**Reflex v1.0 Design Document**

Free Guest

4/18/2022

1. **Project Background**
   1. The motivation for this project is based on a project I did as an intern in 2021. I was working on developing a PCB that would function as a data acquisition and compute system for another motorized system. The idea was that this board would read from a couple of pressure sensors and adjust the motors to keep the weight reading from the sensors in a specific tolerance. To keep myself from getting arrested by the corporate police, I won’t disclose much more than that in terms of specifics of the system and to be completely honest that is not that important. What is important though is the functionality and capability of the system.
   2. Essentially what we are dealing with is a closed loop control system. In a closed loop system, there is some input which quite literally feeds back into the system. This input is also sometimes called feedback and, in our case, the feedback is our pressure sensors. So, whenever you hear or see someone talking about feedback in a control system, they mean a sensor input. This feedback allows for the system to see what is going on and make corrections if it is going off course so to speak. The sensors will output a raw analog voltage depending on the amount of weight applied to them. We will be using a microcontroller to process our data so we will need to digitize this signal in order to get it into our processor. After that we (somebody, not me though lol) can do firmware stuff to it. Once the firmware does it’s thing, we send the updated information to the motors to do their thing. Sounds simple right? We’ll let’s talk a little bit more about the system specifications.
2. **System Specifications**

So, this board will need to interface with our sensors first. This will require careful design of this interface in order to allow for our system to be able to accurately read the data. This will require an analog signal conditioning chain before being converted into a digital value for the processor. The signal will need to be filtered first to remove any unwanted frequencies and prevent aliasing. Pressure sensors often require much higher resolution compared to other sensors so we won’t be able to use the ADCs on the microcontroller. We will need to select an external ADC with >= 24 bits of resolution to accomplish this task.

This system will need to be flexible enough to interface with other devices. Since this system was intended to be used in a manufacturing environment, it is a good idea to include as much flexibility as possible as you never really know who may end up using it and in what way. Based off that, I decided to use UART, JTAG, USB, and WIFI for my offboard communications.

UART is a very common asynchronous microcontroller communication protocol and is always good to have on your system due to its simplicity and reliability. In our case, we will use UART for debug or an alternative to usb.

USB is used by every modern computer system and allows for easy connection to a computer or another host device. I will use the 5V phantom power from the usb to allow for the user to power the device using usb. It will also give flexibility in programming and debugging as we can use the usb cable if you want without using JTAG.

JTAG is used for in circuit programming and debugging. For our case, we have the option of using serial wire debug or swd using a jtag connector which simplifies the signal count.

WIFI is a will give us the flexibility to have a wireless link to the device. We will be able to connect to a network to upload data to a server or even control it remotely

The system can either run with a host computer or without one. When it is connected to the computer, it will communicate its status with it. In the case that it is not connected to a host computer, we will be using a lcd display on our board to display status and other updates.   
  
For power, we will have the option of running with a dedicated power supply or using usb power. The sensor interface will need to be powered at 5V or greater since the pressure sensor requires an excitation voltage.

For this first version, we will only be focusing on the controller and will add the motor controller later. For future updates though, it will be good to have a high speed link between the motors and mcu for real time motor control.

1. **Component Selection**
2. **Block Diagram**
3. **Schematic**
4. **Schematic Testing and Validation**
5. **PCB Stack Up**
6. **PCB Layout**
7. **PCB Simulation**
8. **Signal Integrity**
9. **Manufacturing**
10. **PCBA**
11. **PCB Bring up and Validation**
12. **Final Thoughts and Lessons Learned**