy of the temperature sensor. As discussed previously, the resistance, inductance, and capacitance of the cable have a negligible effect for the cable length used in this application. By ignoring these, the calculations for the time constant are significantly simplified. When the sensor's power line is high and the storage capacitor is charging, the effective circuit looks like the circuit shown in Figure 10. The 1 kΩ pull-up resistor is used to sense the square wave from the temperature sensor. The diode and capacitor model the parasitic power circuit, as explained previously. Finally, the constant current supply models the current drawn by the microcontroller.

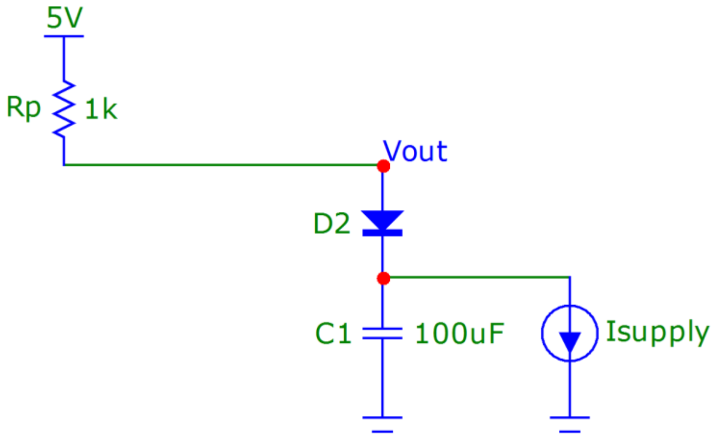


Figure 10. Effective circuit diagram for the temperature sensor when the storage capacitor is charging. There is a 1 kΩ pull-up resistor used by the master to sense the square wave from the temperature sensor. The constant current supply models the current drawn by the microcontroller.

The diode in the circuit above is forward-biased in this configuration, so it simply drops a small voltage across it. This voltage drop can be subtracted from the 5 V source, and it can be removed from the circuit. The current supply can also be removed from the circuit in order to compute the time constant. This leaves only the pull-up resistor and storage capacitor in the circuit, which results a time constant

$$\tau = R_p C_1$$

For this application, the pull-up resistance is 1 $k\Omega$ and the storage capacitance is 100 μF. These values result in a time constant, τ, of 100 milliseconds. This means that it will take approximately 100 milliseconds to charge the storage capacitor to 63 percent of its full charge. Note that the time constant will vary slightly for different cable lengths, however, as shown above, the resistance and capacitance of the cable have little effect on the overall resistance and capacitance of the circuit for the cable lengths used in this application.

One final thing that needs to be taken into account are the values of the pull-up and pull-down resistors. In order for the master to be able to read the square wave from the temperature sensor, the power line must be pulled below 2.4 V for the master to detect a zero and it must be pulled above 2.6 V for the master to detect a one. When the sensor pulls the power line low, the effective circuit looks like the circuit shown below in Figure 11. The 1 kΩ pull-up resistor is used to sense the square wave from the temperature sensor. The 330 Ω pull-down resistor is used by the sensor to pull the power line low, while also limiting the current drawn by the sensor.

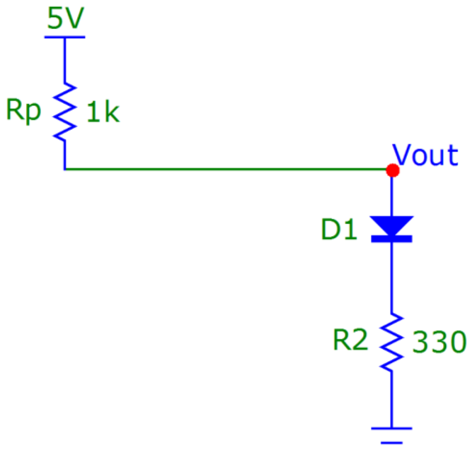


Figure 11. Effective circuit diagram for the temperature sensor when the sensor is pulling the power line low. There is a 1 kΩ pull-up resistor used by the master to sense the square wave from the temperature sensor. There is also a 330 Ω pull-down resistor used by the sensor to pull the power line low.

The value of the pull-up resistor determines the voltage level the sensor will be able to pull the power line down to, since the pull-down resistor value on the sensor is fixed. Assuming the diode drops a maximum voltage of 0.4 V, the voltage dropped across the two resistors is equal to 4.6 V. This results in a current equal to 4.6 V divided by the total series resistance of the two resistors. In order to keep the power line below the 2.4 V threshold for the master, the following equation must hold

$$5\ \mathrm{V} - \left(\frac{4.6\ \mathrm{V}}{\mathrm{R_p} + \mathrm{R_2}}\right) R_p < 2.4\ \mathrm{V}$$

After substituting 330 Ω for R_2 and solving the inequality for R_p, the pull-up resistance must be greater than 430 Ω for the master to be able to detect the power line is low. As stated above, the power line must also remain above 2.6 V for the master to be able to detect the power line is high. This means that as soon as the sensor releases the power line from pulling it low, it must transition above this threshold for the master to accurately measure the sensor’s temperature. Assuming the maximum voltage drop of 0.4 V across the diode in Figure 10, the voltage stored on the capacitor must remain above 2.2 V for the sensor to be completely accurate. This implies the capacitor voltage should not drop below 2.2 V during the time the sensor is pulling the power line low.

Assuming that the parasitic power circuit is designed correctly and the capacitor voltage is greater than 2.2 V at the end of the cycle where the sensor is pulling the power line low, the voltage on the line detected by the master should be above the threshold. The current drawn by the temperature sensor in the worst possible case is 1.2 mA. In order to keep the power line above the 2.6 V threshold for the master, the following equation must hold

$$5\ \mathrm{V} - (1.2\ \mathrm{mA})R_p > 2.6\ \mathrm{V}$$

Solving for R_p in this inequality, the pull-up resistance must be less than 2 kΩ for the master to be able to detect the power line is high. Therefore, the allowable range of pull-up resistors for the master to be able to accurately calculate the sensor’s temperature is from 430 Ω to 2 kΩ. This range was calculated using several simplifying assumptions, and therefore the pull-up resistor value should be chosen close to the middle of this range to ensure proper operation. As mentioned before, a 1 kΩ pull-up resistor is used for this application, as it provides a large safety margin on both sides of this range.

Summary

A detailed analysis of the hardware design for a low component count temperature sensor was carried out. Different types of capacitors were introduced, along with their advantages and disadvantages. The selection of the type and size of capacitor to use for local energy storage was explained. The hardware configuration for a minimal component temperature sensor was discussed in detail. Finally, an analysis was performed to determine the impact that cable length has on the sensor output. The remaining chapters will focus on the implementation and calibration of the minimal component sensor discussed here.

CHAPTER IV

HARDWARE IMPLEMENTATION

In this chapter, the hardware implementations of the two-wire temperature sensor will be described. The implementation of the prototype sensor will be discussed briefly before the production version is analyzed in more detail. The development of a junction box for reading the temperature from multiple sensors will be explained. Finally, the process of sealing the sensors will be described in detail.

Prototype Temperature Sensor

The original prototype version of the two-wire temperature sensor is shown below in Figure 12. The dimensions of the printed circuit board are 0.8125 inches wide by 1.25 inches long. The prototype PCB has nine components on it not including the cable. An ATtiny85V microcontroller is located at the center of the board. Just above the microcontroller is a 6-pin header used for programming the ATtiny85V. The DS18B20 digital temperature sensor IC is located along the top of the board. This IC provides temperature measurements for the microcontroller to send using pulse width modulation. There are two surface mount tantalum capacitors on the right side of the board that are used for parasitically powering the microcontroller. Two of the three resistors on the left are pull-up resistors and the third one is the pull-down resistor for pulling the power line low. Finally, there is a two common-anode diode package on the lower left side.

As stated above, the prototype temperature sensor uses the DS18B20 digital temperature sensor IC for making temperature measurements. This method worked fine most of the time, however, at very cold temperatures, below 0℃, the sensor no longer functioned properly. There are a couple of possible reasons it would not work correctly. First, since the capacitance of the tantalum capacitor drops 10 to 20 percent of its nominal capacitance at room temperature, the capacitor may not be storing enough energy when the sensor pulls its power line low. After doubling the capacitance by

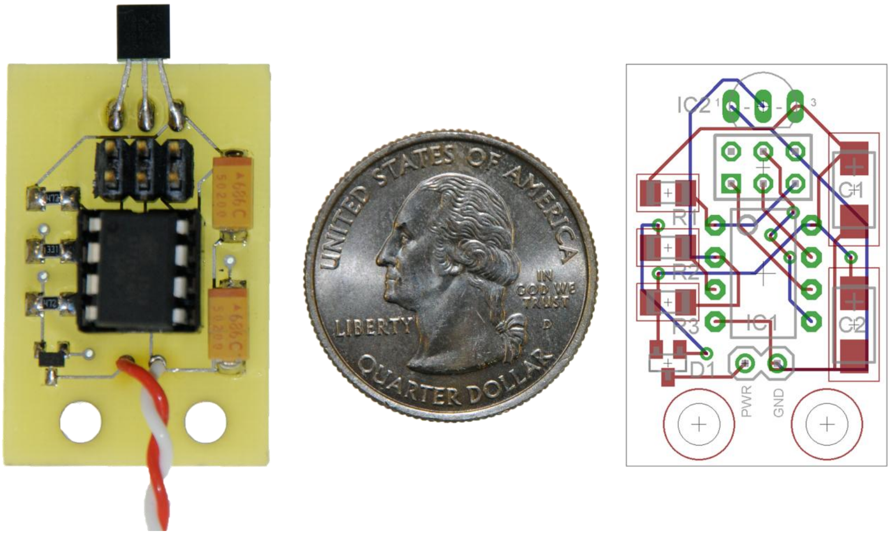


Figure 12. Photo of prototype temperature sensor with printed circuit board. There are two capacitors along the right side, an ATtiny85V in the center, a six pin programming header just above the microcontroller, a DS18B20 temperature sensor along the top, a two common-anode diode package on the bottom left, and three resistors along the left side. The twisted pair supplies ground along the white wire and power/communication on the red wire.

placing an equal valued capacitor in parallel with the first one, this problem still occurred. The other possible issue could be that at cold temperatures, the timing of the 1-Wire communication line is no longer within the allowed range of the DS18B20. If this is true, the sensor becomes useless at cold temperatures. The next version of the temperature sensor does not use the DS18B20, so this is no longer an issue.

Production Temperature Sensor

The production version of the temperature sensor uses the ATtiny85V's internal temperature sensor, thus eliminating the need for the DS18B20 and reducing the component count by two. A pull-up resistor used on the DS18B20's data line was the second component removed. Also, only one capacitor is used, since the power consumption was reduced significantly without the DS18B20. The production version worked well over the entire temperature range. The one issue is that each sensor measured a different temperature. According to the ATtiny85V documentation, the temperature sensor will read within ±10℃ if it is not calibrated. This stems from the manufacturing process variations in the semiconductor junction used for measuring temperature. In order to provide accurate temperature measurements, the internal temperature sensor will need to be calibrated. Methods for calibrating the sensor are discussed later.

The production version of the two-wire temperature sensor is shown in Figure 13. The dimensions of the printed circuit board are 0.2725 inches wide by 0.6 inches long. The PCB only has six components on it, not including the cables. An ATtiny85V microcontroller is located in the center of the top side of the board. The surface mount tantalum capacitor along the top of the board is used for parasitically powering the microcontroller. There are also two resistors and a two common-anode diode package on the bottom side of the board. One of the resistors is a pull-up resistor on the reset pin of

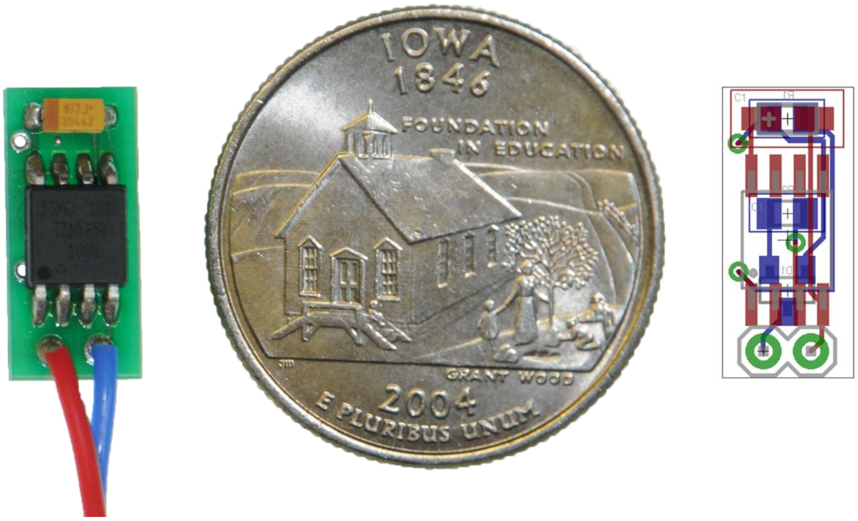


Figure 13. Photo of production temperature sensor with printed circuit board. There is one capacitor along the top, an ATtiny85V in the center, a two common-anode diode package and two resistors on the back. The twisted pair supplies ground along the blue wire and power/communication on the red wire.

the microcontroller, and the other resistor is the pull-down resistor for pulling the power line low during communication.

The 8-pin dual inline package (DIP) microcontroller was replaced with an 8-pin small-outline integrated circuit (SOIC) package. This significantly reduced the size of the sensor, however, since the 6-pin programming header was removed to save space, it is now much more difficult to program the microcontroller. Two different 8-pin SOIC to 8-pin DIP programming adapters, shown in Figure 14, can be used to program the ATtiny85V. After the ATtiny85V has been programmed, it is soldered to the temperature sensor board, making it difficult to reprogram.

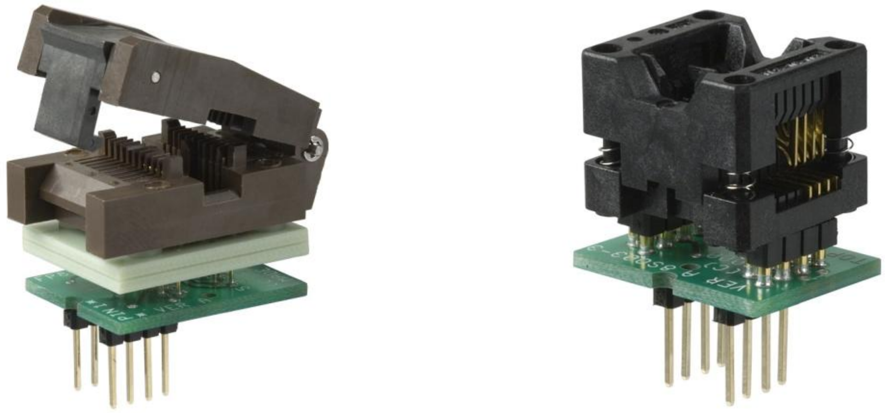


Figure 14. Two different styles of 8-pin SOIC to 8-pin DIP programming adapters from Logical Systems Corporation. These are used for programming the ATtiny85V microcontroller before soldering it to the temperature sensor PCB.

A significant amount of time went into reducing the size of the temperature sensor. Smaller packages were used for the microcontroller and storage capacitor to reduce the size. Also, components were placed on both sides of the PCB, unlike the prototype sensor where there were only parts on the top side of the board. This re-design helped eliminate a significant amount of wasted space from the prototype board. Although all of the components come in smaller packages, any further reduction in size would be marginal. The boards can be sent out to a company for assembly, and since all of the components are surface mount, it would be much cheaper than the prototype version. It is also cheaper to seal these sensors, as it requires less heat shrink or epoxy to be used than on the prototype sensor.

Sensor Junction Box

In order to read the temperatures from all of the sensors, a junction box was developed. The junction box provides the data logger with a simple, RS-485 interface for reading the temperature sensors. Figure 15 shows the inside of the junction box. There is a temperature junction board along the left, which is described in detail below. Along the right is a soil moisture junction board developed in [15] for reading four soil moisture

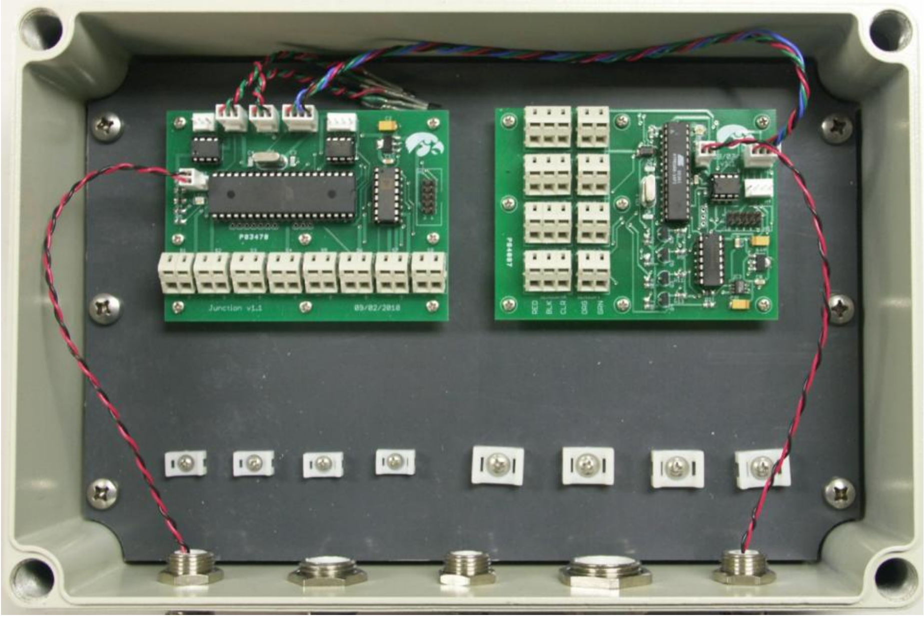


Figure 15. Junction box used for connecting eight soil temperature sensors and four soil moisture probes. The junction box communicates with the data logger using RS-485. The cables for these sensors and the cable from the data logger are run through the cord grips along the bottom. There are two LEDs, one for each board, that are used to show the system is functioning correctly. Eight cable tie mounts along the bottom provide strain relief for the cables.

probes. There are eight cable tie mounts along bottom that are used to provide strain relief for the sensor cables. Each of the boards is connected to an external LED that provides confirmation that the system is functioning correctly. The sensor cables are fed through the cord grips along the bottom, which are then tightened to hold them in place. The aluminum enclosure provides a weather tight seal to protect the electronics from the elements. A cable is also run through the center cord grip that connects to the data logger. This cable provides RS-485 communication, as well as power to the junction box.

A master microcontroller is needed to read the temperature from the sensor, and convert this from a duty cycle to its corresponding temperature. Figure 16 shows the master microcontroller board that is able to read eight different temperature sensors and

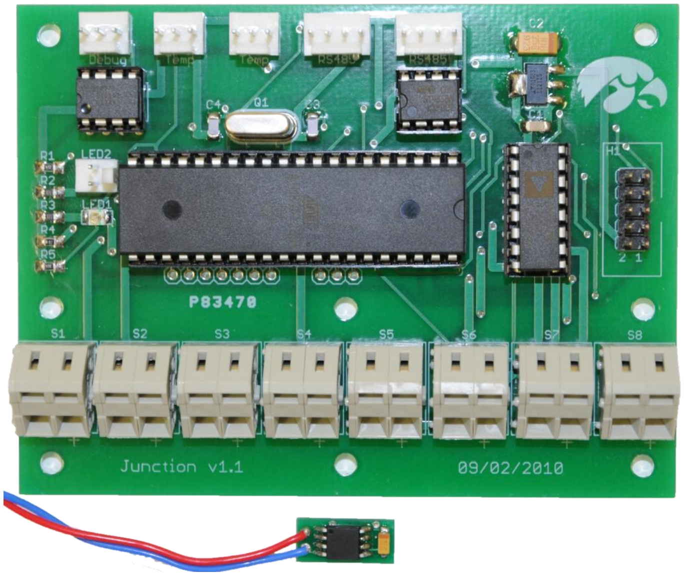
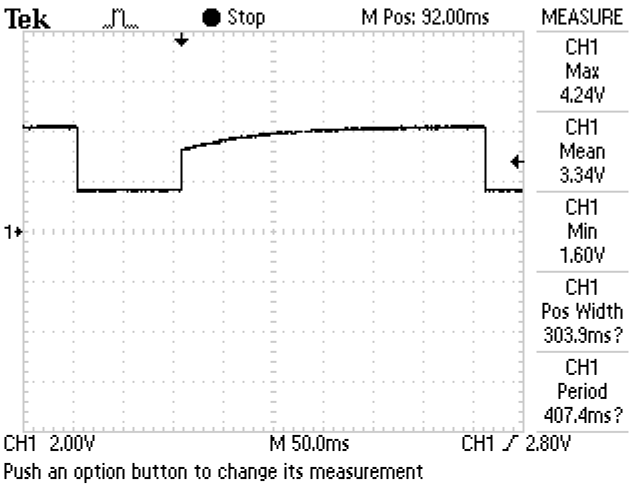


Figure 16. Photo of temperature junction board with temperature sensor. The large IC in the center is an ATmega1284P microcontroller, the chip in the upper left is an RS-232 driver for debugging, the IC above the top right side of the microcontroller is an RS-485 driver for communication, and the chip to the right of the microcontroller is an 8:1 analog multiplexer. The eight connectors along the bottom are for the eight temperature sensors, the header on the right is for programming, and the connectors along the top are for communication.

relay the data over a RS-485 bus to a data logger. This master microcontroller waits for the command to read the temperature sensors. Once this command is received, it starts by reading the air temperature from two DS18B20 temperature sensors that plug into the headers at the top of the board labeled Temp. After reading these temperatures, the master is ready to begin reading up to eight of the soil moisture temperature sensors.

It is relatively easy to determine if a working temperature sensor is attached to each connector since the sensor simply outputs a square wave. By sensing the changes in current, the master will either see changes on its input line if a functioning sensor is attached, or it will not see any changes if there is nothing attached. A simple resistor pulled up to V_{CC} is used to detect the square wave current pulses from the temperature sensor. The master microcontroller will timeout after one second if there are not any changes on its input pin and move on to the next temperature sensor. To switch between the eight possible temperature sensors, a multiplexer is used. This multiplexer simply connects one of eight pins to the input of the microcontroller based on the address the microcontroller selects using three address pins. If a working temperature sensor is attached to a connector, the master microcontroller can perform any operations on the temperatures it gets back from the sensor, including averaging. This makes it possible to get a consistent temperature from each of the sensors.

An example of the temperature sensor's output waveform is shown below in Figure 17. The measured temperature is about +23 degrees Celsius, when the room temperature is actually +23.7 degrees Celsius. The temperature sensor's power line is initially around 4.24 V while the sensor is measuring its temperature. Some of the voltage from the 5 V powering the master is dropped across its pull up resistor from the current the sensor is drawing. During this time, the sensors storage capacitor is charged so that it can provide power to the microcontroller when the power line is pulled low. Once the sensor has measured its current it starts with the power line high for about 304 ms as shown on the oscilloscope output. Then the sensor pulls the power line low to around 1.60 V. The reason the power line is not closer to 0 V is because some voltage is dropped across the pull down resistor and another 0.4 V is dropped across a diode. At this time the storage capacitor provides power to the microcontroller. The line is held low for about 103 ms, and then it is pulled high again. This process repeats until the master removes power.



● Stop M Pos: 92.00ms

MEASURE

CH1 Max 4.24V

CH1 Mean 3.34V

CH1 Min 1.60V

CH1 Pos Width 303.9ms?

CH1 Period 407.4ms?

CH1 2.00V M 50.0ms CH1 ∕ 2.80V

Push an option button to change its measurement

Figure 17. Screenshot of output waveform from a temperature sensor displayed on an oscilloscope. The temperature information is contained in the ratio between the on time and off time of the square wave.

The sensor's temperature is contained in the duty cycle of the square wave and can be calculated using its on time and period. First, the master divides the on time of the wave by its period to obtain the duty cycle, $D \sim 0.7460$. Next, it multiplies the duty cycle by 400 to obtain the corresponding ADC value, $ADC = 298$. Now 230 is subtracted and the number is multiplied by $65/70$ to convert the ADC value to a temperature relative to -40°C. Finally, 40°C is subtracted to obtain the sensor's actual temperature of $+23$°C.

Sealing the Sensor

Sealing the temperature sensor is a vital step in keeping the sensor functioning reliably in harsh environments. There are several potential solutions for sealing the sensor, however only two are discussed here. One option is to use two-part epoxy to permanently seal the temperature sensor. The second option is to instead use heat shrink with an adhesive filling to seal the sensor.

A two-part epoxy, such as MG Chemicals 832B encapsulating and potting compound, works well for permanently sealing electronics. Figure 18 below shows a temperature sensor completely sealed in epoxy. This epoxy can protect the sensor from heat, moisture, vibration, impact, electrical shock, and conductivity, among other things that occur frequently in outdoor environments. The epoxy is also non-conductive, so it will not short any of the electronics inside. One major disadvantage to using epoxy is that it is permanent, so once the sensor is sealed, there is no way to update the sensor or reuse components. However, at the same time, the epoxy protects the electronics much better than heat shrink does.

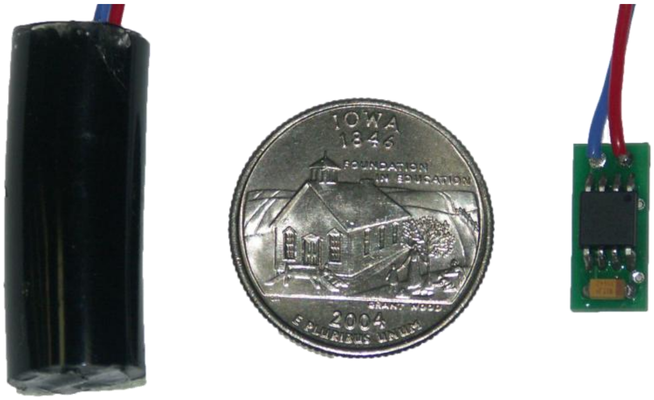


Figure 18. Photo of temperature sensor sealed in epoxy on the left. The temperature sensor on the right is identical to the one sealed inside of the epoxy. The twisted pair supplies ground along the blue wire and power/communication on the red wire.

An alternative to using epoxy to seal the temperature sensor is to use heat shrink to protect it from the elements. By using heat shrink that is lined with adhesive, the sensor can be sealed so that moisture cannot infiltrate it. The advantage to using heat shrink instead of epoxy is that the electronics can be modified or reused by simply cutting off the heat shrink and peeling the adhesive off.

Figure 19 shows a sensor after it has been sealed using heat shrink lined with adhesive. In this case, ¼” by 1 ½” black adhesive lined heat shrink tubing from Power Phase was used. It is rated for a temperature range from −67°F to +230°F and has a shrink ratio of 3 to 1. This means the heat shrink will go from ¼” to 0.080” when heat is applied, resulting in a tight seal around the entire sensor. Both ends of the heat shrink are pushed together while the adhesive is still hot in order to seal the ends as well.

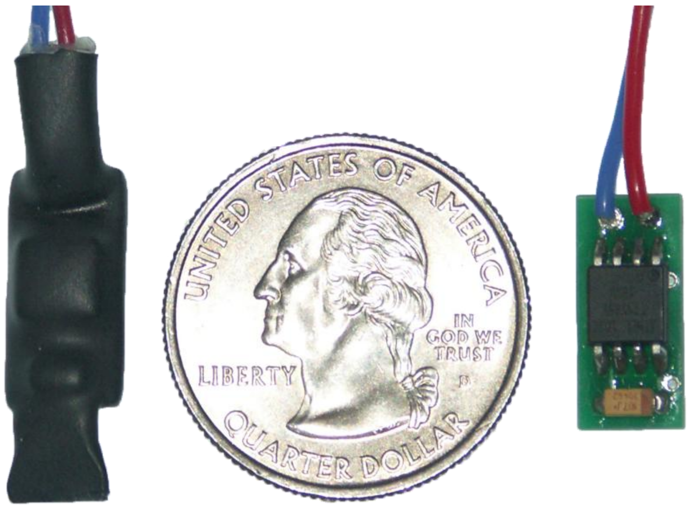


Figure 19. Photo of temperature sensor sealed in heat shrink on the left. The temperature sensor on the right is identical to the one sealed inside of the heat shrink. The twisted pair supplies ground along the blue wire and power/communication on the red wire.

Summary

The hardware implementations of the two-wire temperature sensor were discussed in detail. The implementation of the prototype sensor was discussed briefly before the production version was analyzed in more detail. The development of the junction box for reading the temperature from multiple sensors was explained. Finally, two methods for sealing the sensors were provided. With an understanding of how the hardware was implemented, the problem of calibration can be addressed. The following chapter discusses various methods for calibrating the temperature sensors.

CHAPTER V
CALIBRATION

As mentioned earlier, the ATtiny85V's internal temperature sensor suffers from variations in the manufacturing process, resulting in a significant offset when making a temperature measurement. In order to provide more accurate temperature measurements, the sensor needs to be calibrated. An overview of how the uncalibrated temperature sensors perform will be given before discussing the methods used for calibration. The relationship between the internal bandgap reference voltage and temperature offset will be explored. A simple method for calibrating the sensor based on the temperature offset will be discussed. A method for calibrating the sensor using both a slope and temperature offset will also be explored. Finally, the process of storing calibration data in EEPROM will be explained.

Uncalibrated Sensor Overview

According to the ATtiny85V datasheet there is a linear relationship between actual temperature and its internal temperature sensor output voltage. If this is the case, it is very easy to calibrate the temperature sensor to provide accurate temperature measurements. Only two temperature measurements are required to calibrate the sensor if the relationship is linear. By making two measurements at different temperatures, both the slope and offset of the line can be calculated. The ATtiny85V datasheet suggests that the slope is very close to one, which means only one temperature measurement may be necessary to calibrate the sensor. A few experiments were conducted to verify the relationship is linear.

Figure 20 shows an early experimental test setup to determine the measured temperatures of the uncalibrated temperature sensors. A multiplexer was used to switch between seven sensors and an oscilloscope was used to measure the on time and period of the square wave output. The oscilloscope was setup to average across 128 samples to eliminate any variance between readings, as the output of the temperature sensors vary slightly between measurements. The Omega HH-42 handheld thermistor thermometer was used as the reference. The HH-42 is accurate to 0.01℃ and operates from −20℃ to +130℃. It is also calibrated by the National Institute of Standards and Technology, or NIST.

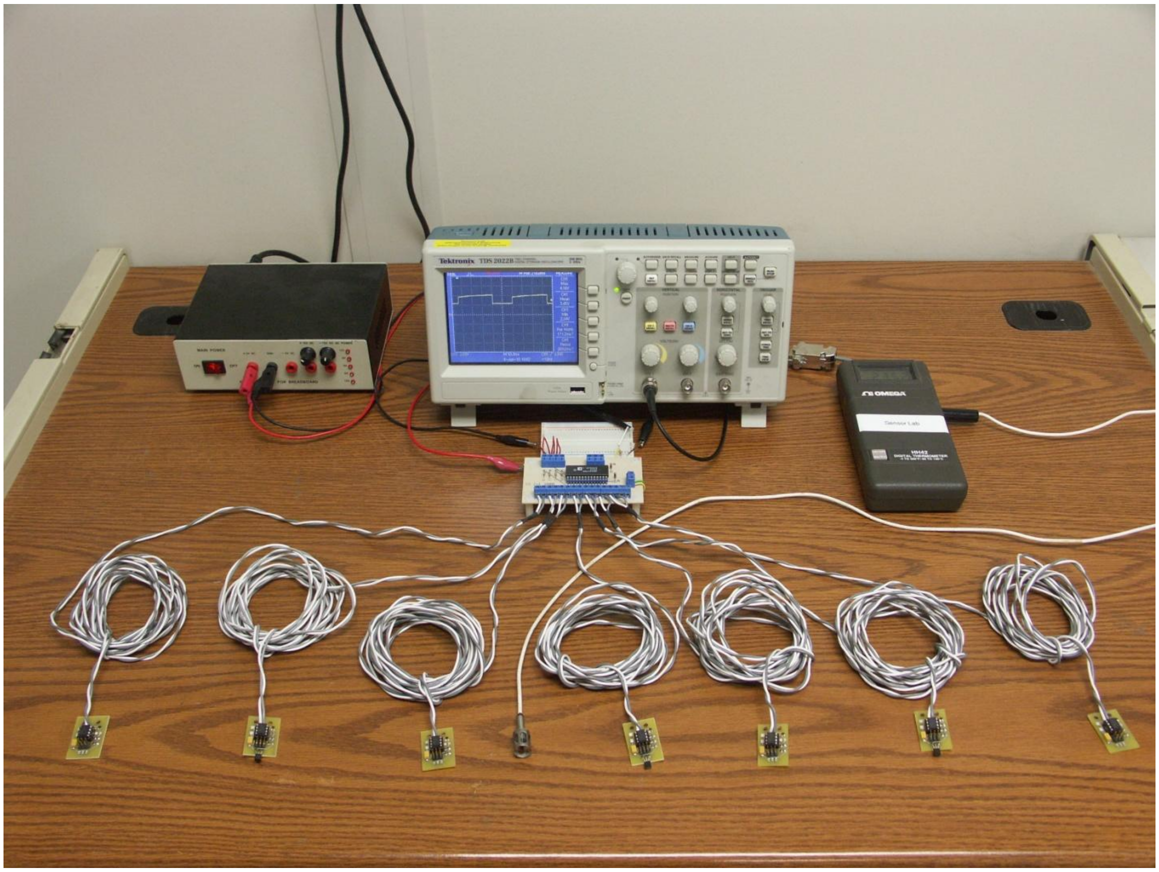


Figure 20. Experimental temperature measurement setup for measuring each sensor's temperature offset. Seven temperature sensors are hooked up to a multiplexer to select which sensor's output is displayed on the oscilloscope. The NIST calibrated temperature sensor on the right is used as the reference.

The experiment consisted of taking temperature measurements from each sensor at three different temperatures across its range. Readings were taken at approximately

$-15°C$, $-5°C$, and $+29°C$. The first two sets of measurements were made with the sensors inside a freezer in the lab. The last set of measurements was made with the sensors at room temperature. These three measurements were taken for each sensor to calculate both the slope and the temperature offset. Three different temperatures were used to verify that the sensor output was indeed linear.

The experimental results are shown below in Figure 21, where the actual temperatures from the HH-42 reference are on the x-axis and the measured temperatures from the sensors are on the y-axis. It is obvious that there is a linear relationship between the actual and measured temperatures, where the temperature offset varies between $-5°C$ and $+15°C$. Also the slope for all of the sensors is very close to one, however most of

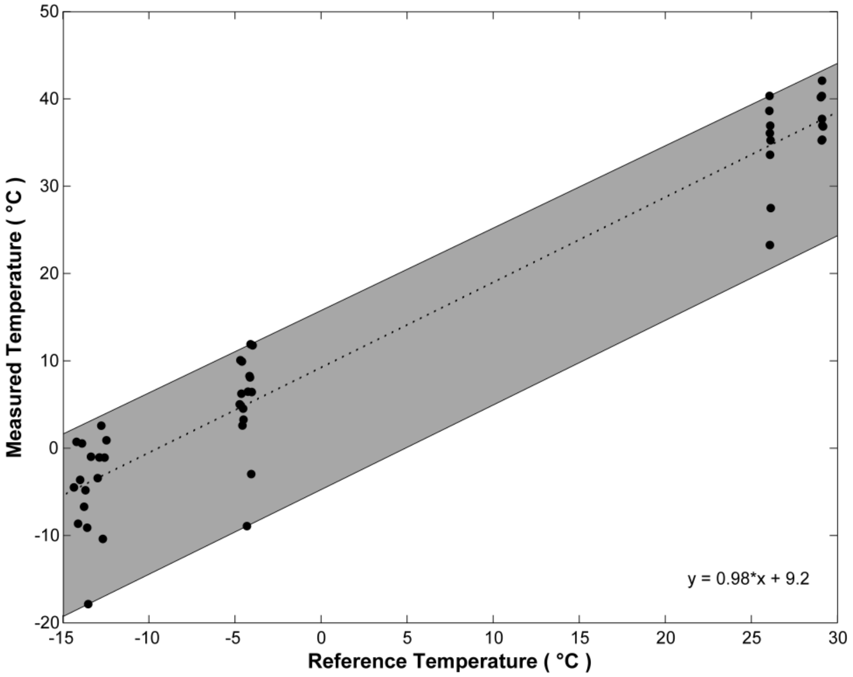


Figure 21. Data collected from 16 temperature sensors across their temperature range. The dotted line indicates the linear regression with the equation shown in the bottom right corner. The shaded region contains all of the temperature measurements and depicts the spread due to each sensor's temperature offset.

the slopes are slightly less than one, as depicted in Figure 22. These results imply that the sensors can be calibrated fairly well by using only a temperature offset, however, the temperature readings will vary more at the high and low extremes.

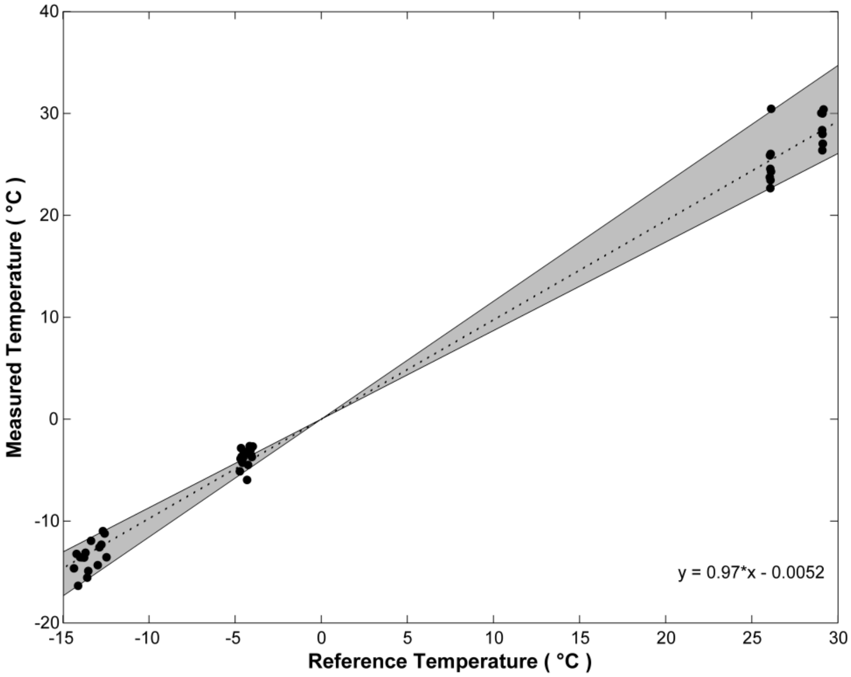


Figure 22. Data collected from 16 temperature sensors across their temperature range after subtracting out each sensor's temperature offset. The dotted line indicates the linear regression with the equation shown in the bottom right corner. The shaded region contains all of the linear regressions for the temperature measurements and depicts the spread due to each sensor's variation in slope.

Table 4 summarizes the slope and temperature offsets for 16 different sensors and Table 5 shows the average, minimum, maximum, and standard deviations. On average, the slope was very close to one, which means a single point calibration can be used for fairly accurate readings. The temperature offset varies quite a bit from sensor to sensor, but the majority of the sensors read a higher temperature than the reference measured.

Table 4. Summary of experimental temperature measurements

Sensor #	Slope	Temperature Offset (°C)
0	1.03	10.15
1	0.91	13.95
2	0.93	9.92
3	1.05	7.71
4	1.05	6.45
5	0.93	8.30
6	0.96	14.11
7	0.98	6.88
8	1.16	−2.95
9	0.93	10.96
10	0.95	11.51
11	1.01	10.90
12	1.00	14.46
13	0.90	10.15
14	0.92	14.89
15	0.87	0.60

Table 5. Statistical summary of experimental temperature measurements

	Slope	Temperature Offset (°C)
Average	0.97	9.25
Max	1.16	14.89
Min	0.87	−2.95
Std Dev	0.074	4.88

An experiment was also setup to determine the thermal time constant of the sealed temperature sensor. The thermal time constant is defined as the amount of time it takes for the sensor to reach 63% of its final temperature. First, the sensor and reference are placed in antifreeze at a temperature of about -12℃. Once the sensor and reference temperatures have reached a constant value, they are both removed from the antifreeze and placed in hot water at approximately +80℃. The temperature sensor and reference are read using MATLAB and the measurements are written to a text file along with the time the measurements were taken. Figure 23 shows the results of the experiment. The thermal time constant for the reference is under a minute, while the thermal time constant for the sensor is approximately two minutes.

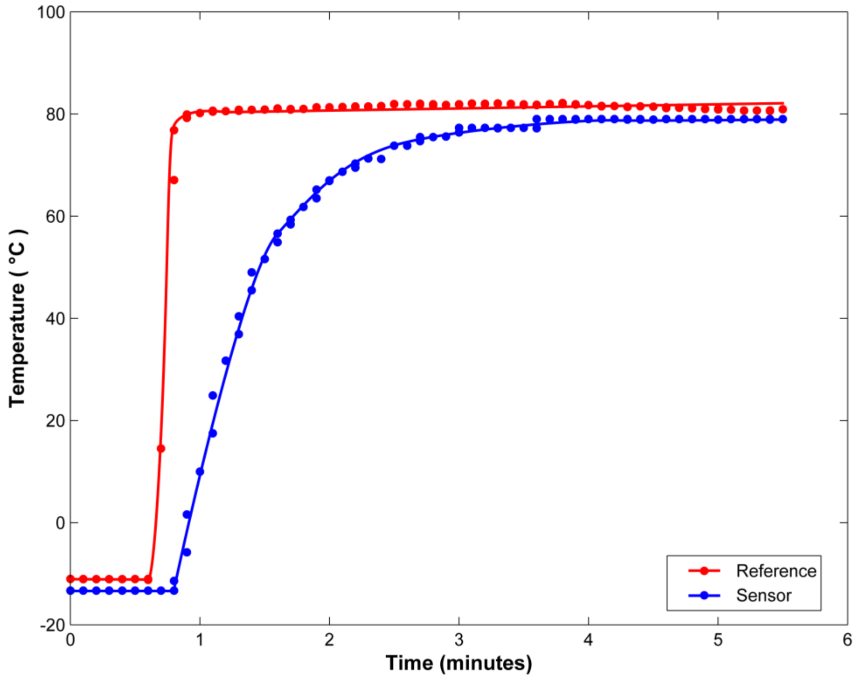


Figure 23. Comparison of thermal time constants for the NIST calibrated reference in red and temperature sensor in blue. The thermal time constant is the amount of time it takes for the sensor to reach 63% of its final temperature. The sensor's thermal time constant is approximately two minutes.

Another experiment was conducted to verify that the relationship between the internal temperature sensor's voltage output and temperature is linear across the entire range from −10℃ to +50℃ . This experiment was adapted from the experiment described in [15]. Both the test fixture and method used to conduct the experiment were reused. The experimental test setup is shown below in Figure 24. Eight temperature sensors are placed in the plastic container along with the NIST reference. The container is then filled with antifreeze, which was cooled to -15°C in a freezer. The motor is used to stir the antifreeze in order to maintain a constant temperature throughout it. The junction board located on top of the test fixture is used to read the temperature from the eight sensors. A computer running MATLAB code collects the temperature measurements from the sensors and the reference and writes these values into a comma delimited text file along with the time the measurements were taken. This process continues until the antifreeze has warmed up to room temperature. The antifreeze is

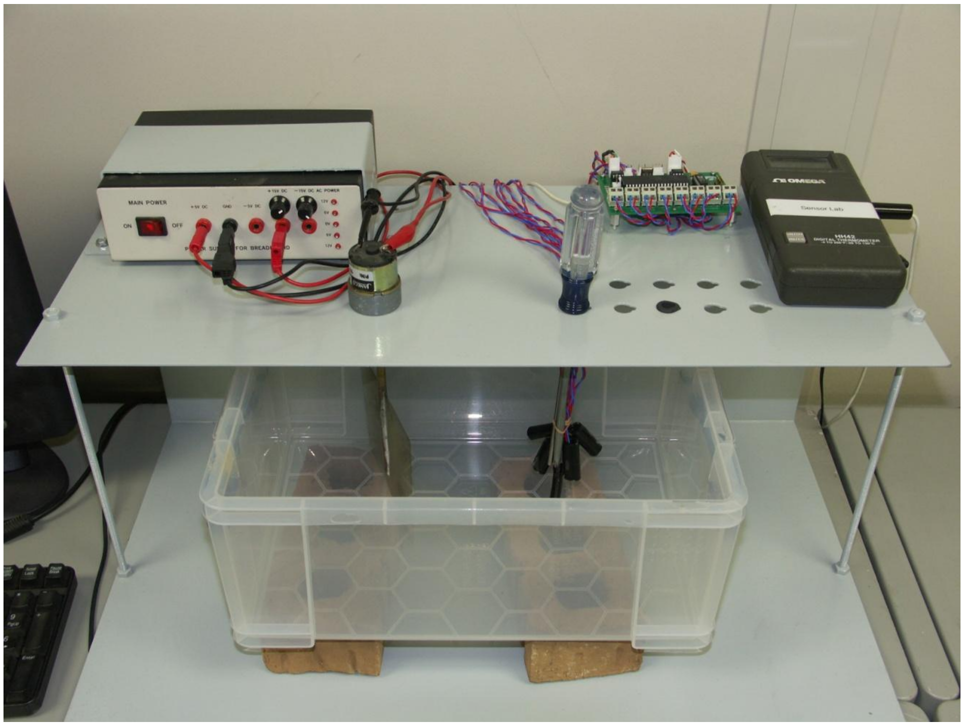


Figure 24. Temperature cycle experiment adapted from [15]. Eight temperature sensors are placed in a plastic container with the NIST reference. The container is filled with antifreeze to test the cold temperatures and hot water to test the hot temperatures. A motor is used to stir the liquid to maintain a constant temperature throughout the liquid. A computer running MATLAB code collects the temperatures from the sensors and reference and stores them in a text file with the time the measurement was taken.

removed from the container and placed in the freezer for the next test. Next, water is poured into a metal pot and placed on a burner. The water is warmed up to approximately +80℃ and then poured into the plastic container. The same process is used for the hot water test as the cold antifreeze test. Once the water has cooled off to room temperature, the experiment is complete. The data from the two tests is combined to obtain temperature measurements for the entire range from −10℃ to +50℃.

The experimental results are shown below in Figure 25, where the actual temperatures from the HH-42 reference are on the x-axis and the measured temperatures from the sensors are on the y-axis. There is a linear relationship between the actual and measured temperatures across the entire range from −10℃ to +50℃, where the

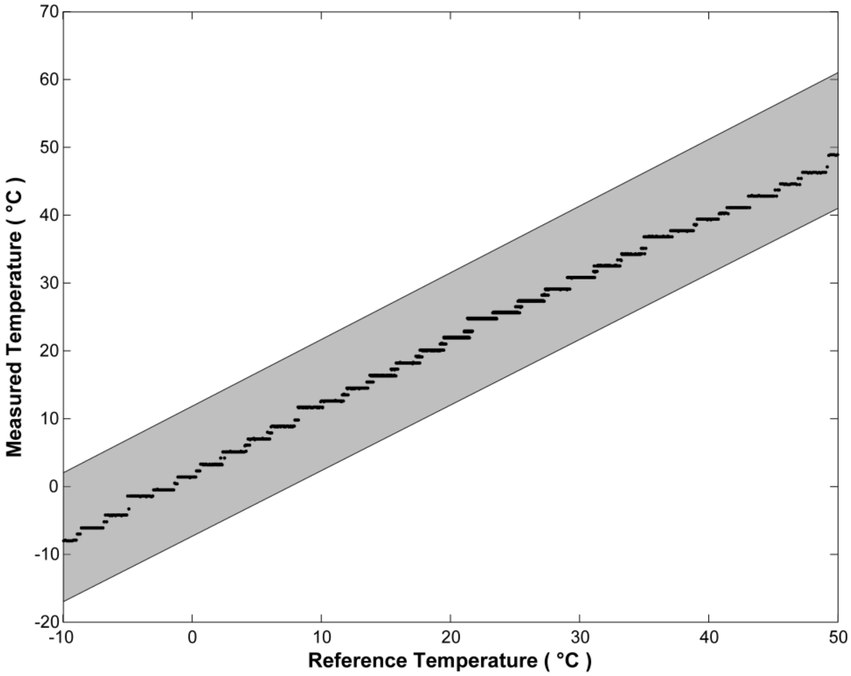


Figure 25. Data collected from 24 temperature sensors across their temperature range. Nearly all of the measured temperatures are within the ±10℃ range specified. The shaded region contains all of the temperature measurements and depicts the spread due to each sensor’s temperature offset. The steps along the y-axis result from quantization of the measured temperature.

temperature offset varies between about $-10°\text{C}$ and $+10°\text{C}$. Also the slope for all of the sensors is very close to one, however most of the slopes are slightly greater than one, as depicted in Figure 26. These results imply that the sensors are linear across their entire range and therefore can be calibrated using a slope and offset. It may be possible to use only a temperature offset, since the slope is close to one, however, the temperature measurements will vary more at the high and low extremes.

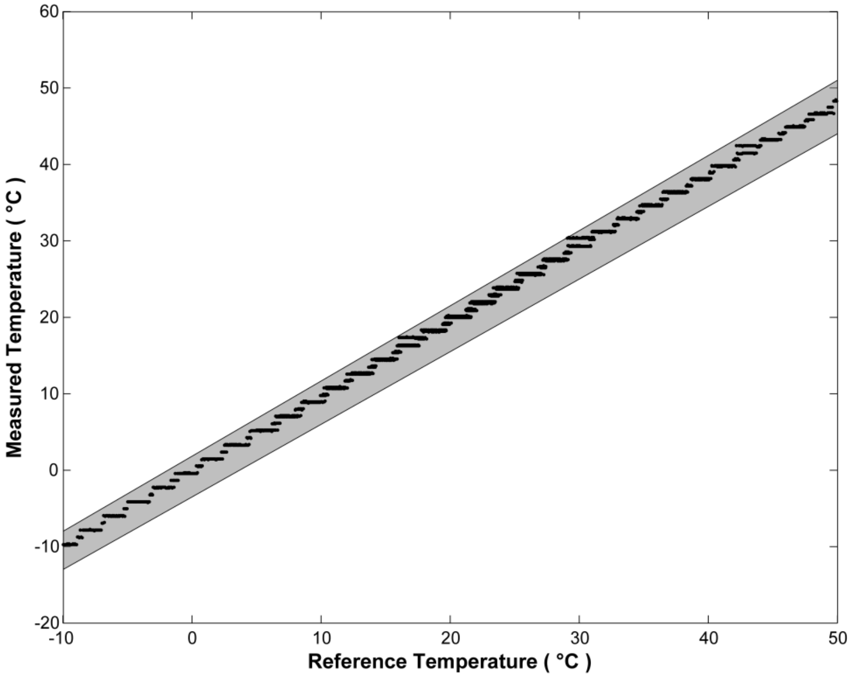


Figure 26. Data collected from 24 temperature sensors across their temperature range after subtracting out each sensor's temperature offset. The shaded region contains all of the temperature measurements and depicts the spread due to each sensor's variation in slope. The steps along the y-axis result from quantization of the measured temperature.

Figure 27 below shows the expected temperature measurement error from using a linear equation to calibrate a temperature sensor. The results for the other temperature sensors before calibration are very similar. The calibration is weighted more heavily at room temperature than at the high and low temperature extremes. This provides a temperature measurement within about ±1℃ in the middle of the sensor's range, where it will operate the majority of the time. The temperature measurement error is within ±2℃ across the entire range from −10℃ to +50℃. The error introduced by using a slope and temperature offset is small enough that the temperature sensors can be calibrated using a simple linear relationship.

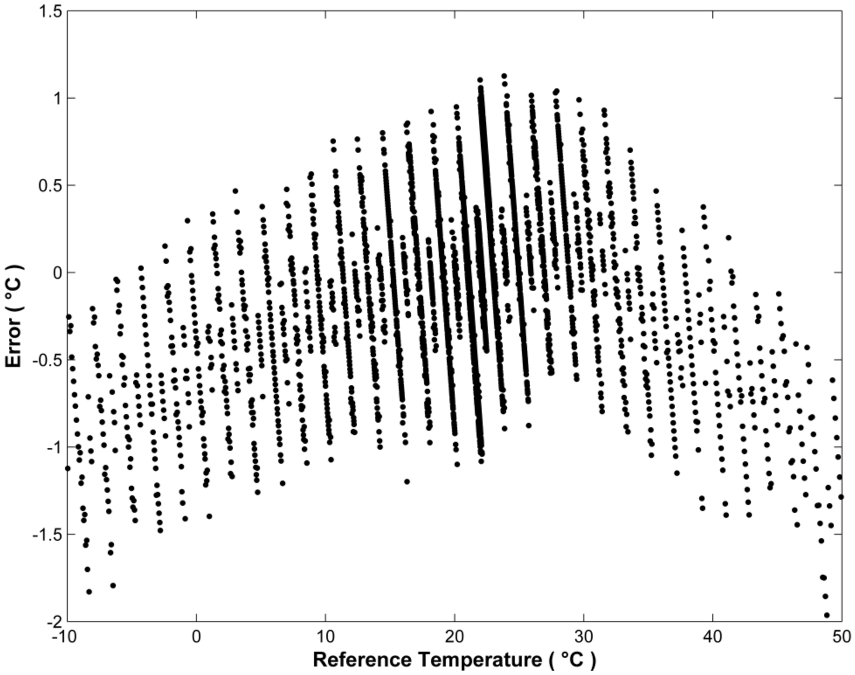


Figure 27. Expected temperature measurement error for the linear calibration of a temperature sensor from −10℃ to +50℃. The calibration is weighted more heavily at room temperature than at the extreme high and low temperatures to provide readings within about ±1℃ in the middle of the sensor's range. The temperature error is within the required ±2℃ range.

Table 6 summarizes the slope and temperature offsets for a representative sample that includes eight sensors. The results from the other 16 temperature sensors are very similar to these eight. The slope is slightly greater than one for all of the temperature sensors, which will result in a larger error than expected. This error is large enough that if only a temperature offset is used for calibration, the sensor will not be accurate to $\pm 2°C$ as required. The temperature offset varies quite a bit from sensor to sensor, but the majority of the sensors read a higher temperature than the reference measured. The coefficient of determination is also included in the table below. This value is very close to one for all of the temperature sensors, which means a linear equation accurately models this relationship with little error. Since the temperature sensors are linear, they can be accurately calibrated using only a slope and temperature offset.

Table 6. Summary of experimental temperature cycle results

Sensor #	**Slope**	**Temp Offset (°C)**	R^2
DG1	1.05	2.74	0.9964
DG2	1.06	−11.09	0.9977
DG3	1.05	1.94	0.9967
DG4	1.03	1.99	0.9971
DG5	1.06	−0.43	0.9972
DG6	1.04	4.23	0.9980
DG7	1.03	−8.98	0.9983
DG8	1.03	6.17	0.9980

Bandgap Voltage and Temperature Offset Relationship

The concept of calibrating temperature sensors by accounting for an offset in voltage is not a new one. This technique is commonly used to develop CMOS smart temperature sensors. See [2] for an overview of these smart temperature sensors. In [4], a temperature sensor is described that is calibrated by trimming the bandgap voltage reference. In [18] and [19] two methods for calibrating temperature sensors by measuring the on-chip reference voltage and trimming the sensor to account for the offset are described. These two techniques allowed the sensors to be calibrated within $\pm 0.25°\mathrm{C}$ across a range of $-55°\mathrm{C}$ to $+125°\mathrm{C}$. These results show that it is possible to calibrate a temperature sensor by measuring and accounting for a voltage offset in a reference. This method is much faster and less costly when compared to conventional temperature calibration techniques.

The ATtiny85V datasheet states the temperature from the internal temperature sensor varies from chip to chip due to variations in the manufacturing process. However, it does not provide any details as to what these variations are. They may stem from the internal bandgap reference voltage, as this also varies between ICs. An experiment was conducted to determine if there is a relationship between the microcontroller's bandgap voltage and temperature offset.

The analog comparator in the ATtiny85V is configured as shown in Figure 28. The ATtiny85V uses its internal 1.1 V bandgap reference as an input to the positive input

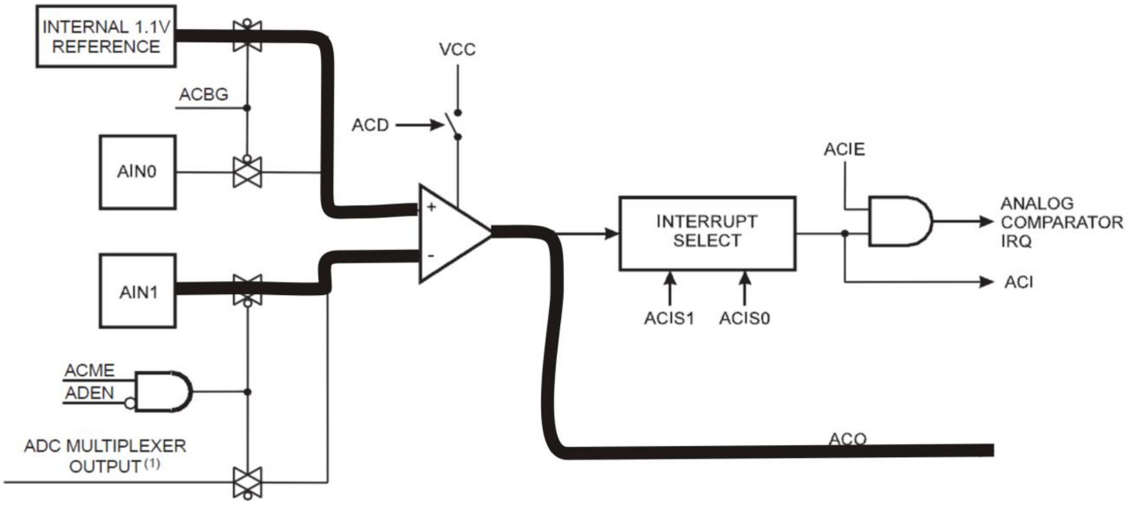


Figure 28. ATtiny85V analog comparator configuration for bandgap voltage reference experiment. The internal 1.1 V reference is attached to the positive input on the comparator and an external voltage source on pin AIN1 is attached to the negative input. The output of the analog comparator, ACO, is monitored for changes in state. See [1] for the original analog comparator image.

of the comparator. A voltage source is attached to the negative input on pin AIN1 and slowly increases the voltage from 1.0 V to 1.2 V until the output of the analog comparator, ACO, changes state. As soon as the comparator's output changes state, the input voltage to the comparator's negative input is equal to the actual voltage of the internal bandgap. The same process is repeated with the voltage source decreasing from 1.2 V to 1.0 V, as there is a small amount of hysteresis built into the analog comparator.

The bandgap voltage experiment is shown below in Figure 29. In order to automate the task of measuring the actual bandgap reference voltage, an Agilent 34970A data acquisition/switch unit is used, as shown on the right side of the figures. The data acquisition unit is connected to a computer using a RS-232 serial cable, and is controlled using MATLAB. The 34970A not only acts as a voltage source to output a voltage between 1.0 V and 1.2 V, but it also detects when the comparator's output changes state. The MATLAB code sends a series of commands to the data acquisition unit to output a range of voltages and once the comparator changes state, it records the bandgap voltage.

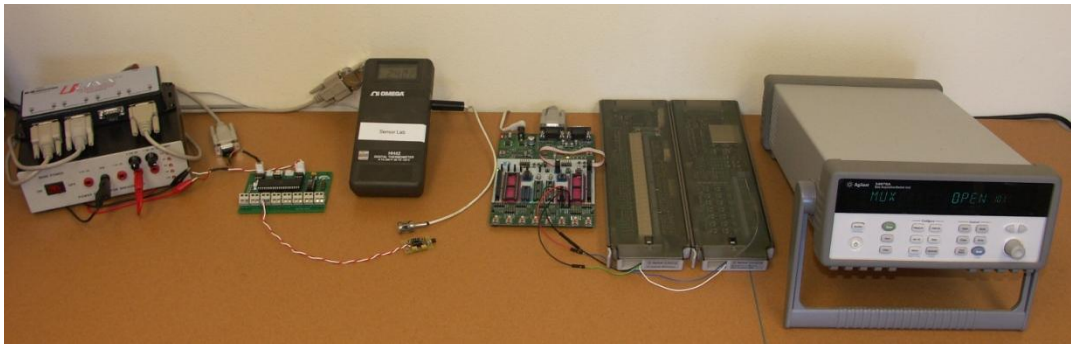


Figure 29. Photo of bandgap voltage experiment setup. From left to right: power supply and USB serial adapter, junction board, NIST calibrated reference, STK500 development board, Agilent 34901A 20 channel multiplexer, Agilent 34907A multifunction module DIO/DAC, and Agilent 34970A data acquisition/switch unit.

After the bandgap voltage is measured, the microcontroller's temperature needs to be measured using the internal temperature sensor. The ATtiny85V is programmed to send the current temperature to a junction board using the pulse width modulation scheme previously explained. The microcontroller is programmed using the STK500 development board shown just right of center in Figure 30. The junction board on the left converts this signal to a temperature and sends the measured temperature to a computer using a RS-232 serial cable. At the same time, the NIST calibrated reference temperature, shown just left of center in the figure below, is read using a MATLAB script that also communicates with a computer through a serial port. Once all of this data is collected, the temperature offset is calculated by subtracting the temperature of the reference from the temperature of the sensor.

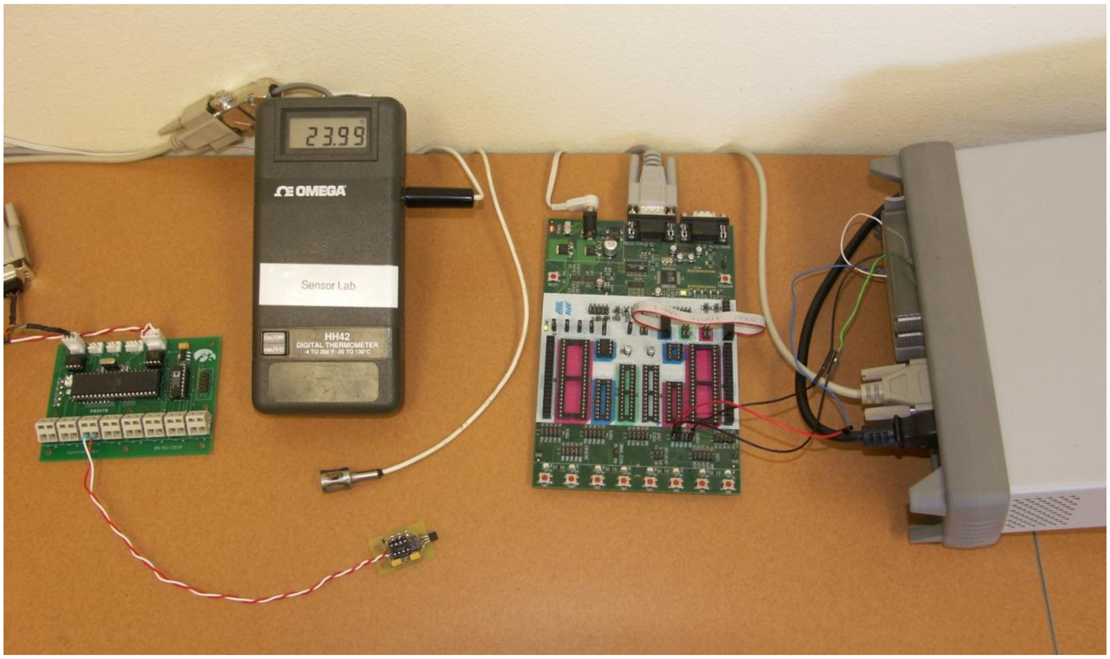


Figure 30. Close up of bandgap voltage experiment setup. From left to right: junction board for measuring sensor temperature, NIST calibrated reference, STK500 development board with ATtiny85V, and Agilent 34970A data acquisition/switch unit.

Figure 31 shows the results from the experiment explained above. The measured internal bandgap voltage is plotted along the x-axis and the measured temperature offset between the internal temperature sensor and NIST calibrated reference is displayed on the y-axis. Each data point represents a different temperature sensor, for a total of 16 sensors. It is evident from the plot that there is not a linear relationship between the internal bandgap voltage and temperature offset. For example, the two points with a measured bandgap voltage around 1.125 V have very different temperature offsets. One has a temperature offset of about +11°C, while the other's temperature offset is around 0°C. There is no way to determine what the correct temperature offset is for this bandgap voltage.

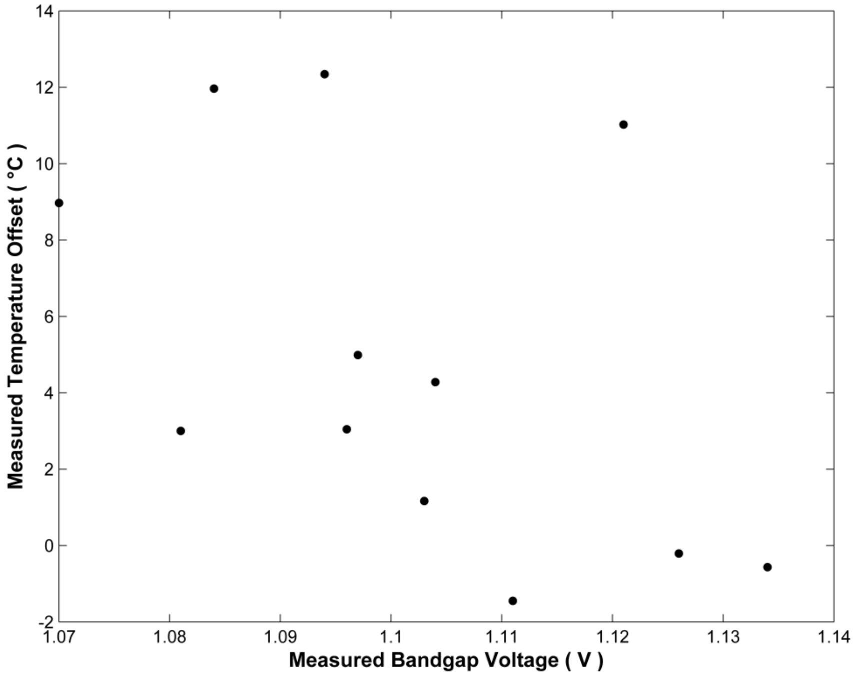


Figure 31. Results from experiment that measured the bandgap voltage and temperature offset for 16 temperature sensors. There is not a linear relationship between the bandgap voltage and temperature offset that can be used to calibrate the temperature sensors.

The internal bandgap voltage cannot be used to calibrate the sensor since there is not a relationship between the temperature offset and bandgap voltage. The temperature offset may result from variations in the manufacturing of the actual temperature sensor and therefore cannot be measured without measuring the offset itself. The next section addresses the process of calibrating the sensor using temperature measurements.

Temperature Offset Calibration

In order to use a single point temperature calibration, the sensor temperature must be compared to a reference temperature to determine the offset. The DS18B20 1-Wire digital temperature sensor was used as the reference. It is accurate within ± 0.5℃ from -10℃ to $+85$℃ and operates from -55℃ to $+125$℃. This sensor was chosen as it provides a 1-Wire interface that only requires an external pull-up resistor. It was used on the original version of the temperature sensor before it was eliminated to reduce the number of components needed.

The procedure to calibrate each sensor with a single temperature offset goes as follows. First, an ATtiny85V is plugged into a STK500 development board and the AVR Studio software connects to the STK500 over a COM port. The microcontroller FUSE bits are configured and the calibration software is loaded onto the microcontroller using this software. Next, the microcontroller is removed from the development board and placed on the calibration board. The calibration board uses a DS18B20 temperature sensor as its reference. A bench-top power supply supplies ground and $+5$ V to the calibration board. It should be noted that the single point calibration is performed at room temperature, or about $+25$℃.

Once the calibration board is powered on, the calibration code begins execution. The first thing the software does is measure the ATtiny85V's internal temperature sensor. Next, the temperature is read from the DS18B20. If this temperature is valid and within the sensor's range it is converted to an equivalent ADC reading. The difference between the ATtiny85V's ADC value and the one calculated from the DS18B20's temperature is stored in the microcontroller's EEPROM. This value is stored as an 8-bit signed integer, which is valid for the range $-128℃$ to $+127℃$. This range is much larger than necessary, since the temperature should be within $\pm 10℃$ according to the ATtiny85V's datasheet. This entire process only takes a couple of seconds.

The ATtiny85V is then removed from the calibration board and plugged back into the STK500 board. Using the AVR Studio software, the value stored in EEPROM is read to verify that the calibration worked correctly. Next, the temperature sensor code is loaded onto the microcontroller using the AVR Studio software. The programmed microcontroller is plugged into another board used for testing the calibrated sensor. This test board is plugged into the junction board discussed previously. The junction board is connected to a computer through a COM port and is powered by the bench-top power supply. The junction board has two DS18B20 temperature sensors onboard that read the temperature in the same area as the sensor under test. Once the junction board has measured the DS18B20s' temperatures, as well as the temperature from the sensor under test, they are displayed in a terminal window on the attached computer. If the temperature from the sensor under test is within two degrees of the reference, it is considered calibrated.

Although this approach provided a simple solution for calibrating the sensors, it did not meet the temperature accuracy requirements. All of the sensors calibrated this way are within the required $\pm 2℃$ around room temperature, however, at temperatures toward the outside of the range this accuracy is not met. At temperatures below $0℃$ and above $+40℃$ variations in the sensors' slopes result in significant errors. In order to provide an accuracy that meets the temperature sensor's requirements, the slope needs to be taken into account by performing a two point calibration. Since the experiment described previously verified that the temperature sensor provides a voltage linearly proportional to the temperature, a two point calibration allows both the slope and temperature offset to be calculated. The next section discusses an experiment to verify the temperature accuracy requirements can be met using a slope and offset.

Temperature Offset and Slope Calibration

The same experiment used for verifying the relationship between the sensor's output voltage and actual temperature is linear is used for determining the sensor's slope and temperature offset. After running both the hot and cold temperature tests, the results are combined and processed using a linear regression to determine the slope and offset. These values for the slope and offset are programmed into the temperature sensor using the bidirectional communication method described previously. Once these calibration constants are programmed into the sensor, it is placed through the same hot and cold

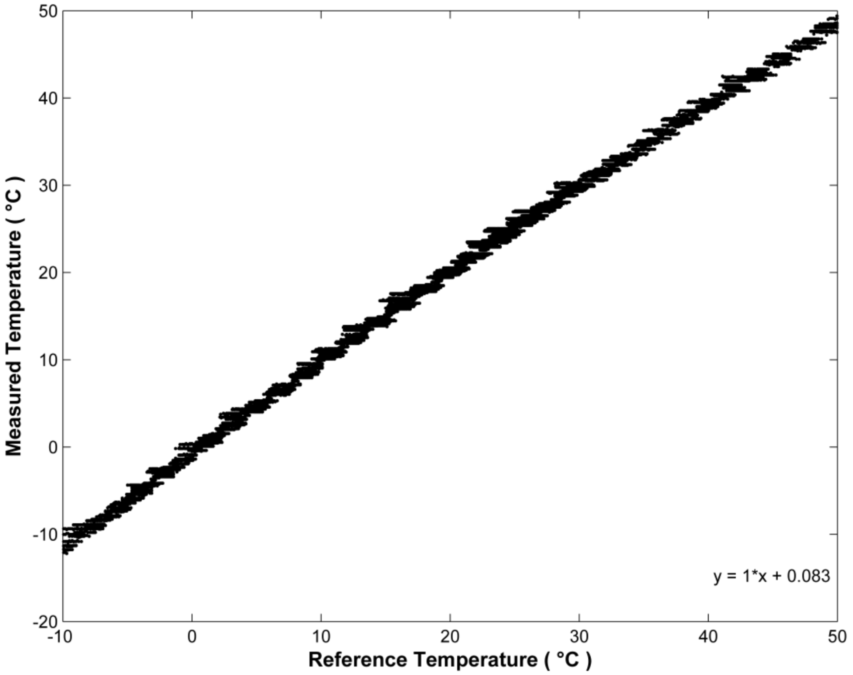


Figure 32. Results for a representative sample of eight calibrated temperature sensors in the range from -10℃ to $+50$℃. The equation in the bottom right corner represents the linear regression. The slope is very close to one and the offset, which corresponds to the average error, is almost zero.

temperature tests as before. After the experiment has completed, the results are combined and processed to verify the calibrated sensor meets the temperature accuracy requirement of ±2℃ over the range from −10℃ to +50℃.

The results from the calibration experiment are shown above in Figure 32. Only a representative sample of eight temperature sensors is shown, however, the other 16 temperature sensors provided very similar results. The equation shown in the bottom right corner represents the linear regression for the calibrated sensors. The slope of the calibrated sensors is very close to one. This means that the calibration corrected the variations in slope between sensors. The temperature offset of the calibrated sensors is very close to zero. This means that the calibration corrected the temperature offset for each of the sensors. These results show that the temperature sensors can be accurately calibrated with an accuracy of ±2℃.

The temperature measurement error after calibration for a temperature sensor is shown below in Figure 33. The other calibrated temperature sensors provided similar results from −10℃ to +50℃. These results show that the sensors are accurate to ±2℃ of the actual temperature across their entire range. The calibration is also weighted more heavily at room temperature than at the extreme high and low temperatures. This provides a temperature measurement within about ±1.5℃ in the middle of the sensor's range, which is where the temperature sensor will operate the majority of the time.

The results from this experiment verify that a linear calibration method will provide an accurate measurement within the specified range. Since only a slope and temperature offset are needed, it is very easy to calibrate the sensors. Only two temperature measurements are required to determine the slope and offset, however, more can be used to ensure an accuracy of ±2℃ is met. After these two calibration constants are determined, they are sent to the temperature sensor using the bidirectional communication method discussed previously. Once the sensor receives these two bytes, they are stored in EEPROM using the procedure described below.

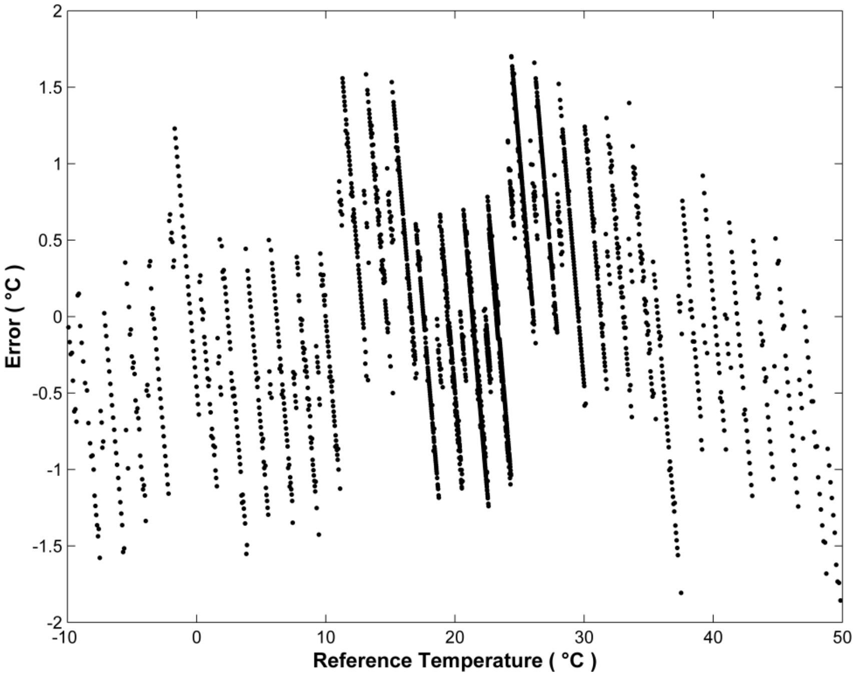


Figure 33. Temperature measurement error after calibration for a temperature sensor from $-10°\text{C}$ to $+50°\text{C}$. The results are very similar for the other calibrated sensors. The temperature error is within the required $\pm 2°\text{C}$ range.

Storing Calibration Data in EEPROM

The ATtiny85V has 512 bytes of internal EEPROM that can be used to store constants or look-up tables. Since EEPROM is non-volatile, this information is retained even after power is removed from the microcontroller. The EEPROM also has 100,000 write cycles, which is significantly greater than what is needed, as the sensors will not be calibrated often. These two features of EEPROM make it an ideal way to store calibration data on the microcontroller for later use.

In order to write data to the EEPROM, the address of where writing should occur needs to be known. A function was developed for writing data to EEPROM given the address and data as unsigned characters. The first thing the function does is wait for any previous writes to finish. Next, the microcontroller must enter EEPROM programming mode in order to write to EEPROM. Both the address and data registers are now initialized to the values passed to the function. Now the EEPROM master program enable bit must be set so that the EEPROM can be written to. Finally, the EEPROM program enable must be set to begin writing the data to the specified address.

One example of calibration data that would be stored in EEPROM is a temperature offset described in the temperature calibration section. Once the offset is obtained by calculating the difference between the internal temperature sensor and the reference, it is written to address 0x00 in EEPROM. When the temperature sensor starts up, it reads this offset from EEPROM. The microcontroller determines if the offset should be positive or negative, since an unsigned character representation of the offset is stored in EEPROM. Once the sign of the temperature offset is recovered, this offset is accounted for when determining the actual temperature to output to the master.

Summary

Variations in the manufacturing process required that the temperature sensor be calibrated in order to provide an accurate temperature measurement. An overview of how the uncalibrated temperature sensors perform was given. The relationship between the internal bandgap reference voltage and temperature offset was explored. A simple method for calibrating the sensor using a temperature offset was discussed. A method for calibrating the sensor using both a slope and temperature offset was also explored. Finally, the process of storing calibration data in EEPROM was discussed.

CHAPTER VI
CONCLUSION

A two-wire, low component count soil temperature sensor was developed for the next generation of rainfall, soil moisture, and soil temperature sites which will be deployed across the state of Iowa. The sensor implemented a two-wire interface where one wire provides ground and the other wire is used for both power and communication. The sensor uses pulse width modulation to send temperature measurements to the master, where the duty cycle is proportional to the temperature. The sensor also parasitically powers itself from the bidirectional data line. In order to minimize the component count, a microcontroller with an internal temperature sensor was used. Finally, the sensor was able to receive data from the master on the bidirectional communication line, which was used for calibrating the sensor.

Future Work

The next step for the Iowa Flood Center is to finalize the design for the next generation rainfall, soil moisture, and soil temperature system. After the system is completed, IFC will begin production of this system. This requires the development of a systematic process for constructing and calibrating these temperature sensors. Currently, this process takes too long and several improvements can be made. Once the accuracy of the calibration is proven, the calibrated sensors only need to be compared to a reference at a few different temperatures instead of across the entire range. The process of storing the calibration constants can also be automated, instead of manually sending these values to the temperature sensor. In addition, the overall process of calibrating the sensors can be automated, which will make the process much less labor intensive. Finally, the construction of the temperature sensors can be carried out by a third-party company in order to reduce production costs.

Another possible improvement that can be made to the temperature sensor developed here involves the use of bidirectional communication. Since the master can now send information to the sensor, the bidirectional communication could be used to reprogram the sensor. There is currently no way to reprogram the sensor once it is sealed, but this provides a solution to this problem. A bootloader could be programmed onto the microcontroller and the master could reprogram the sensor. This would allow firmware upgrades to be done in the field and the sensor would not need to be replaced if the functionality needed to be updated.

Results

A two-wire soil temperature sensor was developed along with a junction box for reading the temperatures from the sensors. The temperature sensor met all of the requirements necessary for this application. A prototype system for measuring rainfall, soil moisture, and soil temperature was deployed at the Iowa City airport. This system demonstrated the reliability and accuracy of the temperature sensors.

REFERENCES

[1] Atmel Corporation. (2008, Jan.). ATtiny25/45/85 preliminary datasheet. Available: http://www.atmel.com/dyn/resources/prod_documents/doc2586.pdf.

[2] A. Bakker, "CMOS smart temperature sensors - an overview," in *Sensors, 2002. Proceedings of IEEE,* 2002, pp. 1423-1427 vol.2.

[3] R. P. Benedict, *Fundamentals of Temperature, Pressure, and Flow Measurements*. New York: Wiley, 1969, pp. 53-126.

[4] H. Dong-Ok, K. Yong-Il, P. Tah-Joon and P. Heon-Chul, "A CMOS temperature sensor with calibration function using band gap voltage reference," in *Sensing Technology, 2008. ICST 2008. 3rd International Conference on,* 2008, pp. 496-499.

[5] H. Fahim Rezaei, N. Sitter and A. Kruger, "Next generation system for real-time monitoring of rainfall, soil moisture, and soil temperature," in *Sensors Applications Symposium, 2011. SAS 2011. IEEE,* 2011.

[6] C. Hagart-Alexander, "Temperature measurement," in *Instrumentation Reference Book*, 4th ed., W. Boyes, Ed. Boston: Butterworth-Heinemann/Elsevier, 2010, pp. 286-306.

[7] C. J. Kaiser, *The Capacitor Handbook.* New York: Van Nostrand Reinhold, 1993.

[8] D. Kalinksy and R. Kalinsky. (2002, Feb. 1). Introduction to serial peripheral interface. Available: http://www.embedded.com/story/OEG20020124S0116.

[9] P. Lee, "Industrial ethernet - Powering the dream," *Computing & Control Engineering Journal,* vol. 17, pp. 24-27, 2006.

[10] Maxim Integrated Products. (2010, Dec. 10). Application note 4542: Long twisted pair reads digital temperature sensor. Available: http://pdfserv.maxim-ic.com/en/an/AN4542.pdf.

[11] Maxim Integrated Products. (2008, Sep. 22). Application note 148: Guidelines for reliable long line 1-wire networks. Available: http://pdfserv.maxim-ic.com/en/an/AN148.pdf.

[12] Maxim Integrated Products. (2008, Apr. 22). DS18B20 programmable resolution 1-wire digital thermometer datasheet. Available: http://pdfserv.maxim-ic.com/en/ds/DS18B20.pdf.

[13] T. D. McGee, *Principles and Methods of Temperature Measurement*. New York: Wiley, 1988, pp. 141-310.

[14] J. Niemeier, A. Kruger, W. Krajewski, W. Eichinger, B. Hornbuckle and L. Cunha, "A High-Temporal and Spatial Resolution Soil Moisture and Soil Temperature Network in Iowa Using Wireless Links," *Eos Trans. AGU,* vol. 88(52), Fall Meet. Suppl., Abstract H13A-0970, 2007.

[15] J. J. Niemeier, "Embedded instrumentation for soil temperature and soil moisture monitoring," 2007.

[16] NXP Semiconductors. (2007, June 19). UM10204: I2C-bus specification and user manual. Available: http://www.nxp.com/documents/user_manual/UM10204.pdf.

[17] Omega Engineering. Ultra-high accuracy and resolution handheld thermistor thermometer. Available: http://www.omega.com/Temperature/pdf/HH40.pdf.

[18] M. A. P. Pertijs, A. L. Aita, K. A. A. Makinwa and J. H. Huijsing, "Low-Cost Calibration Techniques for Smart Temperature Sensors," *Sensors Journal, IEEE,* vol. 10, pp. 1098-1105, 2010.

[19] M. A. P. Pertijs, A. L. Aita, K. A. A. Makinwa and J. H. Huijsing, "Voltage calibration of smart temperature sensors," in *Sensors, 2008 IEEE,* 2008, pp. 756-759.

[20] C. Poki, C. Chun-Chi, T. Chin-Chung and L. Wen-Fu, "A time-to-digital-converter-based CMOS smart temperature sensor," *Solid-State Circuits, IEEE Journal of,* vol. 40, pp. 1642-1648, 2005.

[21] S. Ramo, J. R. Whinnery and T. Van Duzer, "Transmission lines," in *Fields and Waves in Communication Electronics*. New York: Wiley, 1994, pp. 213-273.

[22] D. Schelle. (2009, Aug. 13). Building a bidirectional, single-wire, line-powered transceiver using a few basic parts. Available: http://www.eetimes.com/design/analog-design/4010397/Build-a-bidirectional-single-wire-line-powered-transceiver-using-just-a-few-basic-parts.

[23] R. J. Stephenson, A. M. Moulin, M. E. Welland and et al, "Temperature measurement," in *The Measurement, Instrumentation, and Sensors Handbook,* J. G. Webster, Ed. Boca Raton: CRC Press, 1999, pp. 1-86.

[24] J. Walko, "Poised for power [power over Ethernet]," *Power Engineer,* vol. 19, pp. 38-41, 2005.