K4HJU

Amateur Radio
Woodcreek, Texas

Prototyping and Proof of Concept Models Development Environment

Rev 0.02
Jim Satterwhite K4HJU

A Work in Progress

K4HJU
3/6/2018

Table of Contents

1. INTRODUCTION 3
2. THE PROTOTYPING SYSTEM 4
3. WINDOWS APPLICATION DEVELOPMENT 5
4. MICROCONTROLLER SOFTWARE DEVELOPMENT 5
5. CALCULATION AND ANALYSIS 6
6. COMMUNICATIONS LINK 6
7. SCHEMATIC ENTRY AND LAYOUT 7
8. FPGA 7
9. TEST EQUIPMENT 8
10. OPERATION 8

Table of Figures

FIGURE 2-1 TYPICAL PROTOTYPE SYSTEM 4
FIGURE 3-1 WINDOWS APPLICATION DEVELOPMENT 5
FIGURE 4-1 MICROCONTROLLER DEVELOPMENT ENVIRONMENT 5
FIGURE 5-1 MATHCAD ANALYSIS 6
FIGURE 6-1 USB TO RS232 AND MICROCONTROLLER MODULES 6
FIGURE 7-1 BREADBOARD SCHEMATIC 7
FIGURE 7-2 BREADBOARD LAYOUT 7

1. Introduction

Over the many years of my career I have developed an electronic development system that works really well for me. It consists of two parts, the prototyping system and the development environment. The prototyping system is described in another document. The environment is a collage of functioning parts operating in unison to facilitate the system development. This document will describe the development environment. The environment consists of several parts.

1. The prototyping system.
2. Windows application development.
3. Microcontroller integrated development environment.
4. Calculation and analysis environment.
5. Communications link from Windows application and the prototype.
6. Schematic entry and layout application.
7. FPGA Development Environment.
8. Test Equipment

One thing I need to explain here in the beginning and that is the type of development I have mostly done in my career. Most of my work has been for what I call the niche markets. This is where there is a viable interest in the product; however it is not big enough to attract the attention of the "big boys" who are willing to do a large development project to develop a product that can be produced on a large scale without interruption at low cost. In the niche markets, it is not economic to solve all the problems before manufacture begins and it is reasonable for the line to be down for short periods of time to fix problems as they come up. On large scale production that is not practical. What this means is that the process for an economic development of a large scale product is quite different from that for a small scale or niche product. Further, a large scale "thinker" in a small scale development environment can be disastrous, as can a small scale "thinker" in a large scale development environment.

I have a pretty good set of development skills that cover a wide range of necessary design tasks and this adapts well to the small development environment. I can produce a competent product for small scale manufacture. I cannot and I desire not to compete with a team of specialists in a large scale environment. There are those who prefer the mass market and large scale development. I have enjoyed the work I have done and I am proud of my accomplishments.

Most of my work has been proof of concept work for niche markets. Generally there is not an existing model to work from and it is important to learn as you go. This requires a flexible work environment where changes can be implemented rapidly. This has driven my development of a system and tools to accomplish this competently. I prefer to avoid mass market development and focus on niche market product and to pursue ideas that interest me.

I say this because my system of development might not be applicable to a team of engineers working a large scale product; however I think it will be of interest to the experimenter and the small scale developer. It has served me well and I want to share it.

2. The Prototyping System

As mentioned above the prototyping system is described in another document. It consists of a fast and reliable method of building rugged prototype system circuits.

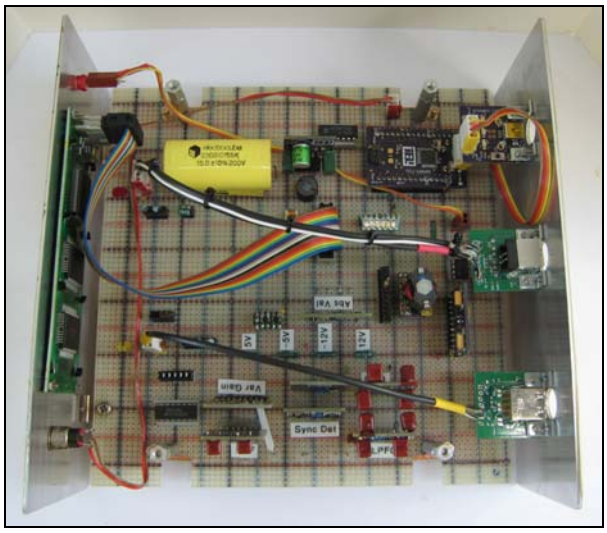

Figure 2-1 Typical Prototype System

The system allows the connection of the various functioning elements in a robust and flexible fashion. This allows the prototype to evolve in a rapid and orderly fashion. It has been my experience that a superior development occurs when a project is allowed to evolve rather be cast in concrete at the beginning. It really helps to incorporate what is learned from the development into the developed system. This can't happen if the beginning of the project is too rigid. My point view of is developed from years of developing systems for the first time, so there is not a model or history to work from. I don't know if what I have said applies that well to an iteration to designed system, unless that system design was a disaster.

3. Windows Application Development

I develop Windows applications that accomplish a number of objectives in systems development. The application provides communication between the prototype and that application. The application can control the prototype and analyze the data coming from the prototype. In some cases, when the Windows application is unnecessary once the development is complete; however it is quite useful during the development. To create the Windows application I use National Instruments Labs Windows and construct code in C. A number of other application development environments could be used.

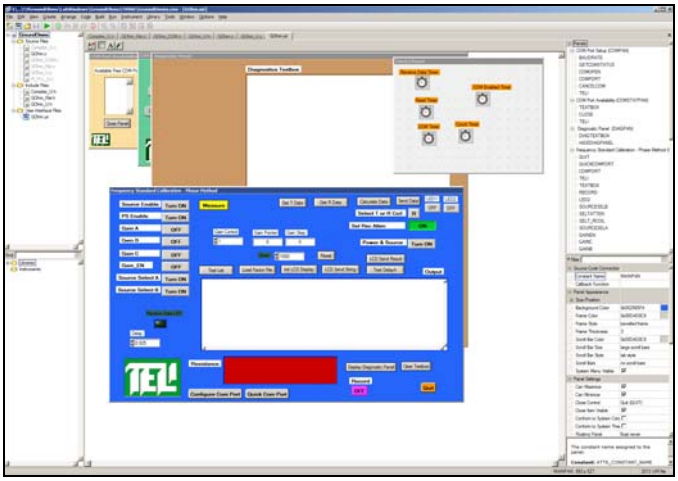

Figure 3-1 Windows Application Development

4. Microcontroller Software Development

Most systems under development include a microprocessor and therefore an environment is very helpful to develop the software. For the time being I have settled on the Atmel AVR processors and Atmel Studio as the development environment. Any number of microprocessors and their development environments could be used as well.

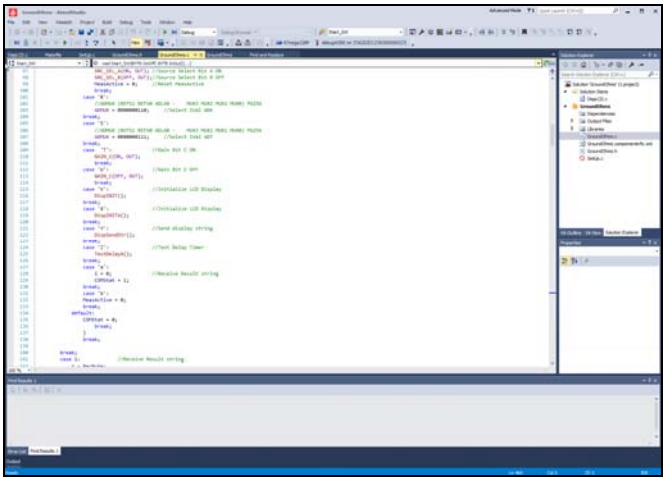

Figure 4-1 Microcontroller Development Environment

5. Calculation and Analysis

It is useful during the development to have the capability to do calculations and analysis of data related to the development. For this I use Mathcad. I like it, I have used it for many years and it has served me well. There are other applications that would work just as well. A lot depends upon what you are comfortable with.

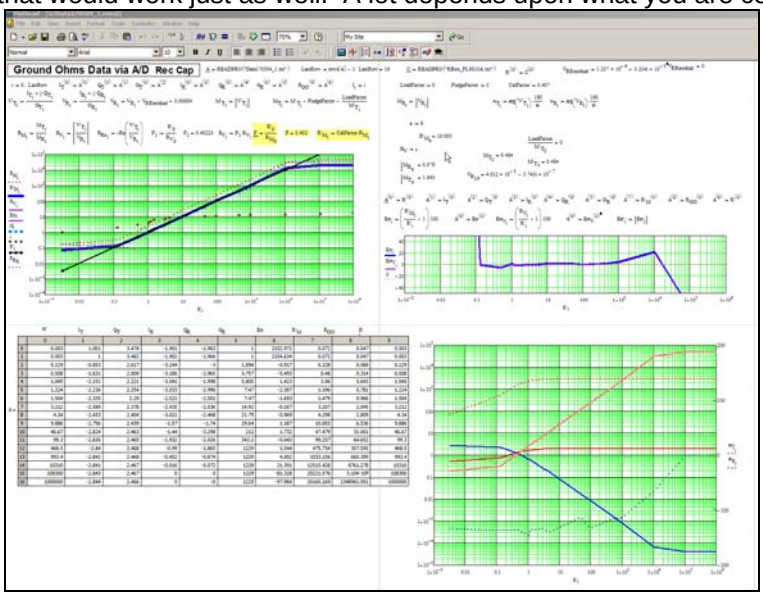

Figure 5-1 Mathcad Analysis

6. Communications Link

For communications I generally use USB to 5V or 3.3V RS232 or in other words a USB COM port. This operates at sufficient speed for most applications. I generally operate at a baud rate of 1.25 MHz. I have built a USB to 5V/3.3V RS232 module that I use with my systems.

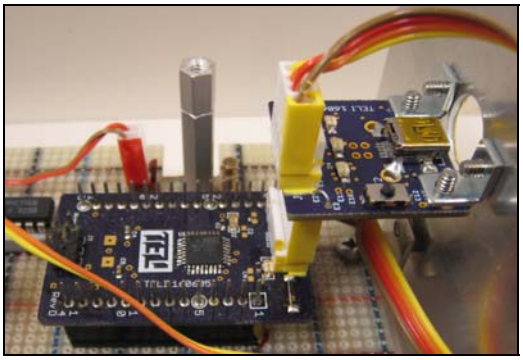

Figure 6-1 USB to RS232 and Microcontroller Modules

7. Schematic Entry and Layout

For schematic entry and layout I use Cadsoft Eagle. It has served me quite well. It allows me to create the original schematics and layouts and then to keep them up-to-date as the system evolves.

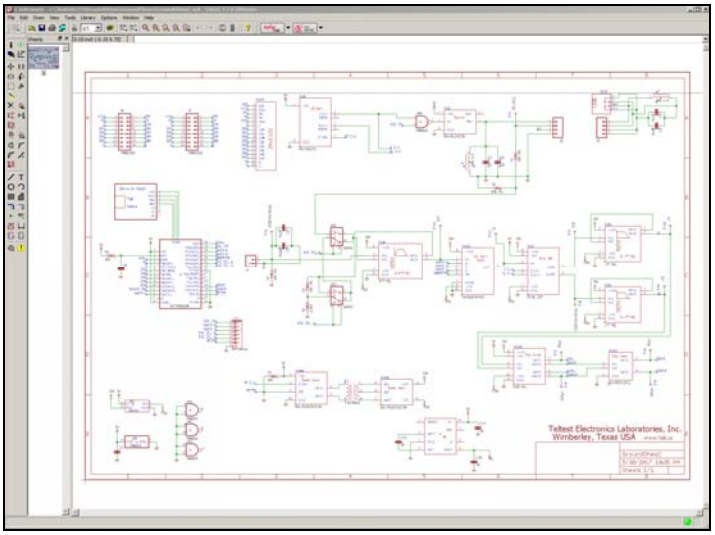

Figure 7-1 Breadboard Schematic

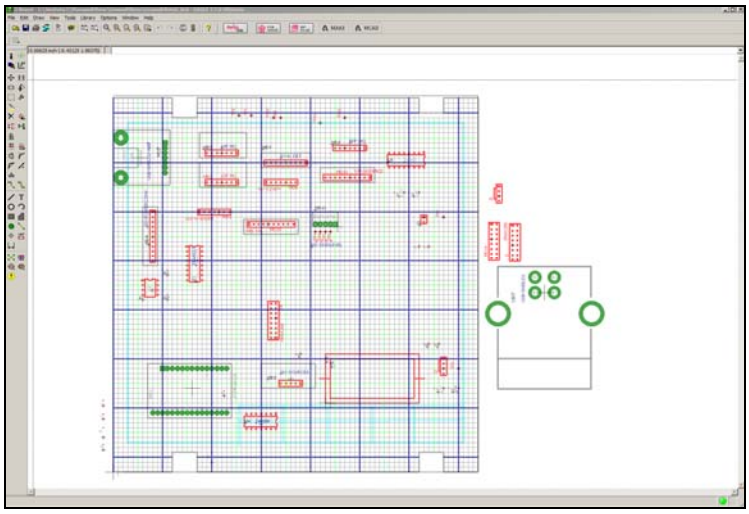

Figure 7-2 Breadboard Layout

8. FPGA

At present I am not using an FPGA or FPGA tool; however in the past this has been useful tool in this environment.

9. Test Equipment

Generally my test environment will involve several pieces of test equipment such as a small digital oscilloscope, a small signal generator and several digital multimeters. Occasionally some more sophisticated equipment such a vector network analyzer or a special signal source will be required.

10. Operation

Operation begins with conceiving the basic design and creating an initial schematic and a layout of the hardware. Once that is complete the next task is to develop and build the hardware. Then is the time to create the basics of the Windows application and the microcontroller software to be able to communicate with the hardware from the application. Once this is operational, it is time to start developing and testing the functional parts of the design. For me this is always a learning process and therefore the system I end up with is not the system I conceived in the beginning and this is a good thing.

All the parts of my development environment operate interactively to facilitate the development process. Changes can be made and evaluated in a real-time environment going from one tool to the other seamlessly. The work flows.

I prefer to at least start the process using the Windows application as the main controller of the system. In this way I have the computing power of the PC computer in a flexible framework. If the completed system does not need that power, the necessary functions can be moved into the microcontroller. Another thing the PC supplies to many of my systems is electrical power via the USB bus. This makes for an easy start.

End of Document

L:\JimData\!\!!Prototype\Docs\ProtoEviron0.02.doc