

B a c h e l o r T h e s i s

Development of a Test Rig for the Automated
Measurement of the Static Force Characteristic
of Pneumatic Muscle Actuators

Michael Drummond

Matr.-Nr.: 351750

Studiengang: BSc Elektrotechnik

Contents

1	Abstract	1
2	Introduction	1
3	Theory	1
3.1	Pneumatic Muscle Actuators	1
3.2	Static Force Maps and Control Theory	1
3.3	The Test Rig	1
4	Hardware Development	2
4.1	Outline	2
4.2	Power Supply	4
4.2.1	DC/DC Regulator	5
4.2.2	Analogue and Digital Power Rails and Ground Planes	8
4.3	Rotary Encoder and Buttons	10
4.3.1	The Rotary Encoder	10
4.3.2	The Buttons	11
4.4	LEDs	12
4.5	Liquid Crystal Display	12
4.5.1	I ² C	12
4.5.2	Level Shifter	12
4.6	CAN bus	14
4.6.1	CAN Transceiver	14
4.6.2	CAN Bus Circuit Protection	15
4.7	Analogue Inputs	15
4.7.1	Voltage Reference	17
4.7.2	Pressure Sensor	17
4.7.3	Force Sensor	18
5	Software Development	21
5.1	Outline	21

5.2	Device Tree Overlays	22
5.3	User Interface	24
5.3.1	I ² C and LCD	24
5.3.2	Rotary Encoder and Button	24
5.3.3	Menu	25
5.4	Control System	26
5.4.1	Analogue to Digital Conversion	26
5.4.2	CAN bus	26
5.4.3	PID Controller with PRU Timer	26
5.5	Data Storage	26
6	Data Acquisition	26
7	Discussion	26
8	Conclusion	26

1 Abstract

2 Introduction

3 Theory

3.1 Pneumatic Muscle Actuators

Pneumatic muscle actuators go by many names, such as braided pneumatic muscles (BPMs), McKibben pneumatic artificial muscles (PAMs), fluidic muscles or biomimetic actuators. First developed in the 1950s for use in an orthopaedic appliance by the physician Joseph McKibben, PMAs have since been put to use in a range of applications from industry, to medicine and entertainment. PMAs typically consist of an elastomeric tube or bladder, a braided mesh of non-elastic fibres that are either worked into the bladder or wrap around it like a sleeve, and two end fittings that seal the tube but for an opening which attaches to a source of compressed air. The non-elastic mesh is attached to the fittings so, when the bladder is inflated via the opening at one end, the bladder can only expand radially and a tensile force is generated in the axial direction.

3.2 Static Force Maps and Control Theory

3.3 The Test Rig

The aim of the thesis was to create a self-contained device with which the static force characteristic of pneumatic muscle actuators (henceforth PMAs) of different sizes could be tested automatically. Static force maps have previously been measured manually ???. The set-up can be summarised as follows: a PMA is held at both ends at a given length with the aid of a rigid metal frame. A force sensor bridges the connection from the PMA to the frame at one end. As the internal pressure of the PMA is increased a tensional force is exerted on, and

can be measured by, the force sensor. The force exerted is measured at multiple pressure points for a given length that the PMA is set at. This is repeated for multiple lengths, resulting in a "map" of the force response of a PMA at different pressures and lengths (see ??).

The manual set-up involved opening a valve connecting the PMA to a source of compressed air by hand, adjusting to a given internal pressure of the PMA and reading the exerted force. The operation of the valve was not only time consuming but a laborious, repetitive action. The force sensor itself was a strain gauge which translates changes in applied force into a voltage. This in turn needed to be amplified before being read with a multimeter, the result was then converted to ascertain the original force exerted.

Following a run through of the pressure points at a given length the PMA then needed to be manually unbolted from the rigid frame that was holding it. The length of the PMA would then have to be changed by adjusting its internal pressure, before being bolted to the frame once more. See ?? for a photo of this set-up.

For this thesis a device was to be developed that reduced the time with which an operator would have to continually contend with adjusting the valve to achieve pressures that are predefined. In addition, the automated measurement of the force applied by the PMA was to be incorporated into the design. Input from an operator would, therefore, only be required when the length of the PMA is changed.

4 Hardware Development

4.1 Outline

Complex systems are best divided into their component parts. Figure 1 gives an indication of the relevant connections between the elements of the device, whether it is an input or output, and by which means the connection is made.

The BeagleBone Black, a single board computer (SBC) running Linux, was chosen as the core of the device. For this application a microcontroller board such as the Arduino Due would potentially have been sufficient. However, it was felt the additional power and possible future expansion options (onboard data processing; web server user interface) available on a Linux powered SBC outweighed the negatives of dealing with a more complex system. Other SBCs considered were the Raspberry Pi, but the sheer number of connection options available to the BeagleBone Black (critical to this project) made it the better choice.

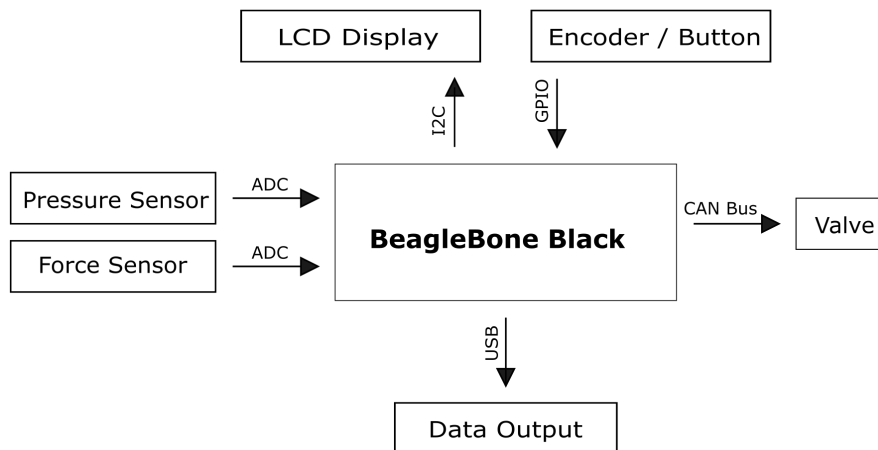


Figure 1: A simplified breakdown of device components, indicating direction of data transfer (I/O) and by which means

The BeagleBone Black is responsible for the control loop feedback mechanism that regulates the PMA internal pressure by means of the pressure sensor input and valve position output. Communication between the BeagleBone Black and the valve takes place over a CAN bus. The pressure sensor signal voltage has to undergo a small amount of signal conditioning before being converted by means of an analogue to digital converter on the Beaglebone Black. The signal from the force sensor is first passed through a measurement amplifier before receiving similar signal conditioning and conversion via the ADC. Once the data is gathered, it can be transferred to a USB stick in an appropriate format for further processing elsewhere. The device and its function is controlled entirely with the use of a rotary encoder with a push button and an LCD Display. A connection to a computer or other device is therefore not required.

The BeagleBone black is designed to mate with expansion boards (known as capes), with which its functionality can be extended. For this thesis a cape was developed to accommodate the additional circuitry required to make the device self-contained. This additional circuitry can broadly be divided into the following sub-circuits: power regulation, analogue signal conditioning, level-shifter for the LCD, CAN bus, buttons and LEDs. Surface mount components were used almost exclusively, to keep the overall size of the cape small. The small size allowed for the device to be fitted inside an off-the-shelf housing, reducing total design time.

4.2 Power Supply

Table 1 outlines the power requirements of the connected devices. Important to note are the supply voltages of the measurement amp and the valve. As a result the system was designed from the outset to take a 24V input, with this being passed on to the measurement amp and the valve, while a DC/DC converter takes this down to 5V to run the BeagleBone Black and the other peripheral devices.

Device	Power requirements	
	Voltage [V]	Current [mA]
BeagleBone Black	5	max. 460
LCD display	5	
Pressure sensor	5	<2
Measurement amp	24	55...75
Valve	17...30	max. 1100

Table 1: Supply voltage and current draw of the connected devices

While the BeagleBone Black has a supply voltage of 5V, this is itself regulated further down internally to 3.3V. As a result the digital I/O pins of the BeagleBone Black all only accept or put out 3.3V. This is particularly relevant to the LCD display, which expects 5V signals over an I²C bus (see 4.5.1). The BeagleBone Black conveniently provides a pin at which to tap the regulated 3.3V and provide a third power rail to a number of sub-circuits describe later.

Furthermore, the analogue to digital converter of the BeagleBone Black only accepts voltages between 0V and 1.8V. Here, again, an analogue voltage reference pin is provided and its use with the ADC is described in 4.7.1.

4.2.1 DC/DC Regulator

There are essentially two types of voltage regulation: linear regulation and switching regulation. In its most basic form a linear regulator uses a series resistor to dissipate excess power as heat. To achieve a stable output voltage independent of the load a variable resistor and optionally a feedback loop is used. Such regulators are practical in their simplicity, but inefficient as excess power is dissipated as heat. If large voltage drops are required, the heat loss can be considerable.

A switching regulator is an active device that achieves an average output by switching the voltage source on and off. A similar feedback mechanism to that found in linear regulators is used to control the output. While more expensive, switching regulators are considerably more efficient than linear regulators, as the only power losses occur during switching and these are small in comparison. Other advantages of switching regulators is that with some clever switching they can generate output voltages higher, or of opposite polarity than the input voltage.

For this device a voltage drop from 24V to 5V was required. Table 1 lists the current draw of the main connected devices. The devices that draw the most current are the valve and the BeagleBone Black itself. The BeagleBone Black can reach its maximum current draw of 460mA during the boot process; while idle, current draw is in the region of 300mA [1]. The valve draws 100mA in mid-position but can reach 1100mA at full stroke. As the valve is not in use during the BeagleBone Black boot process the current draw of the system will, during normal operation, not exceed 1500mA, and this only for short periods of time.

Given the above considerations a linear regulator would have to dissipate almost 30W of power at peak load. This isn't practical for small form-factor devices such as this and would generate too much heat. The better alternative is a switching regulator. Several manufacturers produce switching regulator integrated circuits.

For this application the LM2596-5 Simple Switcher[®] from Texas Instruments was selected. Specifically, this step-down voltage regulator provides a fixed 5V output and is capable of driving a 3A load with excellent line and load regulation. Its comparatively high switching frequency of 150kHz allows for smaller sized filter components, a useful attribute when space is at a premium. [2]

The LM2596 data sheet comes with extensive instructions on selecting the appropriate values for the peripheral components required. A maximum load current of 3A was assumed, and where possible values were selected to minimise output voltage ripple (for example in the selection of the output capacitor). The circuit was subsequently built and tested on perfboard. The final design of the power supply can be seen in Figure 2.

The system receives 24V via a barrel connector from a power supply.

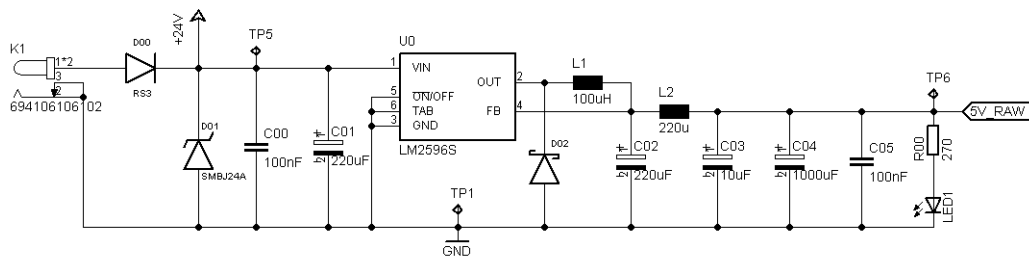


Figure 2: Circuit diagram of the DC/DC regulator and peripheral components

The design incorporates several additional elements:

Over voltage protection. This takes the form of a Transient Voltage Suppressor (TVS) diode anti-parallel with the power source (D01). Well laid out TVS diodes are designed to have negligible impact on the normal operation of a circuit. The breakdown voltage of the TVS diode is usually in the region of 10% greater than the reverse standoff voltage which approximates the circuit operating voltage. When a voltage transient occurs that exceeds the breakdown voltage of the TVS diode, it will limit the voltage spike to the TVS diode's clamping voltage and conduct the transient current to ground [3].

For this device the SMBJ24A 600W TVS diode from Fairchild Semiconductor

was chosen due to its combined small package size, peak pulse power capability and fast response time. [5]

Reverse polarity protection. There are several options for implementing reverse polarity protection. In low voltage systems where a voltage drop can't be afforded, the use of a P-channel MOSFET is common, as its effect on the voltage supply is minimal (Figure 3).

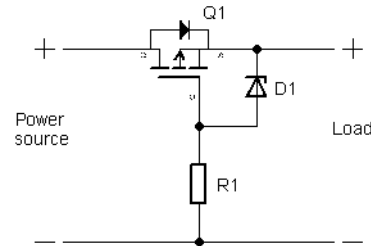


Figure 3: A common protection circuit against reverse polarity

For the system being designed a simpler option is acceptable. A rectifier diode is placed in series with the power source (D00). During normal operation the diode is forward biased and current flows to the circuit. Should the polarity of the voltage be reversed, in other words the diode is now reverse biased, the diode will block and the circuit will see no current. The disadvantage of this simpler solution is that the diode exhibits a forward voltage drop; as a result the circuit will no longer receive 24V. The rectifier diode chosen for the circuit (see below) has a forward voltage drop of 1.3V, so the circuit can expect to see 22.7V instead of 24V. While the nominal operating voltage of the valve is at 24V, the operating voltage range lies between 18V and 30V. Similarly the Measurement amplifier has a nominal supply voltage range of 12V to 24V. In both cases the voltage drop caused by the rectifier is of no consequence.

The RS3G Fast Switching Rectifier from Vishay was chosen for its small package size, high forward current (3A) and fast switching time.

Additional filtering. In addition to the 220 μ F input capacitor (C01) stipulated by the DC/DC regulator data sheet, a 100nF capacitor (C00) was added in parallel to remove high frequency noise. Similarly output ripple and noise were further stabilised with an additional 220 μ H inductor (L2), 10 μ F, 1000 μ F, and 100nF capacitors (C03, C04 and C05 respectively).

An indicator LED. A simple visual indicator placed at the point at which the 5V produced by the DC/DC regulator enters the BeagleBone Black. The BeagleBone Black should enter its boot procedure once 5V are connected, so if the LED is

on and the boot process doesn't occur, a problem can be inferred.

4.2.2 Analogue and Digital Power Rails and Ground Planes

Digital circuitry is noisy, with rapid switching resulting in current transients and electromagnetic interference in the power supply. The logic levels of digital circuits are themselves virtually immune to these effects, however analogue circuitry powered by the same supply can be vulnerable to induced noise. This is particularly important in signal conditioning circuitry where the fidelity of the signal to be measured is important. Similarly, measurements made with respect to ground need a solid ground. Ground planes underneath digital circuits will suffer from switching noise generated by the digital circuit.

To minimise the effect of the digital circuitry on the analogue circuitry a separate analogue ground plane and analogue 5V power rail were introduced. Figure 4 shows how the regulated 5V output from the BeagleBone Black was uncoupled from the 5V analogue power rail. This was achieved with an EMI suppression ferrite bead (L3) and two capacitors (10 μ F and 100nF). Ferrite beads are passive devices that filter high frequency noise by increasing reactance in a specific frequency range and dissipating energy in the form of heat [4]. The chosen ferrite bead has a maximum impedance at 150MHz. The ferrite bead is complimented by two capacitors to ground, forming a low pass filter which further suppresses higher frequencies on the supply rail. Space was made for an additional decoupling capacitor (C09) if further filtering is deemed necessary once the board is taken into operation.

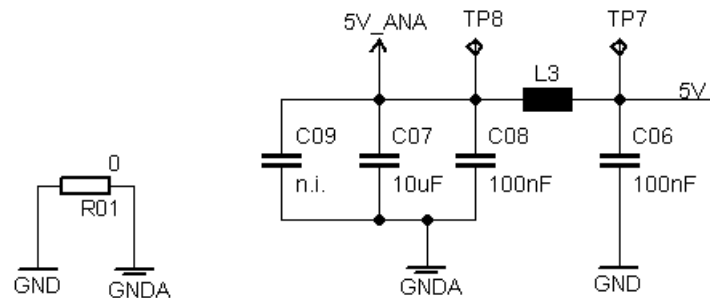


Figure 4: Circuit diagram of the interfaces between the analogue and ground planes (left), and the analogue and digital 5V power rails (right)

The circuit of the connection between the digital and analogue ground (Figure 4) is simply a 0Ω resistor. In itself, this means nothing, as the resistor has no effect other than to connect the two grounds. However, in practical terms during PCB layout, this enforces the connection of the two ground planes at a single point. Known as star point grounding, the principle of connecting grounds at a single point ensures that all voltages are measured with respect to a particular point, and not just at an undefined "ground" [7]. In many cases a convenient point at which to connect the ground planes is at the power supply. Figure 5 shows the bottom of the developed PCB, with digital ground occupying most of the board. Analogue ground is in the bottom left quarter of the board. The two grounds are completely separate, except for the 0Ω resistor which lies at the top of the board, analogue ground trace that runs up the left hand side of the board. The resistor sits next to the power source and defines the common, defined ground between the two sides of the circuit.

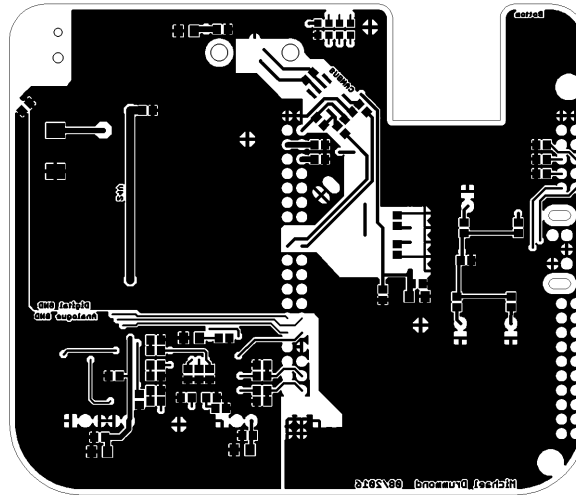


Figure 5: PCB layout showing the ground planes on the bottom of the board

4.3 Rotary Encoder and Buttons

4.3.1 The Rotary Encoder

Control and navigation through the device options (see 5.3.3) is enabled with the use of a rotary encoder. An encoder or jog wheel was considered a more intuitive interface for setting the valve than a series of buttons, and had the added benefit of giving the device a cleaner appearance.

The Texas Instruments Sitara processor on the BeagleBone Black includes an Enhanced Quadrature Encoder Pulse (eQEP) Module as part of its Pulse-Width Modulation Subsystem (PWMSS). The eQEP simplifies the implementation of an encoder by decoding its input and making it available to the programmer as a single integer value that is increment or decrement accordingly. In terms of hardware, the two encoder signal lines simply need to be connected to the correct two pins on the BeagleBone Black, and ground to ground.

The LCD display on the top face of the device and the device's overall compact size meant placing the encoder next to the display was no longer possible. To make its use ergonomic the rotary encoder was moved to the right hand side of

the device, and is thus angled at 90° to the PCB.

4.3.2 The Buttons

The rotary encoder has a built-in push button switch. This momentary switch allows for the selection of options during the navigation of the GUI. As the device was to include the option of expanding its initial functionality, space for a further three switches was made by including two pin Molex connectors. The switch and the three headers are connected between ground and an individual GPIO pin of the BeagleBone Black, and each include debounce circuitry.

This takes the form of a low pass filter () followed by a Schmitt trigger...

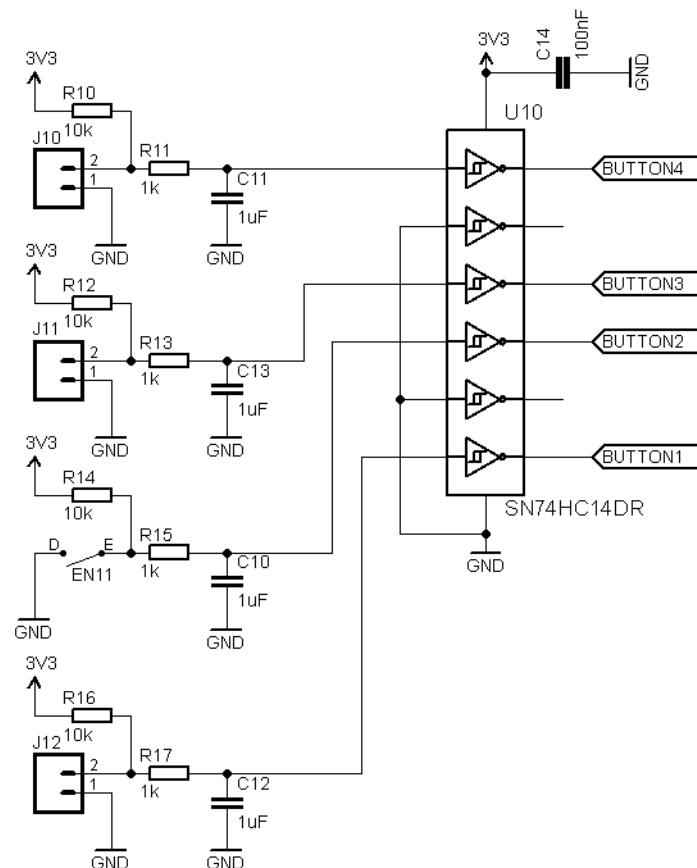


Figure 6: Debounce circuitry of the buttons. Note the encoder connected to button 2.

4.4 LEDs

In addition to the power supply LED (see 4.2.1) a further three LEDs were added to the device to provide someone programming the device with a simple debugging tool or a user with a visual cue that the system is working.

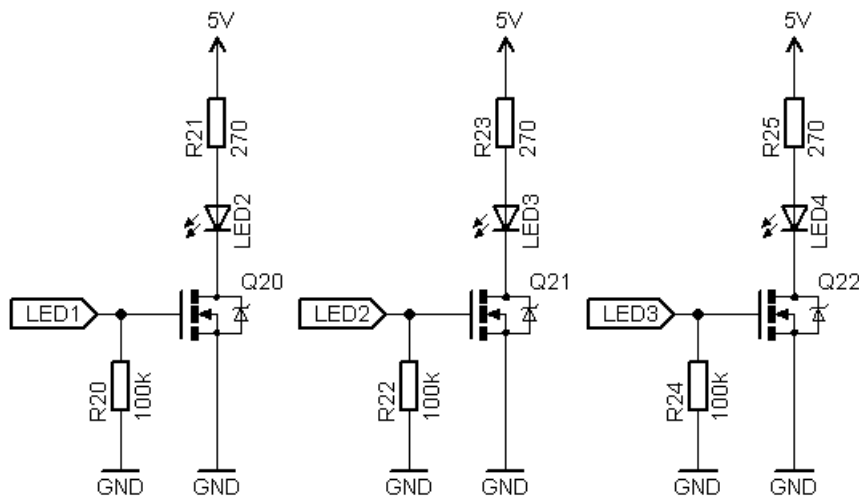


Figure 7: The connected LEDs

4.5 Liquid Crystal Display

The liquid crystal display (LCD) of the device is a 4 line, 20 character, blue backlit display. The LCD's Hitachi HD44780 controller is connected on-board with an I²C interface, cutting down the number of pins required to control the display significantly (from 12 pins in four-bit HD44780 operation, to four pins for I²C). Supply and logic level voltage of the display are 5V.

4.5.1 I²C

4.5.2 Level Shifter

Logic level voltage of the BeagleBone Black pins is 3.3V, so a bi-directional level shifter circuit was placed between the I²C data (SDA) and clock (SCL) pins and

those of the display (Figure 8). The level shifter ensures that the 3.3V logic signals from the BeagleBone Black are read as 5V signals at the display.

The circuit involves an n-channel MOSFET (Q30 and Q31) placed on each of the signal lines (SDA and SCL), with source connected to the 3.3V bus line and drain to the 5V bus line. These are flanked by pull-up resistors to the two power rails (R30 to R33). The gates of both MOSFETS are connected to the lower voltage supply rail (3.3V).

In this configuration, if neither side is pulled down, the pull-up resistors result in 3.3V at one end of the bus line and 5V at the other. Should the 3.3V end of the bus line be pulled down, V_{GS} will rise, the MOSFET conduct and the 5V end of the bus line will also be pulled down. Equally, should the 5V end of the bus be pulled down, the body diode of the MOSFET will conduct, again drawing V_{GS} down until the MOSFET conducts and the bus line is pulled down to the same level at both ends. [6]

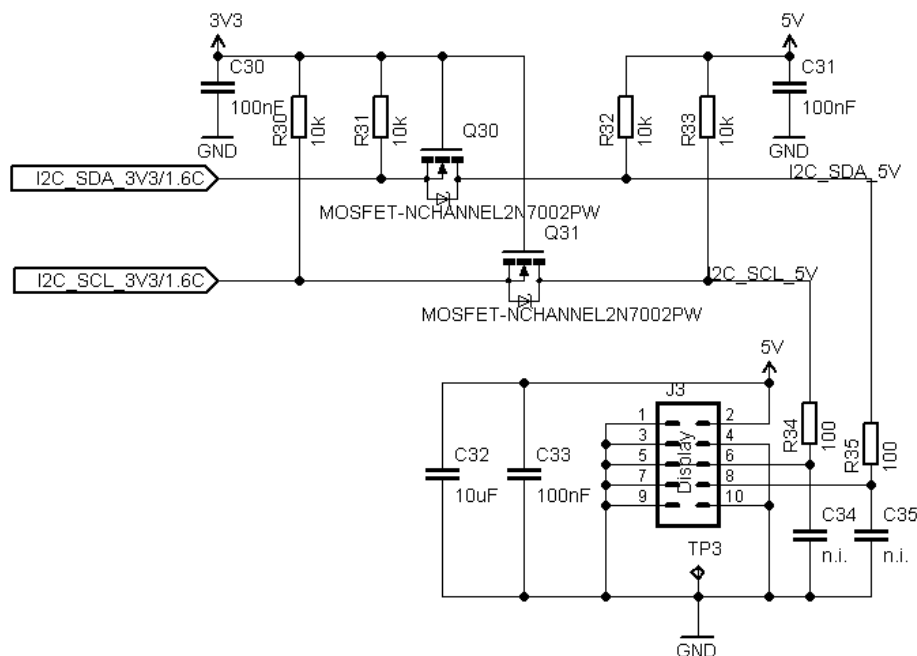


Figure 8: The logic level shifting circuit between the I²C pins of the BeagleBone Black and those of the display

4.6 CAN bus

The AM335x processor on the BeagleBone Black incorporates a module for accessing a Controller Area Network (CAN) bus. The DCAN module provides support for CAN protocol version 2.0 at bit rates up to 1MBit/s [8]. The AM335x includes two instantiations of the DCAN controller (DCAN0 and DCAN1), both are connected to pins on the P9 header of the Beaglebone Black [1]. Due to the high frequency switching on the RXD and TXD lines of the CAN controller these should not be placed over a ground plane. The RXD and TXD of DCAN1 are connected to pins 24 and 26, but their use would interfere with the ground plane that extends from the left to the right side of the board. The ground plane is itself constrained by the analogue circuitry at the bottom left of the board. For this reason, DCAN0 with its RXD and TXD lines connected to pins 19 and 20 was chosen to provide the CAN controller.

4.6.1 CAN Transceiver

Additional hardware, in the form of a transceiver, is required for the connection to the physical CAN bus [9]. As the RXD/TXD pins of the BeagleBone Black operate at 3.3V an appropriate 3.3V CAN bus transceiver was needed. For this the Texas Instruments SN65HVD232 was chosen (U40, Figure 9). Following the data sheet layout guidelines, serial resistors (R40 and R41) were placed to limit current on the digital lines [10]. An optional pull up resistor (R42) was used to drive the recessive input state of the transceiver. High speed CAN (see ISO 11898-2 [11]) requires termination resistors at both ends of the transmission lines. The resistors match the nominal impedance of the cable, which the CAN standard specifies as 120Ω . R43 is such a termination resistor, placed as close to the CAN High and CAN Low pins of the transceiver as possible. In addition, a bypass capacitor was placed as close as possible to the supply pins of the transceiver (C40), and bulk and bypass capacitors were placed on the 24V supply line of the valve (C41 and C42).

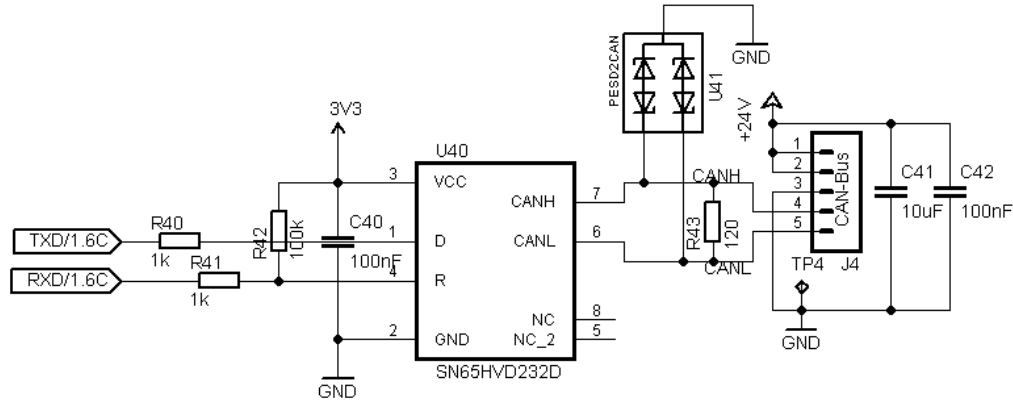


Figure 9: CAN transceiver, ESD protection diodes, and circuitry connecting the on-board CAN controller with the CAN bus

4.6.2 CAN Bus Circuit Protection

The on-chip electrostatic discharge (ESD) protection of the CAN transceiver is good enough for many applications, but insufficient for electrical fast transients (EFT) and surge transients occurring in industrial environments [10]. While this device is unlikely to leave the laboratory, good practice in designing a robust and reliable bus node was followed and additional circuit protection was added. This took the form of a CAN bus ESD protection diode from NXP Semiconductors (PESD2CAN, U41 in Figure 9). This was placed close to where the bus connects with the board to prevent the transients from propagating on the PCB.

4.7 Analogue Inputs

The analogue inputs are core elements of the device. Input from the pressure sensor is used in the control function to find the pressure set point selected by the user. The accurate setting of the muscle is conditional on the accuracy of the feedback response measured by the pressure sensor. Once the set point has been reached, the device takes a force measurement from the force sensor. This data underpins the measurement of the static force characteristic of the PMAs, and so its accuracy is likewise essential.

The AM335x processor of the BeagleBone Black includes a subsystem with an 8-channel general-purpose analogue-to-digital converter (ADC), with optional support for interleaving touchscreen conversions for a resistive panel (abbreviated to TSC_ADC_SS) [8].

The subsystem comprises of a single 12 bit Successive Approximation Register (SAR) ADC that is multiplexed to the 8 channels. The ADC will measure voltages between 0 and 1.8V. The 1.8V supply that drives the ADC is provided as a voltage reference on a pin of the BeagleBone Black, as is an analogue ground pin. Both the pressure and force input signals require conditioning to bring them within the voltage span of the ADC. In both cases this was achieved with a simple voltage divider (Figures 11 and 12).

The ADC has a maximum sampling rate of 200kHz. This is significantly faster than the sub kilohertz sampling rate anticipated for the device, even if measurements are taken from two channels. The sampling frequency determines the input impedance of this ADC, so with both pressure and force measurements each taken at 100Hz the input impedance equates to [12]

$$\frac{1}{65.97 \times 10^{-12} \cdot 200\text{Hz}} \approx 75.8\text{M}\Omega$$

While this input impedance may be sufficient given the device's current application, a change (for example) in measurement frequency could easily bring the introduced error caused by input overloading to an unacceptable level. For this reason a unity-gain buffer was placed between the voltage dividers and the ADC input (Figures 11 and 12). This impedance matching was implemented with the MCP6004 quad op-amp IC from Microchip Technology inc.

A further consideration was the small capacitor that the sample and hold element of the ADC uses. This can cause problems for the buffer's op-amp at high sampling frequencies. If the dynamics of the op-amp are too slow, the capacitive load on the op-amp can result in settling times that can skew the result. To deal with this, a simple RC filter can be placed between the buffer output and ADC input. Space for such filters were made in the design, but given the low sampling frequencies planned for this device, they were not instantiated.

4.7.1 Voltage Reference

The BeagleBone Black 1.8V ADC voltage reference pin was used to drive four op-amps required for impedance matching in the signal conditioning circuitry of the analogue inputs (described above). To prevent the circuit drawing a current that could damage the BeagleBone Black, the output of the voltage reference pin was itself buffered, as in Figure 10. The component used here was a MCP6001 single op-amp, supplied by the 5V analogue rail. In addition to a 100nF bypass capacitor, a 10 μ F bulk capacitor and 5 Ω resistor were placed at the end of the 5V supply to prevent any drops in the supply and maintain an accurate reference.

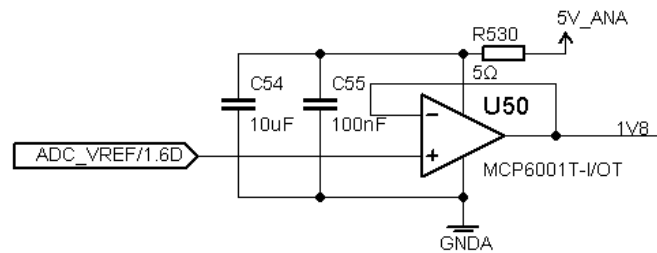


Figure 10: VREF

4.7.2 Pressure Sensor

The 1.8V voltage reference supplies the quad op-amp IC, of which one op-amp is used in the signal conditioning of the pressure sensor input (Figure 11). The pressure sensor is supplied by the analogue 5V rail, filtered and supported by bypass and bulk capacitors. The sensor output is a value between 0 and 5V which is paired down in a voltage divider to bring it to between 0 and 1.8V. Resistors with 0.1% tolerance were selected to improve accuracy. The voltage divider is followed by a unity gain buffer to match the divider's impedance to that of the analogue input pin of the BeagleBone Black (described above). The total resistance of the divider was aimed in the region of 150k Ω . The relatively high resistance prevents the sensor measurement from being negatively impacted by a high current draw. Moreover, the unity gain buffer negates the problems the high resistance of the divider might cause with the ADC input.

The pressure sensor was connected with AIN0 of the BeagleBone Black.

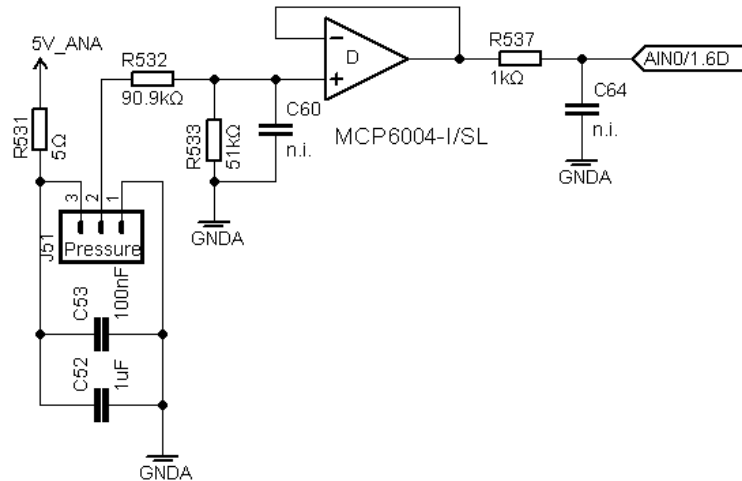


Figure 11: Pressure

4.7.3 Force Sensor

The signal conditioning circuitry connecting the force sensor to the BeagleBone Black is built on a similar set-up to that of the pressure sensor. That is to say, a voltage divider pairs the signal down to between 0 and 1.8V and a unity gain buffer matches the impedance to that of the input of the ADC. However, several further additions enhance the accuracy of the measurements (Figure 12).

The PMAs to be tested by the rig are available in different diameters, from 5mm to 40mm. The larger the diameter of the PMA, the greater the force that it is able to exert. So, while the force sensor and external measurement amplifier are set up to read a force up to ??N and convert it to a voltage between 0 and 10V, the maximum voltage output will only ever be reached by the very largest muscles. It is, therefore, only the force measurements of these PMAs that will use the full span of the 12 bit ADC on the BeagleBone Black, and that of the smaller PMAs will have a corresponding loss in resolution. To compensate for this, three voltage dividers with differing ratios were implemented. A CMOS analogue switch (U52, a Maxim MAX4662) placed before the dividers allows the user of the device to choose between them. During use only one switch will be

"on" at any given time, and the signal will pass through this divider to be read by one of the ADC channels.

The voltage dividers themselves are composed of 0.1% tolerance metal film resistors. To improve accuracy and their flexibility in use, each divider is made up of two pairs of resistors in series. In so doing, sourcing appropriate resistor values for the divider ratio was made easier. The voltage divider ratios, and the switches they are connected to, are listed in table 2. The user will select the input path based on the maximum voltages expected. For any given signal input path the dividers will convert the signal from between 0V and the respective maximum input voltage to between 0 and 1.8V.

Analogue switch	Voltage divider	
	Max. input [V]	Ratio
Switch 1	3.3	1.833
Switch 2	10	5.55
Switch 3	6	3.34

Table 2: Analogue force sensor voltage dividers. Output voltage of the dividers is in all cases between 0 and 1.8V given an input between 0V and the maximum input of the signal path chosen by the switch.

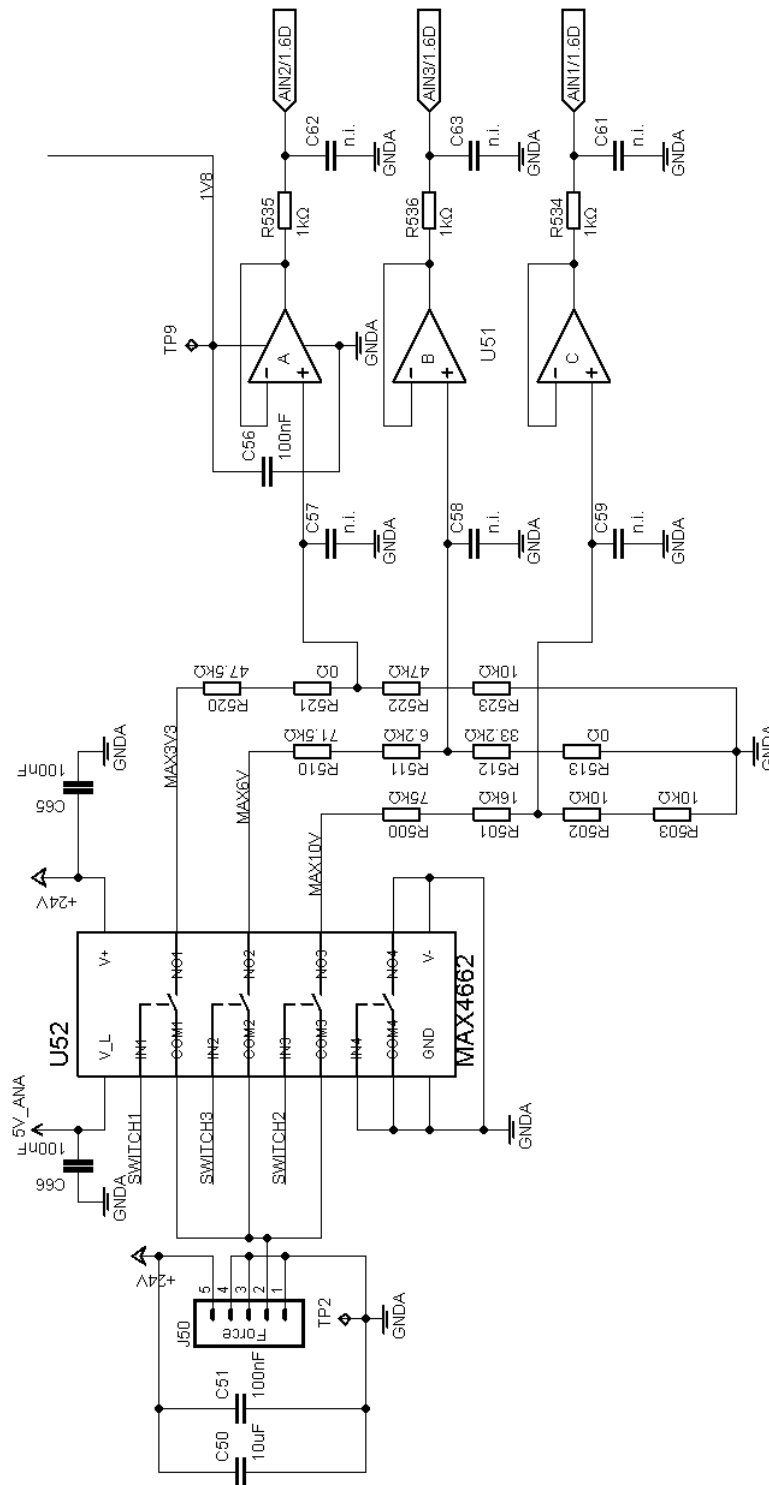


Figure 12: Force

5 Software Development

5.1 Outline

The BeagleBone Black can run a fully fledged Linux operating system, courtesy of the Texas Instruments AM3358 ARM Cortex-A8 processor. Debian 8.6 (Jessie) was installed on the device, and a remote debugging environment was set up according to the instructions in chapter 7 of [13].

The development of the software can conceptually be divided into five areas. The first three form the core of the device's function:

- The analogue to digital conversion of the pressure and force inputs
- The CAN bus output to the compressed air valve
- The PID controller links the pressure sensor with the valve actuator
- Storage of the recorded data on removable media
- The user interface. Fundamental to the user's interaction with the device. Primarily this is the menu, but also includes the initialisation and communication between the BeagleBone Black and the display and encoder.

A standard Linux kernel, such as the one that Debian is built on, is a preemptive kernel. Preemptive scheduling has many advantages, but it cannot execute tasks in real time. This is problematic in implementations of discrete-time PID controllers that are called upon to act at specific time intervals.

The BeagleBone Black's AM3358 includes a Programmable Real-Time Unit Subsystem and Industrial Communication Subsystem (PRU-ICSS), which provides a convenient way to implement real-time functionality where it is needed. The PRU-ICSS can be thought of as two 32 bit on-chip microcontrollers running at 200MHz (known as PRUs). They have full access to the internal memory of the processor and its peripherals, and single-cycle IO access to a number of pins (but not all). So, for example, an initial mock-up of the test rig had a PRU reading pressure values from the ADC and outputting a Pulse Width Modulation (PWM) signal to a valve. The PRU has single-cycle access to the ADC values,

and the software defined PWM output used a GPIO pin.

The PID controller would have been implemented with the help of an on board single-cycle 32 bit multiplier, however, external factors led to the PWM valve being replaced with a CAN-bus valve. The PRUs don't have direct access to the DCAN module that performs CAN protocol communication on the AM3358, so a hard real-time set-up of the control system fully implemented using the Programmable Real-Time units was no longer possible.

To keep the control system simple, yet keep some of the benefits of the PRU-ICSS, the control system was programmed in Linux space, while an accurate timer was implemented using one of the PRUs. The PRU generates an interrupt at precise time intervals, that alert the controller running in a high priority thread in Linux space to calculate a new output value.

5.2 Device Tree Overlays

The device tree is a data structure for describing hardware, that obviates the need to hard code details of a device into an operating system. Aspects of the hardware can be described in the device tree, which is passed to the operating system at boot time. Device Trees have become mandatory for all new ARM SoCs, and Debian Jessie (used on this device) is built on the Linux kernel version 3.16, which implements the Flattened Device Tree (FTD) model [14]. To enable runtime configuration of inputs and outputs on the BeagleBone Black, a system of Device Tree Overlays (DTOs) and a cape manager exist [15].

The following device tree .dts files were programmed, compiled as device tree binary overlays (.dtbo), and place in the /lib/firmware folder. At boot, the system loads the .dtbo files to extend the device tree.

- **ADC.dts** Enables the use of the ADC channels 0 to 6. Pins 32 to 40 of the P9 header are exclusively used by the ADC. These pins include the voltage reference and analogue ground (described in 4.7) along with the 7 aforementioned input channels.
- **DCAN0.dts** Enables one of the DCAN controller modules that performs

CAN protocol communication on the AM3358. DCAN0 was enabled for PCB design layout considerations (see 4.6). RXD and TXD of DCAN0 are multiplexed to pins 19 and 20 of the P9 header of the BeagleBone Black. As the I²C2 bus is set to default on these pins, it was first necessary to modify the device tree binary that is used at boot (am335x-boneblack.dtb). This involved decompiling the binary, commenting out the section related to I²C2 (starting line 78), recompiling and rebooting.

- **eQEP2.dts** The Enhanced Quadrature Encoder Pulse (eQEP) Module (described in 4.3.1 of the AM3358 chip provides an easy way to implement a quadrature encoder. Two of these modules are provided for on the BeagleBone Black; both on the P8 header. The rotary encoder was placed on the right-hand side of the device necessitating the removal of pins 13 to 26 of the P8 header. While both eQEP modules were still accessible, the placement of the buttons and their connection to pins 27 to 30, meant connecting the encoder to eQEP2 was the parsimonious choice.
- **GPIOs.dts** Enables the pins 7 to 9 on the P8 header as outputs for the LEDs. In addition, pins 27 to 30, also on the P8 header, are configured as inputs. All seven pins are pulled down with no signal.
- **I2C1.dts** With I²C2 disabled to allow for DCAN0, and I²C0 not exposed on the expansion headers, only I²C1 was available to communicate with the display. The data (SDA) and clock (SCL) lines of the I²C1 module are available either at pins 17 and 18 or pins 24 and 26 of the P9 header. To maximise space for the ground plane (see ??) the module was connected to pins 24 and 26.
- **PRU.dts** The Programmable Real-Time Units are also enabled over a device tree overlay. In this case both PRUs were enabled, although only one is used. Furthermore, a debug pin (pin 27 on header P9) was enabled to help verify the timer instantiated on the PRU.

5.3 User Interface

The user interacts with the device through an options menu with the help of a rotary encoder and display (described in 4.3 and 4.5).

5.3.1 I²C and LCD

Header and source files (lcdDisplay.h and lcdDisplay.cpp) were developed to take care of the I²C interface to the display and the commands for the LCD itself. The contents of the files is heavily based on a python library for Raspberry Pis with similar functionality [16]. Additional functions that were added, improve the programmer's options in terms of which types can be placed on the display (eg int and float), and the position at which they can be displayed (left or right-hand side).

Following the boot process and the launch of the underlying device software, two things happen related to the display. First, access to the I²C bus is opened and communication with the slave LCD device is initiated with its I²C address. This occurs with the `initI2C()` function. Second, the `initLCD()` function initialises the LCD, a display that uses the Hitachi HD44780 protocol, in 4-bit interface mode [17] (Listing 1). At this stage the LCD is initialised and ready to display the menu or messages to the user.

Listing 1: Accessing the I²C bus and initialising the display

```
167  /*** Initialise the LCD ***/  
168  initI2C();  
169  initLCD();
```

5.3.2 Rotary Encoder and Button

The C++ eQEP API written by Nathaniel Lewis [18] was used as the interface to the eQEP driver. The main function creates an instance of the eQEP class based on the root path of the eQEP2 sysfs entry, and the operating mode of the hardware. In this case the eQEP hardware is run in absolute mode (Listing 2).

This means its position starts at zero and is incremented or decremented by the encoder's movement, as opposed to being reset after a time interval.

Following instantiation, the period at which the eQEP hardware polls the input is set with the `set_period()` function. Smooth operation of the encoder with the menu was achieved with a period of 10ms(or 10,000,000ns).

Listing 2: Setting up the encoder

```
173 // Create quadrature encoder
174 encoder = new eQEP(eQEP2, eQEP::eQEP_Mode_Absolute);
175 encoder->set_period(10000000L); // Encoder polled every 10ms
```

5.3.3 Menu

A menu was created to allow a user of the device to select and change parameters of the test rig. The underlying structure entails a class of menu nodes (class `MenuNode`, described in `menu.h` and `menu.cpp`), in which instantiated `MenuNode` objects contain a vector of further `MenuNodes` (simply named `v`). Each `MenuNode` has a `useNode()` function that is defined with a function pointer at object instantiation. In this way, the user can pass the `MenuNode` a function defining what should happen when the node is "used".

Conceptually one can differentiate between internal and terminal nodes, although this distinction isn't made in the software. Internal nodes will, when their `useNode()` function is called, display the contents of their `MenuNode` vectors, allowing the user to navigate further into the menu. In contrast, terminal nodes will display settings that can be changed by the user.

Besides a *name* variable and the aforementioned vector, `MenuNode` objects contain the member variables `vectorCursor` and `displayOffset`. The first, is essentially an iterator for the vector. When linked with the encoder input, this variable allows the user to navigate the vector, and by extension, the menu. The `displayOffset` variable is needed for displays that have fewer lines than the length of any given `MenuNode` vector. In these cases, the lines that are displayed need to change depending on the `vectorCursor` position in the vector.

The display and navigation of MenuNode vectors is facilitated with the two functions `displayMenu()` and `updateDisplay()` (at the end of `main.cpp`). These functions use the member variables `vectorCursor` and `displayOffset` to display the appropriate nodes of the MenuNode vector and cursor position on the LCD.

5.4 Control System

5.4.1 Analogue to Digital Conversion

5.4.2 CAN bus

5.4.3 PID Controller with PRU Timer

5.5 Data Storage

6 Data Acquisition

7 Discussion

8 Conclusion

References

- [1] G. Coley, *BeagleBone Black System Reference Manual, Revision C.1*, May. 2014
- [2] Texas Instruments, *LM2596 Simple Switcher[®] Power Converter 150kHz 3A Step-Down Voltage Regulator*, 2013
- [3] Vishay General Semiconductor, Appl. Note 88436, *What is a Silicon Transient Voltage Suppressor and how does it work?*, 2007
- [4] Analog Devices, Appl. Note 1368, *Ferrite Bead Demystified*, 2015
- [5] Fairchild Semiconductor, *SMBJ5V0(C)A - SMBJ170(C)A 600 Watt Transient Voltage Suppressors*, SMBJ24A datasheet, 2003
- [6] NXP Semiconductors, Appl. Note 10441, *Level shifting techniques in I²C-bus design*, 2007
- [7] H. Zumbahlen, *Linear Circuit Design Handbook*. Amsterdam, Netherlands: Elsevier Ltd, 2008
- [8] Texas Instruments Technical Staff, *AM335x ARM[®] Cortex[™]-A8 Microprocessors (MPUs) Technical Reference Manual*. Texas Instruments, Oct. 2011 [Revised Apr. 2013]
- [9] Texas Instruments, Appl. Note SLLA270, *Controller Area Network Physical Layer Requirements*, Jan. 2008
- [10] Texas Instruments, *SN65HVD23x 3.3-V CAN Bus Transceivers*, Mar. 2001 [Revised Jul. 2015]
- [11] *Road vehicles – Controller area network (CAN) – Part 2: High-speed medium access unit*, ISO 11898-2:2003
- [12] Texas Instruments, *AM335x Sitara[™]Processors*, AM335x datasheet, Oct. 2011 [Revised Apr. 2016]
- [13] D. Molloy, *Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux*, 2015

- [14] devicetree.org, "The Device Tree Specification," 2016 [Online]. Available: <http://www.devicetree.org> [Accessed: Nov. 07, 2016].
- [15] eLinux.org, "Capemgr," Sep. 20, 2015 [Online]. Available: <http://www.elinux.org/Capemgr> [Accessed: Nov. 07, 2016]
- [16] Captain Stouf, "Raspberry Pi: using a 4x20 characters display," Mar. 20, 2014 [Online]. Available: <http://hardware-libre.fr/2014/03/en-raspberry-pi-using-a-4x20-characters-display> [Accessed: Mar. 24, 2016]
- [17] Hitachi, *HD44780U (LCD-II), Dot Matrix Liquid Crystal Display Controller/Driver*, HD44780U datasheet, Sep. 1999
- [18] Nathaniel Lewis, "C++ eQEP API," May. 15, 2014 [Online]. Available: <https://github.com/Teknoman117/beaglebot/tree/master/encoders/api/c%2B%2B> [Accessed: Apr. 21, 2016]