Paper ID #25860

Students' Familiarization to Methodical Troubleshooting through a Capstone Project

Mr. Peter D. Huerter, Purdue University Northwest

Students' Familiarization to Methodical Troubleshooting through a Capstone Project

Abstract - This paper explains the technical training and methodical troubleshooting methods acquired through the entirety of the Outcome Based Education approach to Electrical Engineering Technology coursework promoted at Purdue Northwest. Many engineering students are focused mainly on theoretical coursework with some structured laboratory experiments. However, troubleshooting of equipment and complex machines are not given sufficient attention for a typical industrial setting during regular engineering coursework. Purdue University Northwest's Outcome Based Education allows students to gain hands-on experience troubleshooting complex circuits, machines, and their subsystems.

In order to familiarize students with troubleshooting and identifying equipment failures, the resurrection of a relatively complex and non-functional NovaMill 3-Axis CNC Milling Machine is selected as a Capstone Senior Design project. The objectives of this project include identifying the different sub-systems of the machine, isolating each sub-system, testing and documentation of initial status, identification of failed sub-systems, repair the failed sub-system, integrate the overall machine, and perform final testing on the completed machine. A procedure will be developed to teach future students similar methodology to identify the failed components of a complex machine and finding solutions to fix it.

The NovaMill 3-Axis CNC Milling Machine explored in this paper is a complex machine comprised of Stepper Motors for each of the three degrees of movement, GeckoDrive Stepper Motor Drivers, a multiple-output voltage AC Power Isolation Transformer, twin AC-DC Voltage Rectifiers, an Automatic Tool Changer, and a high-torque Spindle Motor along with peripheral electrical components such as fused terminal blocks and control relays.

The original Novamill was controlled by an obsolete and proprietary PC-based control system, while the newly repaired and retrofitted machine explored in this paper was modified with a Linux-based control system running on a BeagleBone Black Microcontroller with a communication link for interfacing. Implementation of the BeagleBone Microcontroller was developed to allow remote communication link over Ethernet or direct control with keyboard, mouse, and local display. Understanding and repairing of such a machine requires a holistic understanding of each sub-system and the ways each of the systems interact.

I. INTRODUCTION

Companies often have hard choices to make with regard to the purchase and maintenance of expensive production equipment. Over time, even the best equipment purchased today will become worn down from the stresses of use, and the control system will become obsolete and difficult to interface with newer technology. Frequently, parts manufacturers go out of business or are acquired by larger companies, which can lead to unavailability of direct replacement parts.

One of the key skills for any manufacturing-centered engineer is the ability to troubleshoot legacy equipment that has been modified by persons of varying skill proficiencies and in conditions where proper documentation of updates to the core systems is nonexistent. If components of the legacy equipment are found to be nonfunctioning and no longer available, a skilled engineer can save a company tens or even hundreds of thousands of dollars by finding newer replacements to obsolete components and integrating them into the legacy system. Every machine can be simplified to a series of inputs and outputs, with a control system designed to meet ever-changing needs and increase end-user functionality.

The objective of this paper is to describe in detail the conative steps and methods an engineer must take to troubleshoot a nonfunctioning legacy machine. A Denford NovaMill 3-Axis CNC Milling Machine is considered to demonstrate the specific troubleshooting methods required to diagnose the various problems preventing full functionality.

II. INITIAL SYSTEM CONDITION AND HISTORY

The NovaMill CNC 3-Axis Milling Machine was originally placed in service in February 1999 according to a nameplate found on the side of the control cabinet. The mill was operated by a proprietary printed circuit board based controller connected to a desktop computer running proprietary software via a RS-232 parallel port. Mechanical specifications for the CNC machine are shown in Table 1 below:

Table 1 Mechanical Specifications

Working Table Surface:360mm x 130mm (14" x 5.125")	Longitudinal Travel (X): 225mm (9")
Cross Travel (Y):150mm (6")	Head Travel (Z): 115mm (4.5")
Machinable Area: 225mm x 115mm (9" x 4.5")	

The original computer control software on the machine was no longer functioning, which left the NovaMill CNC machine inoperable. Desiring a control upgrade, the NovaMill was given to a pair of students as a Senior Design project in the fall of 2017 through spring 2018 [1]. The students followed a guide found online for a very similar Denford NovaMill CNC machine to remove the legacy printed circuit board based controller and replace it with a BeagleBone Black Microcontroller, a low-cost, community-supported development platform that runs Linux CNC [2-4]. The BeagleBone Black was interfaced with the NovaMill CNC control panel by utilizing a Probotix Breakout Board and installing the Linux Debian-based distribution of Machine-Kit on the BeagleBone Black.

III. GENERAL TROUBLESHOOTING METHODOLOGY

Troubleshooting, at its core, is a logical and systematic search for the source of a problem. Troubleshooting is always initiated as a result of some sort of abnormal operating condition of a machine or system. Before any effective troubleshooting can occur, it is critical to first understand the functionality of the machine. Any machine can be broken down into a series of inputs and outputs, with specific inputs to the machine generating specific outputs or results. Any unexpected output for a given input may be a symptom of a larger problem and must be analyzed. A hypothesis is developed as to why the machine's desired outputs are not matching the given inputs and data is gathered to evaluate the initial hypothesis.

To better understand the internal logic and workings of any machine, examining any engineering documentation that may be available, including manufacturer's control drawings, schematic diagrams, and system manuals cab be the first step. Inspecting individual components of the larger machine looking for manufacturer's information and serial numbers and searching for online manuals can be very helpful. After analyzing the engineering documents, take the time to interview any recent operators of the machine. Verifying the system operation from the perspective of the operator, and recording any unusual sounds, vibrations, or functions they may have observed is essential. A regular machine operator often understands their equipment's quirks better than any information that can be found in a manual. For the purposes of this project the previous students who worked on this machine were consulted, along with their project advisor. After gathering information on the machine, the process of effective troubleshooting generally follows three phases: (1) problem representation, (2) fault isolation, and (3) solution verification [5].

1. <u>Problem Representation</u>

Problem representation can be assisted by modelling the machine in the form of a high-level functional diagram, as shown in Figure 1 below:

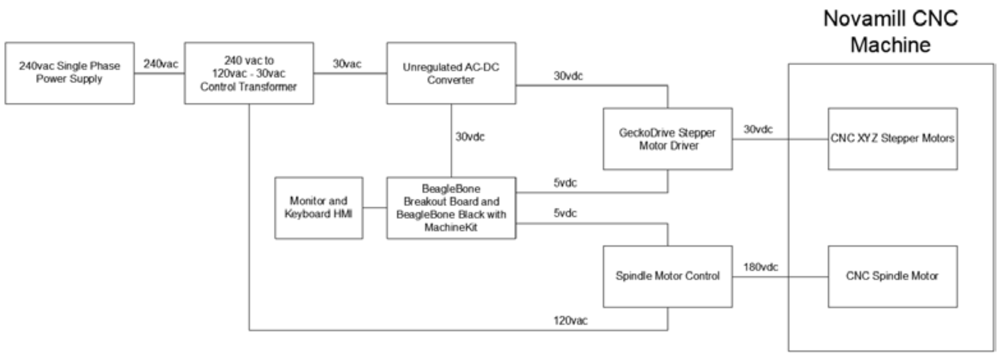


Figure 1: NovaMill Functional Diagram

A functional diagram as shown in Figure 1 helps to display the interconnectivity and signal flow throughout a machine's various subsystems. Noting expected signals, power sources, and voltage levels throughout the machine on the diagram can be critical when trying to troubleshoot complex problems.

2. <u>Fault Isolation</u>

A hypothesis to assess functional problems with a system can be categorized through four discreet levels of analysis: (1) system, (2) subsystem, (3) device, and (4) component.
A system-level hypothesis does not reduce the problem beyond the entire system or machine exhibiting irregular behavior. An example of this, for a water pumping system, could be as simple as, "There is no water flow." While a system-level hypothesis does state the problem, it typically is not very useful for directly discovering a solution. Developing a subsystem-level hypothesis for the aforementioned water system could include: "Maybe the valve is closed", or also, "Maybe the pump isn't working." When choosing which line of investigation to pursue, the best practice is to follow the hypothesis that first is the most likely based on observed system behavior, and second is the least time-consuming to verify. From the subsystem hypotheses given in this example, pumps tend to have a fairly long service life while valves are handled relatively frequently. Additionally, checking a valve may take only a few seconds while troubleshooting a pump could take many hours. If a quick check of the valve shows it to be open, a device-level hypothesis will need to be created which could include: "Maybe the pump is broken or blocked", "Maybe the pump prime mover motor is damaged", or "Maybe the power to the pump motor is off". Following the most likely and simple hypothesis, the power to the pump motor should be checked.

Finding a lack of power at the pump motor, a component-level hypothesis could include: "Maybe the breaker is off or tripped", or also, "Maybe the wires to the pump motor have been damaged or cut." Following the most likely and simple hypothesis, the breaker should be checked. Finding it off and not tripped, the most likely solution to the problem of no water flow is that someone turned the breaker off.

3. <u>Solution Verification</u>

When taking corrective action during troubleshooting, it is critical to first perform a safety assessment before altering the state of the machine. Continuing with the water pumping system example, finding a breaker off does not mean that it should simply be reset without first examining why the breaker was off in the first place. It is possible that a technician on another shift may have deliberately disabled the power for a particular reason, or it may have been mistakenly turned off. A system should always be checked to ensure that all tools and material have been removed and personnel are safe before any sort of power source is reinstated.
When making repairs to a complex system that does not have many obvious problems it is always best troubleshooting practice to change one variable at a time, re-test the system, and if needed form a new hypothesis based on the new system behavior. Changing multiple variables before verifying the solution can result in unnecessary replacement of components and misidentification of the scope of the problem.

IV. IDENTIFICATION OF SYSTEM COMPONENTS

To better evaluate the machine, manufacturer's diagrams for the Control Cabinet were analyzed, and the NovaMill was partitioned into the following key subsystems for evaluation:

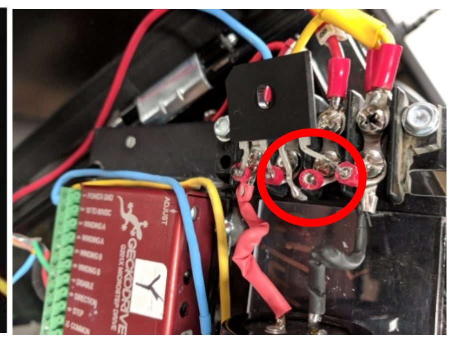
1) Unregulated Power Supply
2) BeagleBone Black and Probotix Breakout Board
3) GeckoDrive G201X Stepper Motor Drivers (X, Y, Z Axes)
4) Stepper Motors (X, Y, Z Axes)
5) Spindle Motor and Spindle Motor Control

The auxiliary systems such as the electro/pneumatic Material Vice, Coolant System, and the electrically-driven Automatic Tool Changer were excluded from evaluation in favor of focusing on the more critical systems of Power Supply, Axial Motion Control and Spindle Motor Control. From the initial analysis of the Control Cabinet and components of the NovaMill, a high-level functional diagram was developed (Figure 1) that detailed the interconnectivity of the key subsystems and the expected voltage levels of the power and signals that pass between the systems. All control cabinet wiring was carefully traced and evaluated against the available drawings. The previous students who worked on the NovaMill and their advisor were interviewed to determine any operational problems during their previous testing, and the student's report [1] was analyzed for any insight into the potential faults. All specific troubleshooting methodology used to evaluate this NovaMill CNC Machine is included in part V of this paper.

V. FAULT ISOLATION AND SOLUTION VERIFICATION

1. <u>System Analysis: Unregulated Power Supply</u>

Based on a conversation with the previous students who worked on the machine, a subsystem-level hypothesis was formed that there was a problem with the Unregulated Power Supply that they had constructed (Figure 2) due to their description of a brief moment of operation before the system stopped working, and the BeagleBone Black had become unresponsive. The Unregulated Power Supply was first visually examined. The Unregulated Power Supply is split into two separate full-wave rectifiers, each constructed of one Vishay GBU8B-E3/51 Diode Bridge Rectifier to convert the AC power to DC power and one Panasonic 15000 microfarad electrolytic capacitor to minimize output ripple. The power supply converts two separate 32.5 VAC outputs from the existing control voltage transformer to two separate 40 VDC power outputs, with the negative terminals on the VDC side of the two power supplies bonded to each other. This configuration was recommended in the guide the students had followed [2] to supply DC power to the Stepper Motor Drivers and the BeagleBone Black through the Probotix Breakout Board. Visual examination of the Unregulated Power Supply resulted in the discovery of loose terminal connections on the screw terminal blocks, as well as broken solder joints (Figure 3) connecting the Bridge Rectifiers to their terminal lugs. Due to these findings, a device-level hypothesis was created that the loose terminals and broken solder joints on the Bridge Rectifier had led to unstable voltages and currents to the BeagleBone Black and Stepper Motor Drivers.

Figures 2 and 3 Dual Unregulated Power Supply and Rectifier Broken Solder Joint

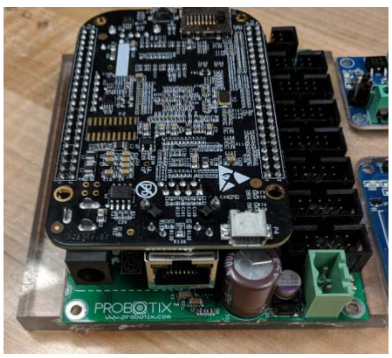
A component-level hypothesis was further generated that the Bridge Rectifiers may had been damaged from the unstable voltages and currents and may need to be replaced. The Bridge Rectifiers were isolated by removing them from the circuit entirely, and bench tested with an oscilloscope, multimeter, and 24VAC Power Supply. A separate control transformer was used to supply the bridge rectifier AC input terminals with 25.05VAC, and 33.43VDC was measured at the DC output terminals. These measurements were verified with the following calculation ($25.05VAC * 1.4 = 35.07VDC$). As the measured DC voltage was roughly equivalent to the calculated DC voltage, the output voltage was determined to be within operational range. To further verify functionality, the Bridge Rectifier output waveform was analyzed with an oscilloscope, and was found to match the expected waveform for a full-wave bridge rectifier.

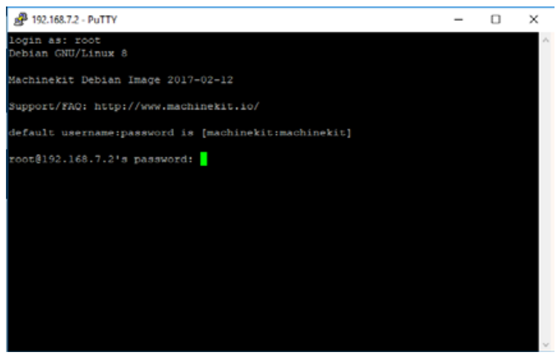
a. <u>Solution Verification: Unregulated Power Supply</u>

The above findings disproved the hypothesis that the bridge rectifiers had been damaged from unstable currents, although the decision was made to replace the bridge rectifiers anyway to safeguard against possible future failure. This choice was made due to the fact that the bridge rectifiers had been part of a system that had been faulted, were already removed from the circuit, and were relatively inexpensive to replace at only $1.42 each.

2. System Analysis: BeagleBone Black and Probotix PBX-BB Breakout Board

Recalling the fact that the previous students had listed the BeagleBone Black as unresponsive, a subsystem-level hypothesis was developed that due to the discovered problems with the unregulated power supply the BeagleBone may have been damaged. The BeagleBone was isolated completely from the control circuitry and disconnected from the Probotix breakout board. The connection between the BeagleBone Black and the Probotix breakout board is detailed in Figure 4. A connection was made between the USB port on the BeagleBone and a USB port on a PC. PuTTY, an open-source terminal emulator, was launched on the PC to attempt to establish an SSH (The SSH protocol, also referred to as Secure Shell, is a method for secure remote login from one computer to another). connection to the BeagleBone. The Beaglebone connected to the PC and displayed the boot screen in the PuTTY terminal (Figure 5) but was completely unresponsive to entry of the default MachineKit password: "machinekit".





Figures 4 and 5 BeagleBone Black Connected to Probotix BBX-BB Breakout Board and Attempt to Connect to Existing BeagleBone via PuTTY

Using PuTTY established that the BeagleBone could handle an SSH connection but was unresponsive to input. WinSCP, software commonly used for file transfer between a local and remote computer, was used to establish an SSH connection to the BeagleBone. The Beagle Bone's directory system was explored and found to be empty of data. These findings supported the hypothesis that power supply instability may have corrupted the BeagleBone.

The connection between the unregulated power supply's 40VDC output and the BeagleBone Black's 5VDC input is made through a Probotix PBX-BB breakout board. A device-level hypothesis was formed that the voltage regulator integrated in the breakout board may have been damaged by the loose connections and broken solder joints on the unregulated power supply. According to the Probotix PBX-BB Wiki page, the breakout board utilizes a MAX5090 based 5V /2A switching regulator (Figure 6) that accepts voltages up to 76VDC, and the 5V rail is bused out to the breakout header pins [6].

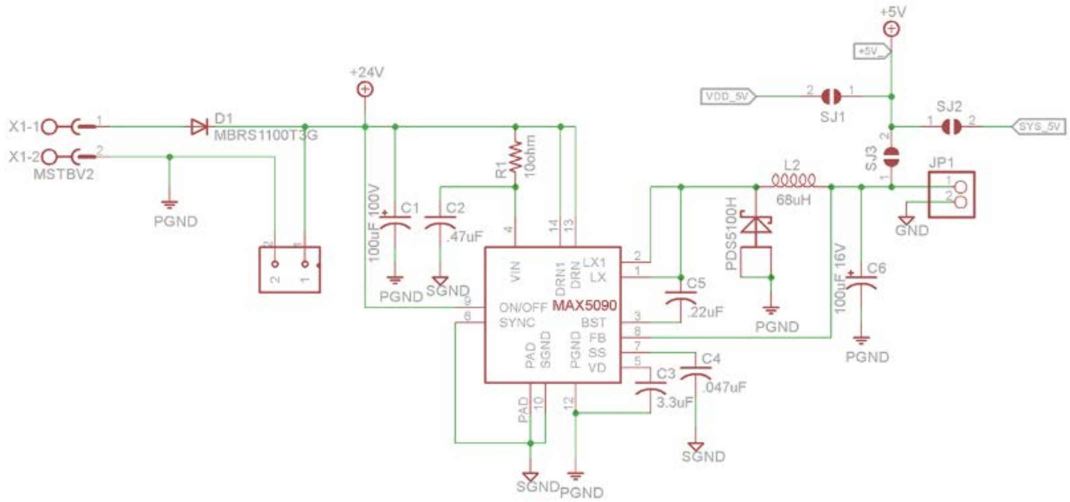
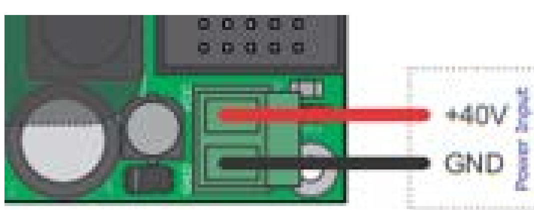
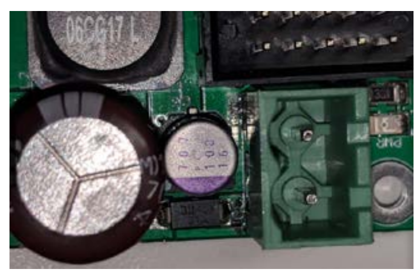


Figure 6 BPX-BB Breakout Board Voltage Regulation Circuit

As most of the voltage regulation circuit shown above in Figure 6 was integrated into the PCB layers, direct examination of the circuit was impractical. Visual examination of the voltage input terminals of the Probotix breakout board found that there was a mismatch between the markings of the VCC and GND terminals on the board and the order specified on the manufacturer's wiring diagrams. A component-level hypothesis was developed that the previous students may have incorrectly followed either the manufacturer's documentation or the board terminal markings when connecting power to the board, potentially damaging the breakout board's MAX5090 voltage regulator. Due to examination of the circuit shown above in Figure V.2.3 this hypothesis was ruled unlikely, but possible and thus worth pursuing. Due to the construction of the Breakout Board, it was difficult to determine if the markings on the board or the manufacturer's wiring diagrams were correct, shown in Figures 7 and 8.





Figures 7 and 8 BPX-BB Power Terminal Wiring Diagram, GND marked as the bottom terminal

The functionality of the Probotix breakout board's MAX5090 switching voltage regulator was tested by applying 31.97VDC sourced from a benchtop power supply to the breakout board's voltage input terminals while measuring the output pins of the breakout board that supply power to the BeagleBone Black. The DC voltage was initially connected with the polarity matching the markings on the board as it was rationalized that the markings on the board were created at the time of manufacture and passed quality control checks, while the wiring diagram on the Probotix PBX-BB Wiki webpage may have been updated to reflect a new design. With the input DC voltage matching the polarity of the markings of the board, there was no measured voltage on the output pins. The polarity of the input DC voltage was changed to match the wiring diagram and with 31.97VDC applied to the input terminals, 7.36VDC was measured on the breakout board 5 VDC output pins. According to BeagleBone.org, the maximum voltage that may be applied to the BeagleBone's power terminals is 5.2VDC [7]. If a voltage greater than 5.2VDC is applied to the power terminals, the BeagleBone will not power up. These findings further supported the hypothesis that the voltage regulator integrated in the breakout board may have been damaged by the loose connections and broken solder joints on the Unregulated Power Supply, and the increased voltage output from the breakout board regulator was preventing the BeagleBone from operating properly.

a. <u>Solution Verification: BeagleBone Black and Probotix Breakout Board</u>

A new Probotix PBX-BB breakout board was procured and tested before installation. The second Probotix breakout board exhibited the same marking mismatch between the power terminals and the manufacturer's documentation, although a slip of paper was included in the packaging that stated this as a known issue. The board was bench tested with 31.97VDC applied to the input terminals and 5.14VDC was measured on the breakout board 5VDC output pins. A new BeagleBone was connected to the breakout board and verified to power on. Utilizing the guide provided to the previous students [3], a distribution of MachineKit was flashed to a new BeagleBone.

3. <u>System Analysis: GeckoDrive G201X Stepper Motor Drivers (X, Y, Z Axes)</u>

Due to the fact that damage was found to the BeagleBone subsystem from its connection to the faulty unregulated power supply subsystem, a subsystem-level hypothesis was developed that the three GeckoDrive G201X stepper motor drivers installed by the previous students [1] may have been damaged as a result of unstable voltages and currents from the problems discovered with the unregulated power supply as described in section V, part 1 of this paper. Visual examination of the stepper motor driver terminal wiring produced a notable find: An unnecessary connection had been made by the previous students between each of the X, Y, and Z Driver's 18 to 80VDC power supply terminal and the Probotix breakout board's header pins, shown in Figure 8.

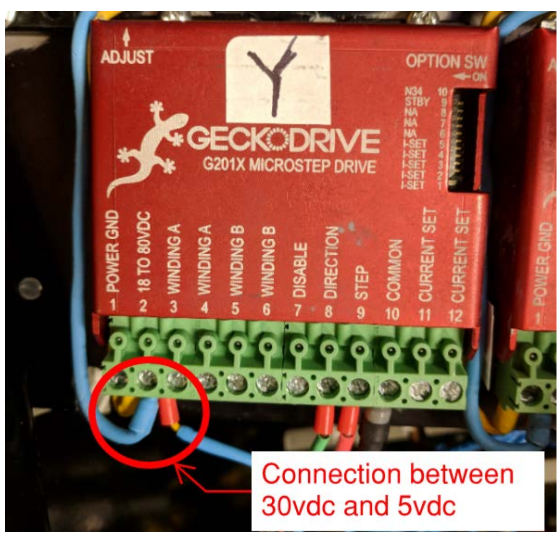
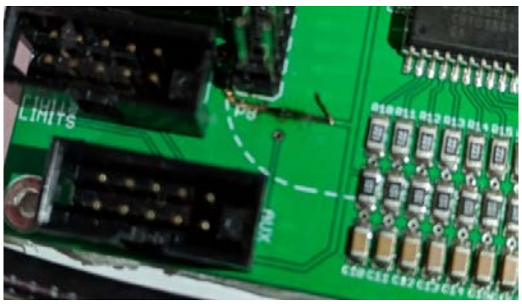
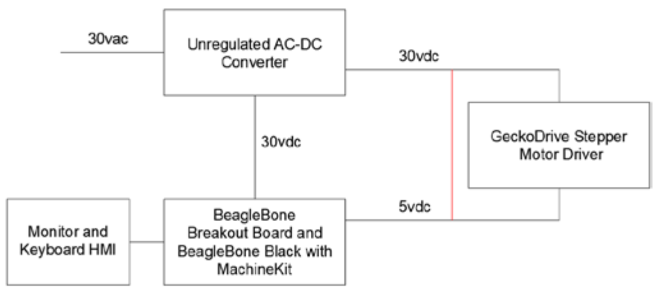


Figure 8 Unnecessary Connections on Stepper Motor Driver

In the red circle shown in Figure 8 above, the wire with the blue ferrule is connected between terminal 2 on the stepper motor driver and the +40 VDC terminal of the unregulated power supply, and the wire with the red ferrule is connected between terminal 2 on the stepper motor driver and the Probotix breakout board header Pin Connector. This was noted as unusual, as there should not have been any reason for the stepper driver supply voltage to be connected to the breakout board. A decision was made to study the Probotix breakout board in greater detail to discover what effect if any this connection had made on the circuitry and components of the breakout board and BeagleBone Black. Closer examination of the Probotix breakout board found a completely burnt-out trace in the board shown in Figure 9. The burnt trace was continuity-tested to all of the Breakout Board's pins and was found to correspond to the 5VDC rail that is bussed to its breakout header pins and connects to the BeagleBone power supply pins. Electrically, a connection had been made as shown with the red line in Figure 10.





Figures 9 and 10 BPX-BB Burnt-Out Trace and Effective Electrical Connection due to mis-wiring on Stepper Motor Driver

This connection on the driver's power supply terminal would result in the full 40VDC supplied to the stepper drivers completely bypassing the breakout board's MAX5090 voltage regulator and applying 40VDC to the 5VDC power supply pins to the BeagleBone Black. This finding was determined to be the primary contributing factor in the problems discovered with the BeagleBone Black and Probotix breakout board discussed in section V, part 3 of this paper.

a. <u>Solution Verification: Stepper Motor Drivers (X, Y, Z Axes)</u>

The GeckoDrive stepper motor driver functionality was first tested by connecting each of the three drivers to a single stepper motor, the Z-Axis stepper motor, that was tested and found to match the manufacturer's information as described in section V, part 4 of this paper. The Driver was supplied 40VDC from the repaired unregulated power supply, and the signal inputs were connected to an Arduino Uno to supply a digital on/off signal to toggle the driver's direction input and a pulse width modulated signal to provide a drive signal to the driver's step input. A program was written for the Arduino Uno to drive the stepper motors 4000 pulses, then switch direction and drive another 4000 pulses, repeating infinitely. The Z-Axis stepper motor was first connected to the Z-Axis driver, with the Arduino Uno supplying the drive signals to the driver. The Z-Axis driver was verified to operate the Z-Axis stepper motor as programmed. Next, the Z-Axis stepper motor was connected to the Y-Axis driver, with the Arduino Uno supplying the drive signals to the driver. The Y-Axis driver was verified to operate the Z-Axis stepper motor as programmed. Last, the Z-Axis stepper motor was connected to the X-Axis driver, with the Arduino Uno supplying the drive signals to the driver. The X-Axis driver was verified to operate the Z-Axis Stepper motor as programmed. Having verified that the X, Y, and Z-Axis stepper motor drivers all responded appropriately to input drive signals from the Arduino Uno and operated the Z-Axis stepper motor, the stepper motor drivers were connected to their respective stepper motors. The Arduino Uno was again connected to each of the stepper motor drivers, and each driver was found to operate its matching stepper motor. This configuration of Arduino Uno, stepper motor driver, and stepper motor was used throughout the testing of the stepper motors in Section V, part 4. The above findings disproved the hypothesis that the stepper motor drivers may have been damaged as a result of unstable voltages and currents from the problems discovered with the unregulated power supply, although problems with the wiring of the stepper motor driver were a major factor in the damage to the Probotix breakout board and BeagleBone Black.

4. <u>System Analysis: Stepper Motors (X, Y, Z Axes)</u>

Due to the fact that irregularities were found with the power supply wiring to the stepper motor drivers in section V part 3 of this paper, a subsystem-level hypothesis was developed that the X, Y, and Z-Axis stepper motors may have become damaged while the system was handled by the previous students. The X, Y, and Z-Axis stepper motors were visually examined. Comparisons between the Stepper Motor Data Sheet values and observed or measured values are shown in Tables 1, 2, and 3 below.

Table 1 X-Axis Stepper Motor Data Sheet / Observed or Measured

Stepper Motor X	Data Sheet	Observed Value
Model Number	103-770-1640	103-770-1647
Step Angle	1.8°	1.8°
Current Rating	1.4A	1.4A
Winding Resistance OR-BK	2.6 Ω	2.57 Ω
Winding Resistance YL-RD	2.6 Ω	2.57Ω
Rotation Verified	yes	yes

Table 2 Y-Axis Stepper Motor Data Sheet / Observed or Measured

Stepper Motor Y	Data Sheet	Observed Value
Model Number	103-770-1640	103-770-1647
Step Angle	1.8°	1.8°
Current Rating	1.4A	1.4A
Winding Resistance OR-BK	2.6 Ω	2.87 Ω
Winding Resistance YL-RD	2.6 Ω	2.68 Ω
Rotation Verified	yes	yes

Table 3 Z-Axis Stepper Motor Data Sheet / Observed or Measured

Stepper Motor Z	Data Sheet	Observed Value
Model Number	103-807-6341	103-807-0645
Step Angle	1.8°	1.8°
Current Rating	3.05A	3.05A
Winding Resistance OR-BK	0.95 Ω	2.25Ω
Winding Resistance YL-RD	0.95 Ω	2.26Ω
Rotation Verified	yes	yes

The manufacturer did not have data sheets available with model numbers that were a perfect match to the model numbers found on the X, Y, and Z-Axis stepper motors, so a near-equivalent motor from the same series was chosen for comparison. Finding the measured winding resistances to be near the Data Sheet values, the stepper motors were passed to a function test found in section V part 4a of this paper.

a. <u>Solution Verification: Stepper Motors (X, Y, Z Axes)</u>

Having verified the functionality of the stepper motor drivers in section V part 3 of this paper, a test was constructed to assess each stepper motor's performance over time. Each stepper motor was mechanically decoupled from its load, and driven by its designated stepper motor driver for 60 seconds (No-Load Test). The temperature of the casing of the stepper motor was measured in 15 second intervals and recorded and shown in Figures 11, 12 and 13. Next, each stepper motor was given an hour to cool to ambient temperature. After verifying that the stepper motor had cooled to ambient, the stepper motor was recoupled to its load and driven by its designated stepper motor driver for 105 seconds (Load Test). The temperature of the casing of the stepper motor was measured in 15 second intervals and recorded and shown in Figures 11, 12, and 13. The time-interval for both the no-load and load tests was originally designed at 60 seconds. As the X-Axis Stepper was tested, some problems discussed in the observations found after Table 1 became apparent around 55 seconds into the load test prompting the load test for all three Stepper Motors to be extended to 105 seconds.

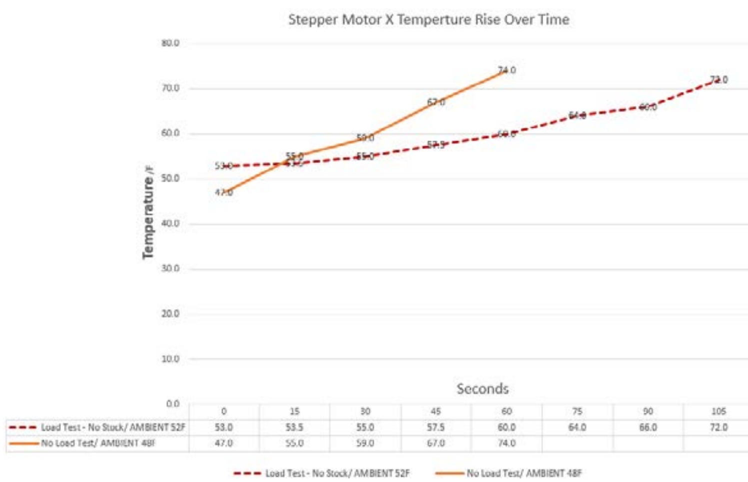


Figure 11 X-Axis Stepper Motor Test Results

X-Axis stepper Motor observations included a random intermittent grinding sound that occurred under both no-load and load conditions, along with increased motor noise while operating. Temperature rise on the stepper motor increased sharply when the grinding sound began to 19 degrees Fahrenheit over 105 seconds. The X-Axis stepper motor was able to drive the X-Axis of the CNC machine.

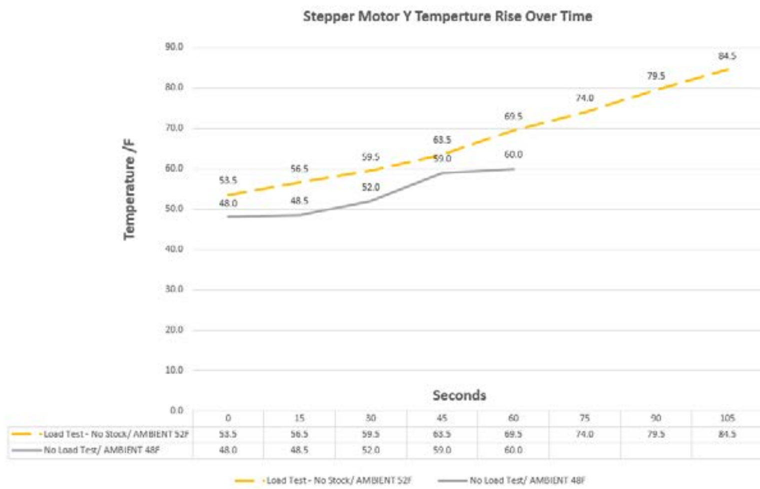


Figure 12 Y-Axis Stepper Motor Test Results

Y-Axis stepper motor observations included increased motor noise while operating. Temperature rise on the stepper motor was 31 degrees Fahrenheit over 105 seconds, although the Y-Axis stepper motor did not exhibit a grinding sound like the X-Axis stepper motor. The Y-Axis stepper motor was able to drive the Y-Axis of the CNC machine.

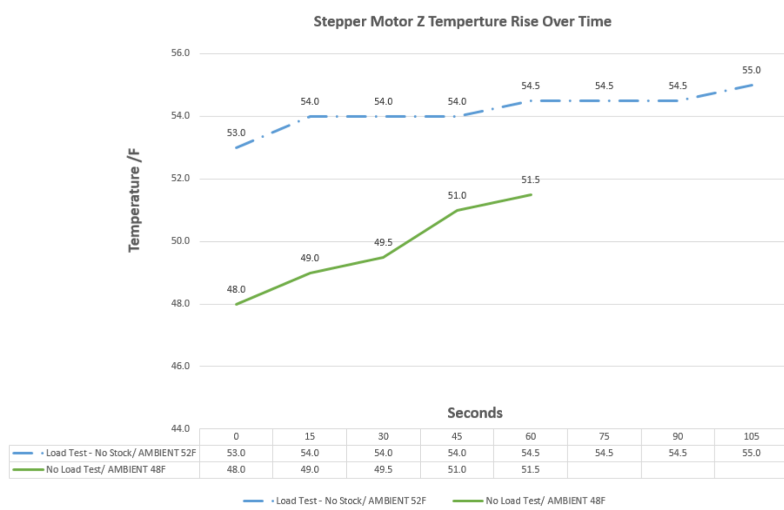


Figure 13 Z-Axis Stepper Motor Test Results

Z-Axis stepper motor observations included very quiet motor noise while operating. Temperature rise on the stepper motor was 2 degrees Fahrenheit over 105 seconds, and the Z-Axis stepper motor did not exhibit a grinding sound like the X-Axis stepper motor. The Z-Axis stepper motor was able to drive the Z-Axis of the CNC machine.

The above findings proved that due to the large temperature rise found on the X and Y Axis Stepper Motors, the motors are likely due to fail soon. The Z-Axis stepper motor appears to be in proper operating condition.

5. System Analysis: Spindle Motor and Spindle Motor Control

Analysis of the previous student's report [1] compared to the wiring found in the control panel determined that the previous project had not gained control of the motor that rotates the CNC Machine's tooling, the Spindle Motor.

The Spindle Motor Control Board (Figure 14, below) was examined, and all wiring connections were checked against the Novamill Control Panel drawings shown in Figure 15.

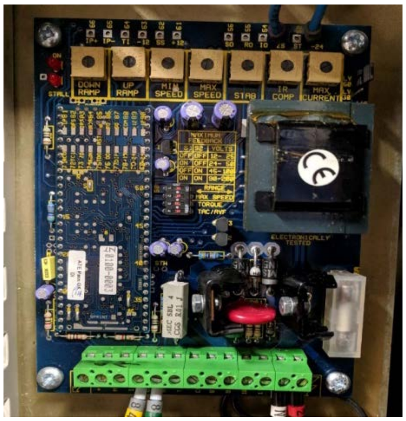


Figure 14: Spindle Motor Control Board

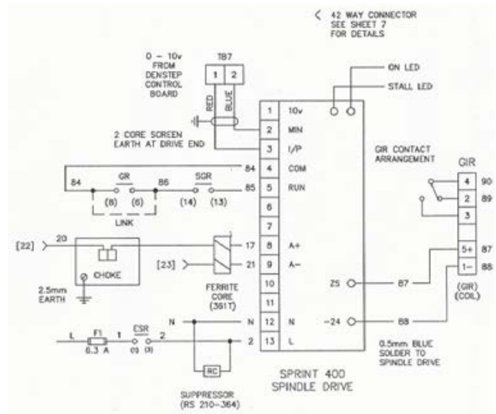


Figure 15: Spindle Motor Control Board Wiring

As most of the control wiring to the Spindle Motor Control Board detailed in Figure 14 had been removed and not replaced by the previous students, and based on previous findings throughout this project while troubleshooting the power supply and stepper motors, a subsystem-level hypothesis was developed that the spindle motor and spindle motor control board may have been damaged by previous attempts to gain control of the spindle motor.

To verify the function of the spindle motor control board the manufacturer's drawing detailed in Figure 15 was analyzed, and a manual for the Sprint 400 Spindle Drive. A wiring diagram for basic control connections was found in the Sprint 400 Spindle Drive Manual (Figure 16 below) that demonstrated that the board could be controlled with an analog input supplied from a 10k Ω potentiometer for speed control and a digital input from a normally open momentary pushbutton as a run signal. Additionally, by moving a jumper on the Spindle Drive, the board could be powered with the more common 120 VAC instead of 240 VAC.

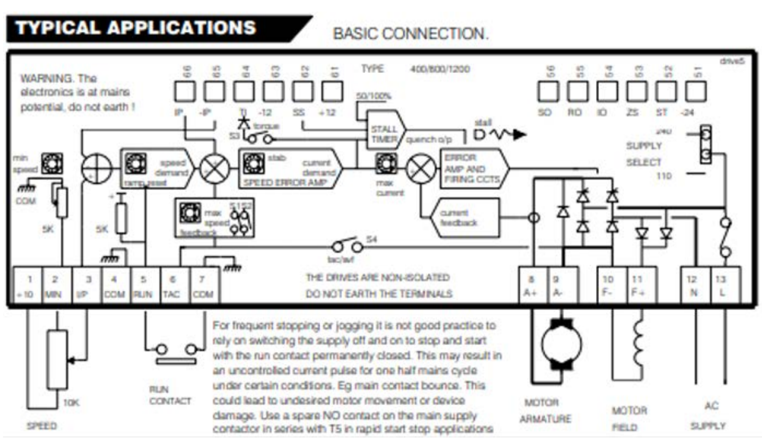


Figure 16: Sprint 400 Spindle Drive Basic Connections

The Sprint 400 Spindle Drive was completely removed from the control panel and control panel wiring and bench tested. A digital multimeter was connected to terminals 8 and 9 of the spindle drive to measure output DC voltage to the spindle motor, a 10k **Ω** potentiometer was connected to terminals 1, 2, and 3 for speed control, and a normally open momentary pushbutton was connected to terminals 5 and 7 as a run signal. Last, the jumper on the board was moved from the 240 VAC position to the 120 VAC position, and 120 VAC was connected to terminals 12 and 13 to supply power to the spindle drive. The 120 VAC power was energized, and voltage was measured across terminals 8 and 9 while the momentary pushbutton was pressed. Finding an initial value of 100 VDC at terminals 8 and 9, the potentiometer was adjusted while the momentary pushbutton was held closed. The output voltage on terminals 8 and 9 with no load was observed to vary between 0 VDC to 165 VDC. Based on the Spindle Drive test findings, the spindle drive was determined to function properly. As the spindle motor functionality had not yet been tested, the Spindle Drive was reinstalled in the Control Panel but not connected to the spindle motor. To better understand the spindle motor design, the CNC machine was dismantled until the nameplate of the spindle motor and wiring diagram are shown in figures 17 and 18 respectively.

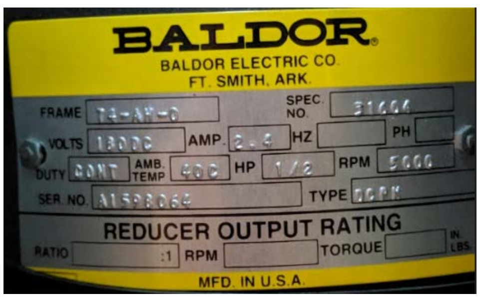


Figure 17 Spindle Motor Nameplate

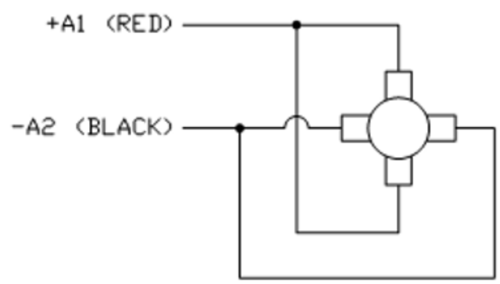


Figure 18 Spindle Motor Wiring Diagram

A continuity test was performed between the spindle motor positive and negative power leads, and 5.5 ohms of resistance was measured. The nameplate value from Figure 17 for rated voltage (180VDC) and the measured resistance (5.5Ω) were used to calculate the stall current I_{STALL}= 32.73A. The calculated stall current (32.73A) and the nameplate current (2.4A) were used with the measured resistance (5.5Ω) to calculate the Back EMF Voltage as follows:

$$V_{BEMF} = \left(I_{STALL} - I_{N/L}\right) \times R = \ (32.73\ A - 2.4\ A) \times 5.5\ \Omega = 166.82\ V$$

The calculated back emf value (166.82 V) was used with the nameplate rated voltage (180 VDC) and measured resistance to confirm the expected current on the motor:

$$I_{N/L} = \frac{V_{SUPPLY} - V_{BEMF}}{R} = \frac{180\ V - 166.82\ V}{5.5\ \Omega} = 2.4\ A$$

Finding no problem with the measured resistance value of the motor, the Spindle Motor was determined to be safe to connect to the Sprint 400 Spindle Drive.

a. <u>System Analysis: Spindle Motor and Spindle Motor Control</u>

To fully test the combined functionality of the spindle motor power and control subsystem, while the spindle motor was still on the bench the spindle motor positive lead was connected to terminal 8 and the negative lead was connected to terminal 9 of the Spindle Drive. Additionally, a digital multimeter was connected to these terminals to monitor voltage while the spindle motor was under test. A 10k Ω potentiometer was connected to terminals 1, 2, and 3 for speed control, and a normally open momentary pushbutton was connected to terminals 5 and 7 as a run signal. Last, the jumper on the board was verified to be in the 120VAC position, and 120 VAC was connected to terminals 12 and 13 to supply power to the spindle drive. The potentiometer was adjusted while the momentary pushbutton was held closed and the output voltage on terminals 8 and 9 while connected to the spindle motor was observed to vary between 0 VDC to 155VDC. The spindle motor speed was observed to vary with the applied voltage. Having successfully tested the speed control of the Spindle Motor in a bench test, the Spindle Motor was reinstalled in the CNC machine. The above test was repeated with the spindle motor coupled to the CNC spindle. The potentiometer was adjusted while the momentary pushbutton was held closed and the output voltage on terminals 8 and 9 while connected to the Spindle Motor was observed to vary between 0 VDC to 155VDC. The spindle rotational speed was observed to vary with the applied voltage. Based on the above test results, the spindle drive and spindle motor were determined to be in working condition under manual control, although as no previous means of automatic control.

VI. ANALYSIS OF COMPONENT TESTING

The overall troubleshooting and analysis of the NovaMill subsystems resulted in the following observations and repairs:

1. *Unregulated Power Supply*
    a. Screw terminals connecting the power supply subcomponents were found loose, potentially causing unstable voltages and currents from high-resistance connections. All screw terminals were tightened.
    b. Lugs used to connect the Rectifiers to the screw terminals were filled with solder instead of crimped, with broken solder joints found. The Rectifiers were replaced and new terminals were crimped to the new rectifier leads.
2. *BeagleBone Black and Probotix Breakout Board*
    a. The existing BeagleBone was found to be unresponsive and directories were empty, despite handling a SSH connection through PuTTY and WinSCP. This BeagleBone was set aside and a new BeagleBone was configured with MachineKit.
    b. The existing Probotix Breakout Board was damaged by a wiring error found on the power terminals of the Stepper Motor Drivers. A new Probotix Breakout Board was purchased to replace the damaged board.
    c. The distribution of MachineKit recommended by the guide [3] developed critical errors on the new BeagleBone. Due to the success in using an Arduino Uno to test the Stepper Motor Drivers, the recommendation of this report is to eliminate the BeagleBone and MachineKit entirely in favor of a more robust microcontroller like the Arduino Uno. The Arduino can be programmed as a G-Code interpreter and serially connected over USB to a computer running the free open source software, Universal G-Code Sender.
3. *GeckoDrive G201X Stepper Motor Drivers (X, Y, Z Axes)*
    a. The Stepper Motor Drivers passed a full function test despite wiring problems found on the power terminals that lead to the destruction of the Probotix Breakout

Board. Current limiters on the Stepper Drivers were not set, which may have contributed to the observed problems with the X and Y-Axis Stepper Motors.

4. *Stepper Motors (X, Y, Z Axes)*
    a. The X and Y-Axis Stepper Motors should be replaced due to their loud operation, grinding sounds, and high temperature rise over the 105 second function test. Additionally, the existing Stepper Motors are no longer manufactured or supported by the manufacturer. Recommended replacement X and Y Stepper Motors are detailed in Appendix C at the end of this report. Furthermore, although the Z-Axis Stepper Motor passed the function test it is also no longer manufactured or supported by the manufacturer. If desired, an additional Stepper Motor of the same make and model as recommended for the X and Y-Axes can be used to replace the Z-Axis Stepper, although an adapter plate.
5. *Spindle Motor and Spindle Motor Control*
    a. The Spindle Motor and Sprint 400 Spindle Driver were acceptably tested under manual control. As the previous students had removed the preexisting control wiring and this report determined that the BeagleBone Black and Probotix Breakout Board may not be the best options for system control, a decision was made to leave the Spindle Motor and Spindle Driver as tested.

VII. CONCLUSION

This paper describes in detail the conative steps and methods an engineer must take to troubleshoot a nonfunctioning machine in general and particularly a Denford NovaMill 3-Axis CNC Milling Machine and the specific troubleshooting methods required to diagnose the various problems preventing full functionality, by isolating each problem through hypotheses developed at the system, subsystem, device, and component levels. This means of troubleshooting allows a problem to be rapidly reduced to its root causes. To assist in troubleshooting manufacturer's data was examined to provide a baseline from which the components could be evaluated, and compared to test results for each component as needed. As defective components were identified, some were found to be obsolete and difficult to source. For these components, this paper includes descriptions, model numbers, and sourcing for new components with similar or better functionality than the legacy component. In cases where replacement component's physical size differed from the legacy component, fitting plates were modelled in SolidWorks and drawings were generated to provide ease of integration of the new component in the legacy system.

The cognitive troubleshooting methods outlined in this report are key skills for any engineer to assist in the repair, maintenance, and retrofit of existing equipment. These skills are one of the primary reasons a company will employ an engineer to assist in the repair and upgrade of process equipment, as unexpected equipment failure can frequently cost a company thousands of dollars an hour in lost productivity and wasted overhead. Additionally, further productivity can be lost and additional repair costs can be incurred by misdiagnosing the root cause of a fault and spending time and material replacing components that do not actually correct the problem.

While the Denford NovaMill is a relatively simple machine compared to a full-scale process control system, troubleshooting the NovaMill and its subsystems reinforces the basic cognitive processes required to develop and test a hypothesis for the fault, isolate a fault at any scale, and verify functionality of complex systems.

VIII. REFERENCES

[1] Develop & upgrade a CNC mill conversion. (2018, April 18). *Hesham Magbool and Derek Benakovitch.*

[2] Upgrading a CNC milling machine Part 1. (2015, May 18). Retrieved October 30, 2017, from https://www.rs-online.com/designspark/upgrading-a-cnc-milling-machine-part-1

[3] Upgrading a CNC milling machine Part 2. (2015, June 05). Retrieved October 30, 2017, from https://www.rs-online.com/designspark/upgrading-a-cnc-milling-machine-part-2

[4] Upgrading a CNC milling machine part 3. (2015, June 23). Retrieved October 30, 2017, from https://www.rs-online.com/designspark/upgrading-a-cnc-milling-machine-part-3

[5] https://eric.ed.gov/?id=ED389948

[6] https://www.probotix.com/wiki/index.php/PBX-BB_rev5.2

[7] Meet the Beagles: Open Source Computing. (n.d.). Retrieved November 01, 2017, from http://beagleboard.org/

[8] "Machinekoder/Asciidoc-Sandbox." *GitHub*, 3 Mar. 2015,github.com/machinekoder/asciidoc-sandbox/wiki/Creating-a-Machinekit-Debian-Image#step-8-deactivate-all-unneeded-capes.

[9] CASTLE, ALEX. "Everything You Need to Know about The Beaglebone Black." *Tested*, 4 Dec. 2013, www.tested.com/art/makers/459278-everything-you-need-know-about-beaglebone-black/.

[10] "Beagleboard:BeagleBoneBlack Debian." *Beagleboard:BeagleBoneBlack Debian - ELinux.org*, elinux.org/Beagleboard:BeagleBoneBlack_Debian.