Open Hardware Test Platform

Project Plan

Team 33
Client Undetermined
Advisor: Dr. Geiger
Team Members/Roles:

- Antonio Montoya - Team Lead
- Christian Hurst - Webmaster
- Ben Wiggins - Communications Head
- Braden Rosengren - Hardware Lead
- Chris Little - Key Concept Holder

Team Email: may1733@iastate.edu
Team Website: http://may1733.sd.ece.iastate.edu/

Revised: 17 Oct 2016

Contents

1 Introduction

1.1 Project statement

We are creating a remote hardware testing platform. The platform should allow users to remotely define and upload tests as well as view and download results. The platform will be built on a Raspberry Pi. The Raspberry Pi will host a web server, and users will interface with the system through a web browser based UI/UX. The Raspberry Pi will be able to deploy, at the user's command, user defined and uploaded test scripts, store and report back test results, and allow the user to download the raw test data. The Raspberry Pi will be able to control and query results from common laboratory equipment like power supplies, digital multimeters, signal generators, and oscilloscopes. The platform will also provide the ability to program/configure digital or mixed signal devices under test via SPI, I^2C, or manually controllable digital pins, all with level shifted outputs at the device's operating voltage.

1.2 Purpose

This platform will allow individuals seeking to perform device characterization, or to automate other laboratory tests, to easily design and implement tests that can be launched and have their results reported remotely. The original concept was conceived with users of wafer probing stations in mind, however the functionality of allowing remote test launch and result reporting has a wide array of usage cases, for example, allowing remote device demonstration for instructive or educational purposes. In addition to allowing for remote access, it provides a cheap platform that can be used to implement identical laboratory setups at multiple locations that support uniform test code, which could be of great use to teams working in parallel to do similar characterization or verification on similar or related devices.

1.3 Goals

In our efforts, we hope to demonstrate the viability of the concept outlined in the Project Statement. The scope of this project is too broad to expect to be able to deliver a polished and robust platform in only two semesters of work. It is however, completely reasonable to target a demonstration of the utility of the platform in an arbitrary usage case, which at the moment is taking measurements from a device on a wafer prober. Proof of concept will therefore be our primary goal, and expanding/enhancing functionality will be a secondary goal. We will produce supporting documentation outlining potential usage cases and documenting the additional work required to adapt the platform to be viable for a broad range of usage cases. Pursuant to our primary goal of producing a proof of concept, some elements of the original project concept may be earmarked as stretch goals.

2 Deliverables

At the completion of this project, we hope to be able to deliver the following items:

- Web Server Hosting System
- Breakout board
- Test scripting API
- Demonstration of successful implementation in at least one proposed usage case as proof of concept
- Outline/Documentation of process for adapting the platform to other potential usage cases
- Documentation of Potential usage cases

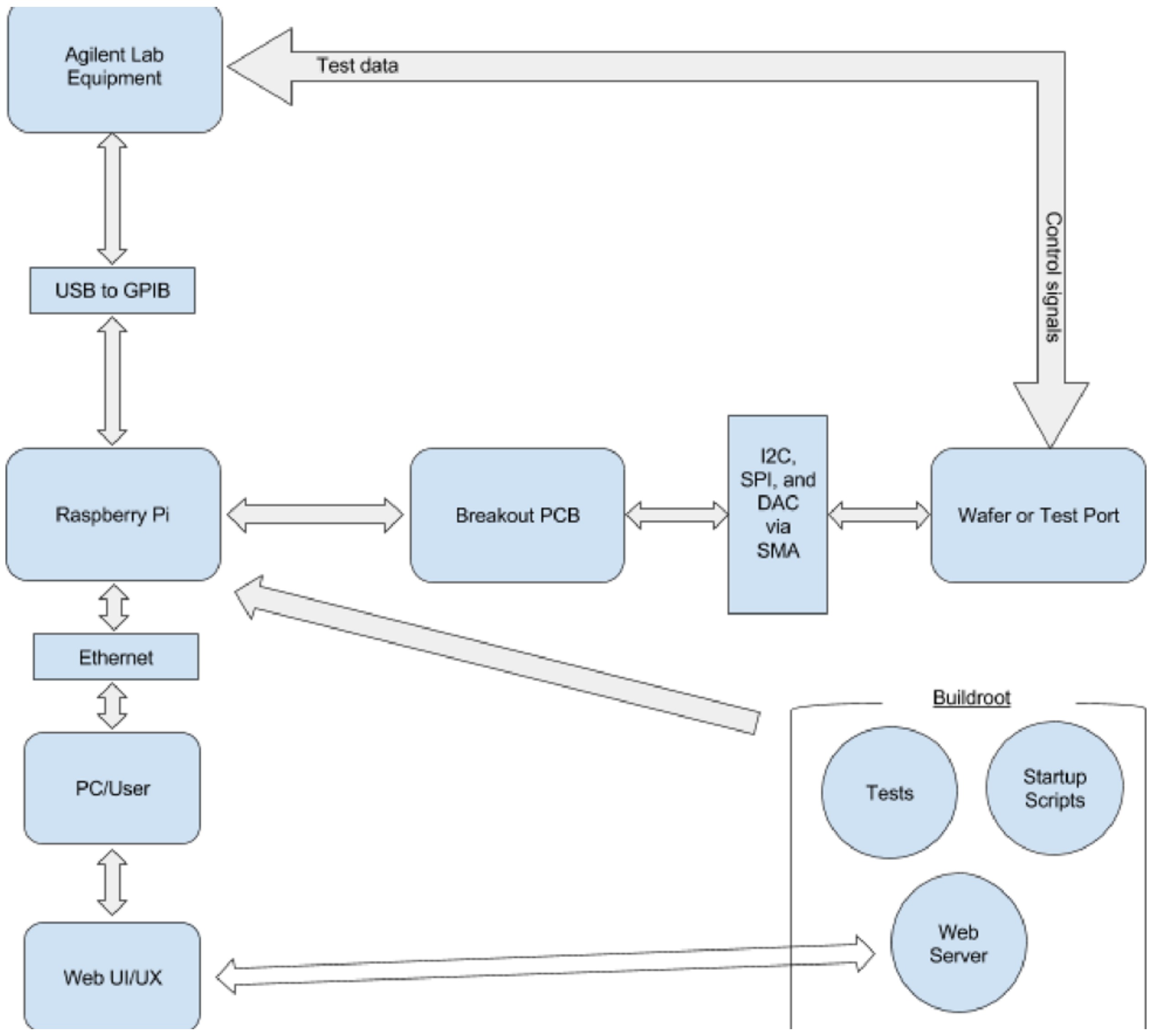
3 Design

The project will be split into hardware and software components to be handled by their respective subteams. The software subteam will handle the creation of the web user interface and the interface with the testing equipment and the onboard ADC and busses. The web UI/UX browser will allow the users to define and upload tests and easily download the results to their work computer. The web content that will displayed in the web browser will be scripted using a lightweight HTTP web server infrastructure. This HTTP web server will interact with the system responsible for scheduling tests, running tests and returning the test results to the user.

3.1 Previous work/literature

This project draws on experience from several of our group members working in industry working with test systems and automation.We will also be taking advantage of numerous programming manuals and other documentation from Agilent, Keysight, and other equipment manufacturers, to assist in developing control schemes for a variety of common laboratory equipment.

3.2 Proposed System Block diagram



3.3 Assessment of Proposed methods

Raspberry PI is a cheap easily accessible development board that our members have had experience with in the past. In addition to this, Buildroot was available which is useful for creating an automated build process and also has a lightweight HTTP web server infrastructure.

We will be utilizing the school's equipment to prototype our control procedures. In addition to power supplies, multimeters and other devices that are accessible in the Coover labs, we will be checking out a GPIB to USB adapter module from the parts shop. We additionally hope to be able to take advantage of the Cascade Systems Wafer Prober in the VLSI/RF lab (Coover 3014) to demonstrate a proof of concept of one of our proposed usage cases.

3.4 Validation

. We intend to take a modular approach to project development, and will be doing testing and verification of each segment/module's functionality in line with the development process. Our goal is to validate the entire system by constructing and executing sample user code for one or two potential usage cases, and demonstrating that the system can deploy user uploaded tests onto attached hardware and report back results in line with expectations, or with test results that can be produced manually. Testable modules and their verification benchmarks are:

- Hosting a user accessible web server.
  - Server can be accessed remotely.
  - All Webpages load on modern browsers
  - Raspberry Pi can handle overhead of web server and background processes.
- Create Robust user interface and user functionality
  - User can upload test code/script.
  - User can view list of available test scripts.
  - User can select to execute an available test script.
- Create Reporting page that hosts results and has formattable display options.
  - User can view test results on results page in format specified by their test code.
  - All past results/data can be viewed and raw data can be downloaded by user.
- Create drivers for communication with lab equipment.
  - System can communicate with and control a basic range of Agilent Devices over multiple connections (LAN,USB,GPIB)
- Create launching program that manages execution of test scripts and storage of results.
  - System can execute user uploaded script and store and report results as directed.
- Create Breakout Board
  - Board allows level shifted SPI, I^2C and binary signal I/O.

4 Project Requirements and Specifications

4.1 Functional

Software Requirements:

- Web Browser based UI/UX
- Webserver
- Ability to define and run tests remotely (With documented user API)
- Raspberry Pi must be able to interface with the test equipment
- Schedule tests and have them run automatically from the Raspberry Pi
- Create software interfaces for the SPI, I2C and ADC
- System should have the ability to start all required processes on power up without human intervention
- Ability to interface with GPIB devices over multiple types of connections (LAN,USB,IEEE-488)

Hardware Requirements:

- PCB breakout board for computer to chip communication
  - SPI and I^2C protocols supported
  - PCB should implement level shifting of digital control signals to acceptable range for use on wafer.
  - Provide outputs via SMA connectors to allow for use with industry standard probe card leads..
- Ability to interface with GPIB
  - GPIB over LAN
  - GPIB over USB
- Create a variable level logic shifter (Optional/Stretch Goal)

4.2 Non-functional

Software Requirements:

- Web interface is well-organized and self-explanatory
- Create instruction manual for building software package

Hardware Requirements:

- Connectors are spaced so that they leave room to easily connect peripherals
- Demonstration of some potential usage cases and proof of concept.

5 Challenges

One of the biggest challenges we face right now is that our funding was pulled out from under us. This may become a self funded project if we either can't get IBM back onboard or can't convince a professor to provide us funding. So far our estimated total cost for three Pi boards, and three revisions of the breakout PCB is $300. So in the event that we are unable to procure funding, this should not be an insurmountable issue.

The usage case that was planned for this platform at its conception was to be used as a controller for tests on a wafer prober station. In order to execute a proof of concept of this usage case we hope to get the Cascade Systems Wafer Prober in Coover 3014 working, but it is currently in a state of disuse and tracking down documentation for its setup and operation, and troubleshooting any problems that may arise may prove to be an involved process unto itself.

A challenge we are looking ahead and seeing on the hardware side of the project is designing a variable level logic shifter, however we have earmarked this as a stretch goal, so we will only tackle this problem if we have excess time after a successful proof of concept.

6 Timeline

6.1 Software Timeline

Semester 1 software timeline	
Week of 10/3	Get PI up and running
Week of 10/10	Implement basic server functionality Compare and choose possible scripting languages
Week of 10/17	Create a basic test runner script that can save files and can do basic file IO
Week of 10/24	Server needs to be able to interact with the test runner
Week of 10/31	Finish Server test runner interaction code. Test IO
Week of 11/7	HTML API Start GPIO API Start GPIB support
Week of 11/14	Implement GPIO API Implement GPIB support
Week of 11/21	Thanksgiving break
Week of 11/28	Put in effort where it is needed
Week of 12/5	Dead Week and Finals Weeks No required work

Semester 2 software timeline	
To be determined once we have a clearer picture of what needs to be accomplished	Testing Stretch goals

6.2 Hardware Timeline

Semester 1 Hardware Timeline	
Week of 10/3	Determine the schematic and layout software that contains the libraries required for this project.
Week of 10/10	Determine modes of communication between Raspberry pi and the test equipment. Also the best uses to breakout the rest of the pins.
Week of 10/17	Researching variable level shifters to 1.5, 1.3, 1.1, 1, and .8 volts. Pinouts to ADC with resolution 10 bits.
Week of 10/24	Revision 1 of PCB needs to be finalized and ordered.
Week of 10/31	Check over design while waiting for PCB.
Week of 11/7	Test PCB
Week of 11/14	Make needed changes to PCB. Reorder PCB
Week of 11/21	Thanksgiving break.
Week of 11/28	Test Revision 2 of PCB
Week of 12/5	Dead Week and Finals Week Documentation and Presentation.

Semester 2 Hardware Timeline	
Week of 1/9	
Week of 1/16	
Week of 1/23	
Week of 1/30	
Week of 2/6	
Week of 2/13	
Week of 2/20	
Week of 2/27	
Week of 3/6	
Week of 3/13	
Week of 3/20	
Week of 3/27	
Week of 4/3	
Week of 4/10	
Week of 4/17	Finalize Project and Documentation
Week of 4/24	Present Project to Whomever

7 Conclusions

We are creating a remote hardware testing platform that will allow users to remotely define and upload tests as well as view and download results. The platform will be built on a Raspberry Pi with a breakout board and open-source software. It will be able to communicate with test equipment over GPIB. Additionally, the platform will have an ADC, as well as I^2C and SPI buses exposed via the breakout board to allow it to further interface with the hardware under test.

8 References

- Buildroot documentation - https://buildroot.org/downloads/manual/manual.html
- Installing operating system images on Linux - https://www.raspberrypi.org/documentation/installation/installing-images/linux.md