Prototype Development Using LSM303, Stepper Motor and ADC Module on Raspberry Pi

Shifa Shaikh, MS.
Electrical Engineering Department, College of Engineering
San Jose State University, San Jose, CA 94303
E-mail: shifabanumohammedrafiq.shaikh@sjsu.edu

Abstract

To build a prototype system using Raspberry Pi, PWM motor drive, NEMA stepper motor, LSM303 sensor and an ADC module with a potentiometer. Additionally, to operate this prototype system by changing the output to the ADC input, so ADC can convert the input to the digital output, and then change PWM output to the driver board to actuate stepper motor. Validation of ADC data is also shown in the paper.

1. Introduction

The project should drive the stepper motor depending on the input from the potentiometer and the ADC module. It simulates a sensor that provides an input to the open loop system, which is stepper motor in our case. We can see the overall implementation as a closed loop system as the LSM30; three axis accelerometer + magnetometer + temperature sensor provides a feedback for the position of the stepper motor rotation.

The data from the ADC need to be validated. So, compensation function is calculated for the validation and power spectrum is plotted.

f(x) + g(x) = ax + b … (1)

Where ax + b is the ideal linear characteristics of the ADC, f(x) is the actual ADC function (piece wise linear), and g(x) is the compensation function to be designed. Note b = 0, and a is a slop for the ADC, for 10 bits ADC, a = 1024/3.3, f(x) is from the ADC data characteristic curve and g(x) is the compensation function to be designed.

So f(x) + g(x) = ax and Based on equation (2), we can write f(x) as f(x) = px + q … (3)

p and q can be found from the actual experiment. Substitute eqn(3) back to (1), we have

px + q + g(x) = ax

px + q + a_g x + b_g = ax

We shall keep the sampling frequency in accordance to the Nyquist theory as follows:

$$f_{sampling} >= 2\ f_{max}$$

2. Methodology

This section shows the systematic implementation of the prototype.

2.1. Objectives and Technical Challenges

The main objective is to drive the NEMA stepper motor using the variable inputs from the ADC module, which is again connected to the potentiometer. The speed is changed according to change from the value of potentiometer.

The challenges that I have been through this project are mainly how to interface the sensor module and the ADC module together on the same I2C interface, validation of data from the ADC and taking a corrective action by increasing the sampling rate was also a tedious job. Driving the stepper motor because the driver module is prone to EMI and so it performs worse when in the proximity of the controller board or the power supply. Continuously monitoring ADC output and thus making the motor spin in desired direction was a challenging job but this overall experience enhanced my knowledge about the sensors and the PID control system used in embedded systems.

2.2. Problem Formulation and Design

The detailed description of the design is in this section. The PWM motor drive and LSM303 sensor are used to drive the NEMA stepper motor and measure the direction and acceleration of the stepper motor respectively. The LSM sensor is mounted on the motor itself so that the movement of the motor can be noted. The direction and the speed (acceleration) are the two main concerns in the project.

We need a stepper motor drive to drive the motor. The PWM output actuates the drive and then drives the motor. The ADC generally does not have one to one mapping of voltage and digital data practically. To have one to one mapping we use potentiometer. By rotating the meter, we can bring down or bring up the value to zero and note the corresponding digital value.

3. Implementation

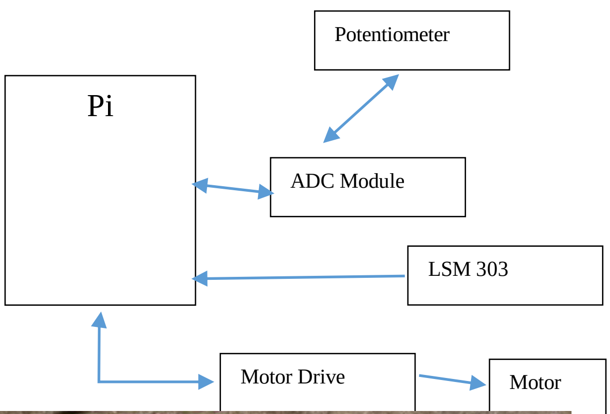
The hardware and software designs are shown in this section.

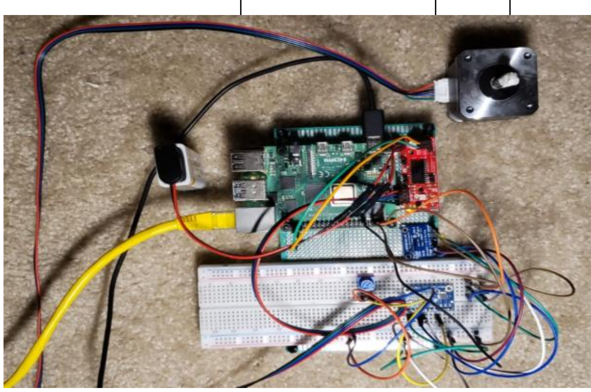
3.1. Hardware Design

List of components:

1) Raspberry Pi 3B+
2) LSM 303
3) ADS 1115
4) NEMA 17 Stepper motor
5) Stepper motor drive module
6) Connecting Wires
7) PCB

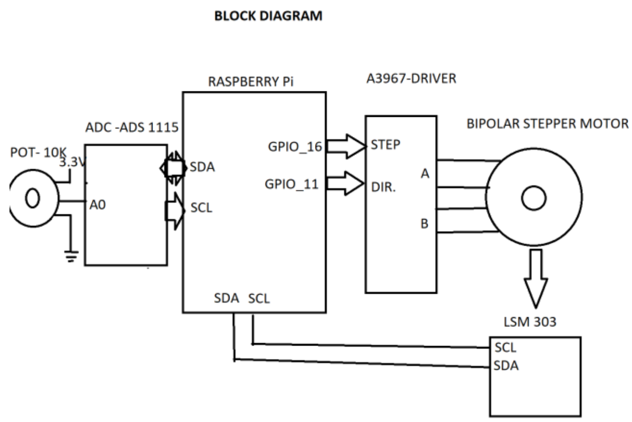
System Block Diagram:





Prototype Integration Picture:

Schematic Diagram:



3.2. Software Design

The software design is divided into two sections viz. User application program and Kernel space program. The User application program first sets the GPIO port P0.4 as the output port, which is used to decide the direction in which the motor should rotate. Secondly, the Potentiometer output is given to the analog input pin of the ADC module. There is a function, which checks for the interrupt. Whenever there is an interrupt the buffer is ready and the digital value is stored into a temporary variable. The output of the ADC is then given as the frequency to the PWM. The PWM output is then given to the motor driver STEP pin, which controls the speed of the motor.

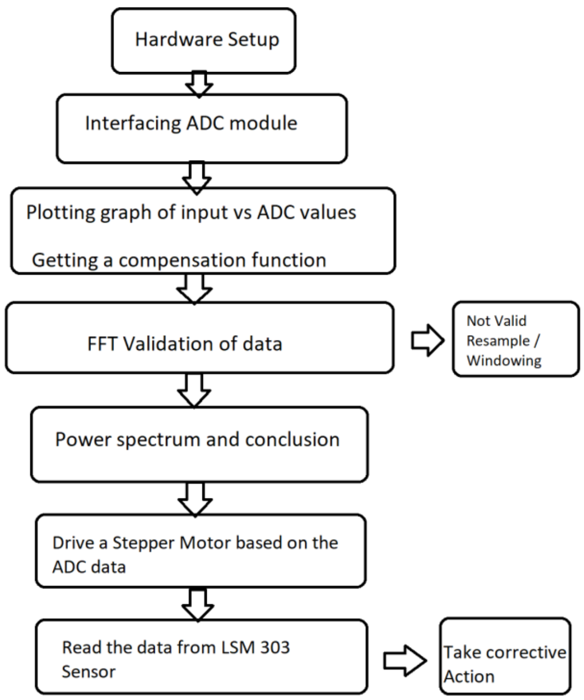
The software development process was divided into two sections. User application program and Kernel space program.

Algorithm:

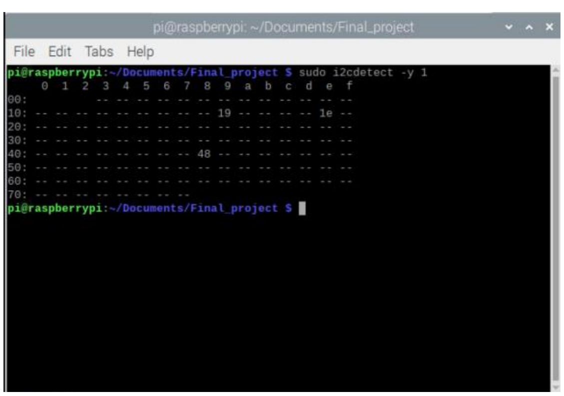
1. Monitor the value in the ADC.
2. Change the value using potentiometer.
3. According to the change in the values, change the speed of the motor attached with the motor drive.
4. Change the direction of the motor.

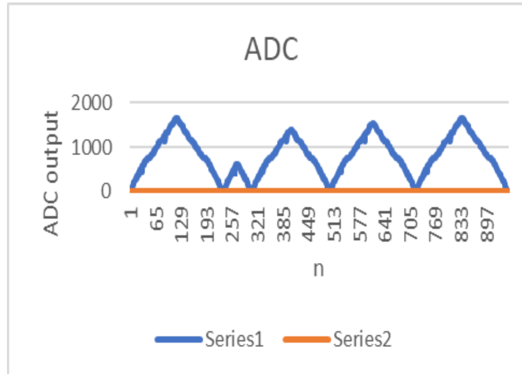
5. Measure the speed and acceleration using LSM303.

Design Flowchart:

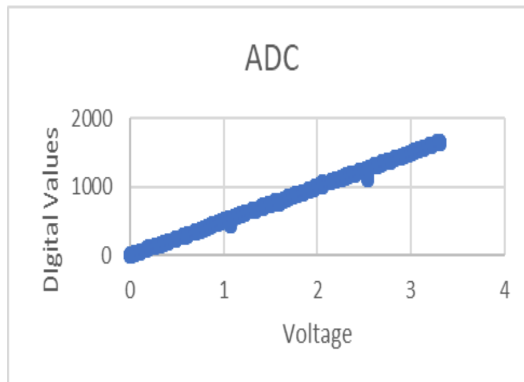


Detection of Slaves connected using I2C:





ADC Output and Voltage:



4. Testing and Verification

1)Input Voltage -Potentiometer

The potentiometer is a manually adjustable variable resistor with 3 terminals. Two terminals are connected to both ends of a resistive element, and the third terminal connects to a sliding contact, called a wiper, moving over the resistive element. The position of the wiper determines the output voltage of the potentiometer.

Moba-Xterm is used as a serial communication between laptop and Raspberry pi via Secure Shell INSMOD the .ko files on putty by using :

STEP 1: KERNEL SPACE:

Compile and build the device driver module using make modules to generate the .ko files for the device driver.

STEP 2: User Space:

Compile and build the user application program to using make to generate the .exe files for ADC

5. Conclusion

The motor successfully rotates in clockwise and anticlockwise directions and the LSM303 measures the acceleration of it.

6. Acknowledgement

I thank Dr. Hua Harry Li for teaching main concepts and guiding to reach the solutions.

7. References

[1] H. Li, "Author Guidelines for CMPE 146/242 Project Report", *Lecture Notes of CMPE 146/242*, Computer Engineering Department, College of Engineering, San Jose State University, March 6, 2006, pp. 1.

[2] Stepper Motor interface with Raspberry Pi. https://www.electronicwings.com/raspberry-pi/stepper-motor-interfacing-with-raspberry-pi

[3] LSM303 interface with Raspberry Pi https://www.digikey.com/catalog/en/partgroup/lsm303-triple-axis-accelerometer-magnetometer-compass-board/54857

[4] ADC interface with Raspberry Pi https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters

8. Appendix

Source code is attached in the zip file.