# Project 1:

## Designing a Microprocessor Based Data Collection System

## (Including Design Trade-offs)

## *Checking the Earth, Sea, and Air*

*Danang Technical University*

## *James K. Peckol*

## Lab Objectives:

The objectives of this lab are to work with the Arduino microprocessor, to utilize our knowledge of memory, and to learn and practice the full product development life cycle while documenting, designing, building, and testing a portable data monitoring and acquisition system. That system must be capable interfacing to several different environmental sensors, collecting then processing the data from those sensors, synchronously transmitting the collected data to a local controller, and ultimately asynchronously transmitting the data to and receiving commands a collection management station.



## Cautions and Warnings:

When you are working with the serial portion of the design, make certain to check all of your connections, signals and voltage levels prior to connecting to the computers. We do not want to risk damaging the equipment.

Contrary to beliefs in some circles, a 4-bit, ripple, binary up counter cannot be converted into the equivalent down counter merely by interchanging power and ground nor by mounting the chips on the opposite side of the PCB. Rather, such an attempt is more likely to release the smoke demon – not a pleasant prospect to say the least.

## Observations and Curious Questions:

If the water leaving a flushed toilet circulates one direction north of the equator and the opposite direction to the south, what direction does it go at the equator? At what point does it change directions? What if half of the toilet is to the north and the other half to the south…then what happens? Does the same thing happen to a ballet dancer? How about a Tasmanian devil?

If you hide in the closet with your cell phone, how does a telephone call know exactly where you are when no one else does? If it can’t find you is it a smart or not smart phone?

## Getting Started…

You have just decided to take advantage of a rare opportunity to participate in the formation an exciting new start up. As a result of the growing interest in and concern about climate change, you and several colleagues have decided to set your own direction and to start a company to develop a multipurpose environmental monitoring and data collection system. Based upon initial discussions with your colleagues, you decide to call the company *Watchin’nCollectinStuffRUs-Un.Ltd* and have put together a team and preliminary set requirements for a small portable low power data logging system.

The product, *CheckIt-StoreIt*, will have the ability to perform many of the basic environmental measurements that people studying and researching factors potentially affecting climate change need to make.

You are working as one of the senior development engineers. As part of the early planning of that project, you and one of the marketing folk are traveling around the country talking with people from a number of different environmental research and engineering firms. You are trying to determine what features your customers would like to see in the next generation product.

You’ve been on the road with this guy for a couple of weeks now and are anxious to get home. All the cities are beginning to look exactly alike. Tuesday, this must be Falmouth, Massachusetts …hmmm, looks just like East Hampton. Hey, I wonder if I can get a good beer tonight…hope we can find some good lobster or scrod or Italian food…maybe a meatball sandwich…glad this is the last stop for this trip. This morning, you’re talking with the people working in environmental and ecological monitoring at the Woods Hole Research Center. They’re interested in a new low cost portable monitoring and logging tools that can be used in a variety of field locations around the world.

Following several hours of discussion with Dr. **Nadine T. Laporte,** one of the Associate Scientists, you identify most of they would like to see. Here’s what she has told you,

Ideally, we’d like to be able to use the same instrument on any of the different monitoring sites around the world.

Today, we have our assistants running most of the tests and collecting the data manually, but, in future, we’d like to be able to automate and collect data from as many of these tests as we can. As we upgrade our systems, we’d also like to be able to operate several of these systems remotely from a single PC. Because we are typically monitoring in remote sites, we’d like our systems to be battery powered and to have those batteries last as long as possible.

Here are some of the things that we are currently doing and would like to automate in future. During the early phases of our research, we would like to identify and measure the concentration of several different pollutants or other chemical elements as well as to measure the temperature at the site, and potentially the flow rate of surrounding rivers and streams.

When measuring temperature, we would like to make up to 10 measurements per hour. We will be using J type thermocouples and the temperatures will be in the range of -30 to +100 C. We must have an error on these measurements of less than 0.1 C. We will need to express the measured temperature in Fahrenheit or in Celsius.

We also have several other sensors, including one for carbon and one for salinity, for which we must measure signals in the range of 10 to 350 ppm (parts per million) for the carbon level sensor and 5 to 50 ppt (parts per thousand) for salinity. The carbon and salinity measurements should made at a rate of 5 or 6 per hour.

Finally, to measure flow rate, the transducer we’re using can measure the flow rate range of 100 to 1000 liters per second.

We want the instrument to be able to collect data from at least four different sensors at any one site.

## Supporting Material

The text *Embedded Systems A Contemporary Design Tool* contains several chapters that are directly relevant to the tasks presented in this project. Copies of this text should be available in the department library. You are strongly encouraged to look through those. These include: Chapters 5, 8, 9, 10, 15, 16, and 17. Chapter 12 (corresponding to the text’s Chapter 9) is also on line at he site we pointed out in class under *documentation* and labeled *Design Cycle*. You will find this particularly relevant during the specification and early design phases. You should also have copies of the class lecture notes.

## Sensors

### Carbon and Salinity

The carbon and salinity sensors to be used produce a voltage output that ranges from 0 to 400 mV. The signals must be measured with 0.1 mV resolution.

For the carbon level sensor, an output of 5.0 to 250.0 mV corresponds to 10.0 to 350.0 ppm (parts per million). For the salinity sensor, the range is 100.0 to 300.0 mV, which corresponds to 5.0 to 50.0 ppt (parts per thousand).

The flow rate transducer produces an output frequency in the range of 1.00 KHz to 10.00 KHz that corresponds to the flow rate range of 100.0 to 1000.0 liters per second.

### Temperature

Before we begin, let’s learn a little about making temperature measurements. There is a wider variety of methods for making such measurements. One common and inexpensive technique is to use a sensing device called a thermocouple.

Thermoelectricity was discovered by Seebeck in 1821. He found that when two wires made of dissimilar metals were connected to each other at two points and the two junctions held at different temperatures that current will flow. The flow will continue as long as there is a temperature difference. The phenomenon is called the Seebeck Effect and the force driving the current is Seebeck thermal emf. This electromotive force (voltage) is the parameter measured in thermocouple thermometry. Thus, a thermocouple is simply junction of two dissimilar metals.

When a circuit containing two dissimilar metals completed (as we see in the following figure) there will always be at least one thermocouple in loop. The simple loop shown contains two dissimilar metals A and B and two junctions: TM – measurement and TR – reference. The amount of current flowing is related to temperature difference.



We measure the voltage as we see in the accompanying figure…

The first operation in converting the measured thermoelectric voltage (VAB) to an equivalent temperature value is the algebraic addition of the voltage measured at the reference junction terminals (VJuncA and VJuncB ) and the calculated reference junction voltage (VRef). The sum represents an approximation of the thermoelectric voltage generated at the temperature-sensing junction (TM in the figure).



As with most real world sensing devices, the thermocouple is a non-linear device. As a result, we approximate the temperature corresponding to its corresponding output voltage according to a power series equation given as:

TM = c0 + c1V + c2V2+ . . . +cnVn

where:

V = thermoelectric voltage (microvolts)  
cn = type-dependent polynomial coefficients  
T = temperature (C)  
n = order of the polynomial

The calculated thermoelectric voltage generated at TM is converted into an equivalent temperature value using such a power series polynomial along with type-dependent coefficient tables.

Tables for each thermocouple type containing coefficients representing quadratic(second order), cubic (third order), or quartic (fourth order) should be available online. Voltage-to-temperature conversion accuracy can be increased by using higher order coefficient tables, but at the cost of longer processing time to perform the calculations. Accuracy can be further enhanced by selecting tables representing the narrowest temperature range for the specific measurement application.

For a fourth order polynomial for a J type thermocouple (-200C to 0C with error range -0.4C to 0.5C and 0C to 760C with error range -0.9°C to 0.7°C), we have the following coefficients in tables 1 and 2 respectively:

|  |
| --- |
| c0 = 0.0 |
| c1 = 1.8843850x10-2 |
| c2 = 1.2029733x10-6 |
| c3 = -2.5278593x10-10 |
| c4 = -2.5849263x10-14 |

|  |
| --- |
| c0 = 0.0 |
| c1 = 1.9323799x10-2 |
| c2 = -1.0306020x10-7 |
| c3 = 3.7084018x10-12 |
| c4 = -5.1031937x10-17 |

Table 1 Table 2

To perform a temperature measurement, we record the reference junction temperature (C) and the microvolt output of the thermocouple circuit. Next, we compensate the circuit output voltage for any deviation from 0C in the reference junction by multiplying the measured reference junction temperature by the appropriate Seebeck coefficient. For the type J thermocouple, the value is 51.71µV/C. If a more accurate conversion is required, the reference junction temperature can be converted into an equivalent thermoelectric voltage using a power series polynomial.

The calculated value of the reference junction voltage is then algebraically added to the thermocouple circuit output voltage measured at the reference junction. The new value represents an approximation of the thermoelectric voltage generated by the temperature-sensing junction (TM) of the thermocouple. The calculated voltage must now be converted into an equivalent temperature value. Conversion is accomplished using a power series polynomial.

For our prototype, we will neglect the reference junction and work strictly with the power series polynomial.

The following table gives the output voltage, in microvolts, for the J type thermocouple for the specified temperatures in Celsius.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C | 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| -200 | -7 890 | -8 095 |  |  |  |  |  |  |  |  |  |
| -100 | -4 633 | -5 037 | -5 426 | -5 801 | -6 159 | -6 500 | -6 821 | -7 123 | -7 403 | -7 659 | 7 890 |
| 0 | 0 | -501 | -995 | -1 482 | -1 961 | -2 431 | -2 893 | -3 344 | -3 786 | -4 215 | 4 633 |
| 0 | 0 | 507 | 1 019 | 1 537 | 2 059 | 2 585 | 3 116 | 3 650 | 4 187 | 4 726 | 5 269 |
| 100 | 5 269 | 5 814 | 6 360 | 6 909 | 7 459 | 8 010 | 8 562 | 9 115 | 9 669 | 10 224 | 10 779 |
| 200 | 10 779 | 11 334 | 11 889 | 12 445 | 13 000 | 13 555 | 14 110 | 14 665 | 15 219 | 15 773 | 16 327 |
| 300 | 16 327 | 16 881 | 17 434 | 17 986 | 18 538 | 19 090 | 19 642 | 20 194 | 20 745 | 21 297 | 21 848 |
| 400 | 21 848 | 22 400 | 22 952 | 23 504 | 24 057 | 24 610 | 25 164 | 25 720 | 26 276 | 26 834 | 27 393 |
| 500 | 27 393 | 27 953 | 28 516 | 29 080 | 29 647 | 30 216 | 30 788 | 31 362 | 31 939 | 32 519 | 33 102 |
| 600 | 33 102 | 33 689 | 34 279 | 34 873 | 35 470 | 36 071 | 36 675 | 37 284 | 37 896 | 38 512 | 39 132 |
| 700 | 39 132 | 39 755 | 40 382 | 41 012 | 41 645 | 42 281 | 42 919 | 43 559 | 44 203 | 44 848 | 45 494 |
| 800 | 45 494 | 46 141 | 46 786 | 47 431 | 48 074 | 48 715 | 49 353 | 49 989 | 50 622 | 51 251 | 51 877 |
| 900 | 51 877 | 52 500 | 53 119 | 53 735 | 54 347 | 54 956 | 55 561 | 56 164 | 56 763 | 57 360 | 57 953 |

## Project:

### General Description

The project will be developed in two phases. The first phase deliverables will include a stand-alone prototype system that will implement the site based data collection, the sensor interfaces, the associated sensor drivers, the environmental measurements, and the display of those measurements. Measurement results are to be presented on an LCD display on the system front panel.

The second phase will add a local controller at the site and the bidirectional communication between the measurement and collection system and the local controller over an I2C serial channel, a bidirectional link from the monitored site to the local collection station that is to be implemented as a serial interface, and the associated protocol supporting both interchanges.

The full project report is to be delivered at the end of the completed project. Low cost and reliability are key goals.

A final Phase 2 high-level block diagram for the system is given in the following figure. Phase 1 will only be a fully operational single measurement and collection system.



For the prototype system, it will only be necessary to implement one fully functional Measurement and Collection System site. The remaining six will be modeled. The final delivered Environmental Monitoring System must have the capability to support up to seven fully functional Measurement and Collection units.

#### Phase 1

Efforts during the first phase of the project will focus on specifying, designing, building, and testing the basic instrument with all of the required measurement capabilities. Such capabilities include the ability to measure and display: temperature, voltage, and flow rate.

The results of each measurement are converted to the appropriate units and displayed on an LCD with the corresponding units.

1. Temperature: Selectable F or C
2. Flow Rate: lps – Liters per second
3. Carbon Level: ppm – Parts per million
4. Salinity Level: ppt – Parts per thousand

The flow rate measurement is to be implemented in hardware outside of the Arduino. The control signals coordinating all such measurements should be generated by the Arduino. Such a decision (in this case made for you) is typical of those we make in trading off which portions of a design to implement in hardware and which to do in software.

The serial port on the Arduino can be used for communication with the PC for debugging and testing purposes.

Groups may implement additional capabilities for extra credit. The required applications will be marked based upon creativity and complexity.

Phase 1 deliverables are specified below.

#### Phase 2

During the second phase, a local controller, an I2C based local area network between the local controller and the measurement and collection system, and a serial link between the local controller and a PC will be incorporated to complete the full system.

The local area network between the remote measurement and collection systems and the local controller can be implemented using the built-in I2C interface capability on the Arduino on both sides of the link. The built-in serial capability on the Arduino can be used to provide the local controller portion of the link and that on the PC can be used to provide the master PC portion.

For this project, we will be working primarily at the *Application Level* of our network protocol stack. See Appendix A. Our application is an addition to the manually controlled instrument developed in Phase 1. The application requirements are extended and specified as follows:

* The application on the PC must support the ability to command any instrument to make all measurements.
* The 16 most recent measurements of each type must be stored in a memory on the local controller for display on demand. Such capability is typical in data logging applications.
* The application must support the ability to retrieve the stored data from the local controller and display the data on the PC screen.
* The local controller connected to the PC shall be the I2C master in the I2C LAN and the measurement and collection instruments shall be slaves.
* A Failure Modes and Effects Analysis must be conducted on the final design. An explanation of such an analysis and the requirements are given in Appendix B.

***Note:*** Whenever interfacing a newly implemented design with another system (such as the computer), always double-check the voltage levels and connections to verify there has not been an error. One does not want to destroy the system one is supposed to be testing with.

Detailed Phase 2 deliverables are specified below.

## Deliverables

The following are the deliverables for the project,

#### All Phases

1. Weekly status report from each team member emailed to the instructor and TA by Friday afternoon describing his or her individual efforts and contributions on the project for the previous week. This is to be an individual, not team, report.

### High Level Phase 1 Deliverables

* The required documentation,
* A working prototype of a full featured measurement and collection system,
* A plan for modeling the full remote system,
* A plan for / definition of the command and control interface to a local PC.

### High Level Phase 2 Deliverables

* The full documentation,
* A working prototype of the local controller,
* A working prototype of the remote measurement and collection systems,
* Support for data logging at each remote node (1 complete unit and 6 modeled uints),
* A working prototype of the local and remote portions of the network.
* A working prototype of command and control via the serial communication link between the local controller and the PC.
* A working prototype of command and control via the network between the local controller and the remote nodes.

#### Phase 1

Week One – ***Preliminary Development –*** Deliverables – See online schedule

For a discussion of the use case diagrams, requirements, and design specifications, see the EE 472 or on-line text for examples, the expected format, and general content.

1. A hard copy of the following are due on the date listed on the class web page, prior to the first design review:

For writing both the requirements and design specifications, see the EE 472 or on-line text for examples and the expected format and general content.

1. UML Use Case diagrams for the local controller; each use case must include the graphic and text portion. The text portion must include: a description of the use case and the identification of any exceptions.
2. A Requirements document for the manual measurement and collection operation that formally captures what the customer at Woods Hole has asked for.
3. A Design Specification for the manual measurement and collection operation that formalizes and provides firm, quantified specifications for each of the identified requirements.
4. A preliminary functional decomposition for the software of the measurement and collection portion of the system.
5. Preliminary block diagrams for the hardware of the local controller and a measurement and collection unit. A copy of the block diagram in this lab is not sufficient.
6. A preliminary Bill of Materials.
7. A full schedule, presented as a Gant Chart, specifying the major tasks and milestones on the project and the primary person responsible for each task. These are due on the date listed on the class web page.
8. A plan for modeling the remote systems on the LAN

Week Two – ***Detailed Development*** – Deliverables – See online schedule

For an explanation of data and control flow, activity, and sequence diagrams, see the embedded systems text for examples, the expected format, and general content.

1. A project design review on the date listed on the class web page. Documentation for the design review should include the week 1 deliverables as well as an electronic version of the following:

* An updated Requirements Specification, Design Specification, Test Plan, and Bill of Materials as necessary.
* Updated UML diagrams.
* An updated functional decomposition for the local controller and measurement and collection system.
* Updated hardware block diagrams for the local controller and measurement and collection units.
* An initial software block diagram for the local controller and measurement and collection unit. This should follow the same general format as that for the hardware block diagram.
* A preliminary Failure Modes Analysis based upon the updated block diagrams. The exceptions identified in the text portion of the Use Case analysis can be a good place to begin thinking about this.
* Data and control flow, activity, and sequence diagrams for the local data acquisition system.
* An initial plan for modeling a slave measurement and collection systems on the LAN.
* An annotated list of the commands and responses for the exchange over the LAN.
* An updated schedule as a Gant chart reflecting the current state of the project.

1. Phase I project design review. Be prepared to justify your design decisions and design.

Week Three – ***Detailed Design, Implementation, and Test*** Phase 1 – Deliverables – See online schedule

1. A Phase I demo on the date listed on the class web page. Documentation for the demo should include an electronic version of the following:

* An updated Requirements Specification, Design Specification, Test Plan, and Bill of Materials reflecting the requirements and specifications for the networked counter operation that formally captures what the customer at Woods Hole has asked for.
* A set of test cases based upon the Test Plan for the current system demonstrating operation under normal and faulted conditions.
* A complete specification for the commands and protocol for the message exchange over the LAN.
* Updated bill of materials and schedule reflecting the current state of the project.

1. A demo of the Phase 1 deliverables. The first phase demo shall show the stand-alone operation of the measurement and collection portion of the system.

Phase II project design review. Note that this occurs at the time of the Phase 1 demo. Documentation for the design review should include:

* Updated and detailed block diagram for the full system.
* Updated Requirements document including additions for the intrasystem networks (I2C and serial) that formally captures what the customer at Woods Hole has asked for.
* Updated Design Specification including additions for the intrasystem networks that formalizes and provides firm specifications for each of the identified requirements.

In particular, this should include the description and specification of the protocols for the exchanges between the measurement and collection system and the local controller and between the local controller and the master PC.

* Updated bill of materials and schedule,

#### Phase 2

Week Four – ***Detailed Design, Implementation, and Test*** Phase 1I – Deliverables – See online schedule

1. Project demo of the complete working system

* One full featured remote node.
* Six modeled remote nodes.
* Full command and control capability via the network between the local and remote nodes.
* Full command and control capability via a serial link between the local controller and the PC.
* A set of test cases based upon the Test Plan for the full system.
* Project demo of the complete working system. The demo will be on the date listed on the class web page.
* Demonstrated standby and operating power consumption for your system.
* Final parts cost.

1. Final project report. The report is due on the date listed on the class web page.
2. There will be a 10% bonus for the measurement and collection design with the lowest parts cost and power consumption. However, elimination of bypass capacitors, bus pull-up resistors and other parts mandatory for a good design is not an acceptable way to cut cost. Projects doing so will not be eligible for the bonus.
3. Additional features will be individually evaluated but only if the main portion of the system is fully functional.

Your final deliverables in your report for this project include,

1. Completed and updated Requirements and Design Specifications including UML diagrams.
2. The final Failure Modes Analysis.
3. Final detailed system block diagrams.
4. A final set of detailed hardware and software block diagrams
5. Schematic / logic diagrams for your system as appropriate.
6. System timing diagram as appropriate.
7. System state diagram(s).
8. Logic equations or Verilog listing.
9. C code software listings for your system.
10. Logic analyzer printouts and accompanying timing analysis as appropriate.
11. Test Plan.
12. Analysis and discussion of problems encountered in the design and implementation of the system as appropriate.
13. Short technical description of the system.
14. Final factory cost (BOM) for your system.
15. Final power consumption measurements.
16. Final updated schedule
17. Demo to your TA or instructorof a working system.

# Appendix A

## Background Information:

Microcomputer communications is a rapidly growing field with an ever-increasing num­ber of applications, ranging from local PC networks to large-scale communication systems. Cen­tral to any communications between electronic devices is a protocol for transmitting and receiving information. Within a computer, data is usually transferred in parallel form, such as on a micro­processor or an I/O bus. While parallel communication is far more efficient than serial at moving large numbers of bits, it is not always as practical. Thus, most communications a computer and other electronic devices that are not in the immediate vicinity is usually done using a serial scheme – RS-232, Ethernet, USB, WiFi, Firewire, etc….

Of course, with two different formats for exchanging data there will be many occa­sions in which data have to be converted from one form to the other. There are a number of ICs available to accomplish this task, such as the INTEL 8274 MPSC (Multi-Protocol Serial Control­ler).

Another important element of communication is ensuring that the data given to the user following reception contains no errors arising from such a transmission. Note that we do not guarantee there are no transmission errors, these happen. Rather, at the end of the day, we guarantee the data to be correct. There are a variety of schemes by which this is accomplished: all begin with recognizing that a transmission error has occurred. We’ll examine one, simple parity checking.

When debugging digital circuits, it is helpful to understand what each part of the circuit is doing and exactly when certain events are occurring. The amount of information can be huge, and often we would like to filter out as much of the unnecessary data as possible. Also, sometimes we are interested in timing measurements, other times we are more concerned with comparing the different states of various parts of the circuit. Use the logic analyzer to do this.

### Serial Communication: Asynchronous vs. Synchronous

Asynchronous communication suggests that there are irregular intervals between the send­ing of data. Suppose that a serial communication line is set up to transmit ASCII characters as typed by a person at a keyboard. The spacing between the transmission of each character will vary widely, and there may be long periods when no characters are typed (coffee breaks, tea breaks, bio break, naps, etc.). In this situation, the receiving device needs to be told when a character is being sent to prepare it to receive that character and sort out which part is data, which part is the error-checking field, and so on.

This is accomplished by a procedure known as *framing*, in which a start bit is placed before the first data character, and a stop bit is placed at the end of the transmission of one character. The start bit enables the receiving device to temporarily synchronize with the transmitting device, while the stop bits allow the receiving device time to get ready for the next frame (see Intel sup­plement).

In contrast, when blocks (usually large ones) of regularly spaced data are transferred over a serial line, the transmitter and receiver can be synchronized, transferring characters at a much higher rate. In this format, known as synchronous transmission, start and stop bits are no longer needed, since the receiving device knows exactly where each character begins and ends.

Although synchronous transfer requires less overhead and therefore is much more efficient, its uses are more limited than asynchronous data transfer, and thus the latter is more widely used.

### Network Overview

Our serial network is designed to exchange information between a central host and a number of data loggers at remote sites. The network implements half-duplex communication, that is, data may be sent in only one direction at a time.

Traditionally, networks comprise a number of layers - each viewed as a virtual machine upon which the layer above operates. OSI uses 7, TCP/IP uses 5. For our design, we will use 4: the *physical laye*r, the *data link* layer, the *protocol layer*, and the *application layer*. These are described in the following.

#### The Physical Layer

The physical layer for our network will comprise 5 lines and their associated signals: transmitted data, received data, signal ground, and two control lines. Signaling levels will be those specified for RS-232 as described above.

#### The Data Link Layer

The data link layer will move characters, expressed in the RS-232 format, from the host to a remote site and from a remote site to the host.

#### The Protocol Layer

The Protocol layer will move commands and data from the host to a node at the remote site and move data from a node at the remote site to the host. Each node within such system is identified by a 3 bit address.

#### The Application Layer

The Application layer will provide the link between an application on the host and one on a remote node. We wish to be able to communicate with and execute applications on up to 7 systems on our network as shown in the example block diagram below. For this project it is sufficient to demonstrate one.



#### Framing

In serial communications, data is grouped into frames, which have already been partially described above. The start of a frame is signaled by the start bit, the end by one or more stop bits. Data and error checking (parity bit) codes are contained within each frame.

In order to perform serial to parallel conversion, our circuit must know what part of the frame is being looked at any one time. For example, we don’t want to be looking for a start bit in the middle of the data, nor do we want to be checking parity when the start bit is being received. One way to keep track of the serial input is to use a counting mechanism activated by the start bit of each frame. The output of the data logger can be used as input to control logic that determines when to clock in a data bit, when to check parity, and so forth.

Please note that basic ASCII data is based on a 7 bit code (0x00 through 0x7F), so when HyperTerminal or the Serial Port Monitor is set for 8 bit data, it will add in an extra bit to ASCII values. Normally it will stuff a ‘0’ in the MSB position unless it has a stream of identical characters. In this case, on the third occurrence, it will begin to toggle the MSB. For example, when transmitting a series of D’s, it will send out, in order:

0101 0010

0101 0010

1101 0010

0101 0010

1101 0010

etc…

This is not something that we have to worry about for this project.

# Appendix B: Failure Modes and Effects Analysis

As an engineer, designing something, building it, and getting it to work properly and reliably is an exciting part of the job. Getting that first LED to turn on is often a major milestone and accomplishment in the process and almost worth getting a beer to celebrate. However, getting the design to work is not the end of the job.

Examining our design and asking how, where, and when it might fail is equally, if not more, important. In this project, we want to take a first look at potential failure modes and failure analysis. Such a practice, known as Failure Modes and Effects Analysis (FMEA) is commonly used in many industries where product safety and reliability are critical.

As our first step, we will take a very high-level view. For this initial step, we will limit our investigation to the various signals in our system and to three kinds of failure modes: *SA0, SA1*, and *Bridge*. These are a signal *Stuck At 0* or *Stuck At 1* or two signals incorrectly/unintentionally connected together called a *bridge*.

We look at each input, output, and internal signal in the system and ask: What are the effects if any of the three failure modes occurs on or to that signal and what are the consequences and severity of such a failure on the operation of the system? In a more complete analysis, we determine the probability or each fault occurring as we assess the risk if such a fault should occur.

For this project, we will examine each input, output, and internal signal for each subsystem and analyze the effects on that subsystem of a SA0 or SA1 fault on that signal, on the operation of the subsystem, and the entire system. We will defer bridge faults.

A typical set of ground rules for conducting such an analysis are:

1. Only one failure mode exists at a time.
2. All inputs (including software commands) to the system or subsystem being analyzed are present and at nominal values.
3. Sufficient power is available to the system.