Automatic Digital Blood Pressure Cuff

Fall 2024

Measuring a person's blood pressure is a standard diagnostic test that doctors use to help assess the person's health. When done in a doctor's office, this measurement typically involves inflating a cuff surrounding the person's upper arm until no blood flow sounds are detected in the lower arm by a stethoscope. The pressure in the cuff is then slowly released. The pressure at which blood flow sounds first return is the systolic blood pressure (maximum pressure exerted by the heart) and the pressure at which the sounds can no longer be heard is the diastolic blood pressure (minimum pressure when the heart is resting). This type of measurement is a bit complicated for untrained people in a home setting, so automatic measurement devices that require no expertise to use have been developed. You will design and implement such a device for this project.

Since this device has many similarities to other biomedical measurement devices (SpO2, ECG, EEG), this project will serve as a good introduction to biomedical instrumentation.

Project Description:

A basic block diagram of the system is shown below in Figure 1. The measurement will be initiated by pressing a start switch, after which the cuff will inflate to an appropriate pressure and then the pressure will be slowly reduced. The measured pressure curve will look similar to the blue curve in Figure 2. The small ripples in the blue pressure curve as the cuff deflates are critical to determining the blood pressure and will need to be amplified and filtered to produce a curve similar to the red curve in the figure, referred to as the oscillometric pulse. The fundamental frequency of this curve is your heart rate. Both the overall pressure curve and the oscillometric pulse signal need to be amplified and filtered sufficiently so that they can be read into the Arduino through the analog to digital converter (ADC) as accurately as possible. This means that they should span as much of the 0V to 5V range of the Arduino's ADC as possible and any extraneous noise, such as 60Hz interference from the power line, should be reduced to a negligible level. These pressure signals will then be processed using the ideas in reference [2] to compute and display the systolic and diastolic pressures as well as the heart rate. Blood pressure is typically measured in units of millimeters of mercury (mmHg), so you will have to figure out how to convert your measured voltages into mmHg.

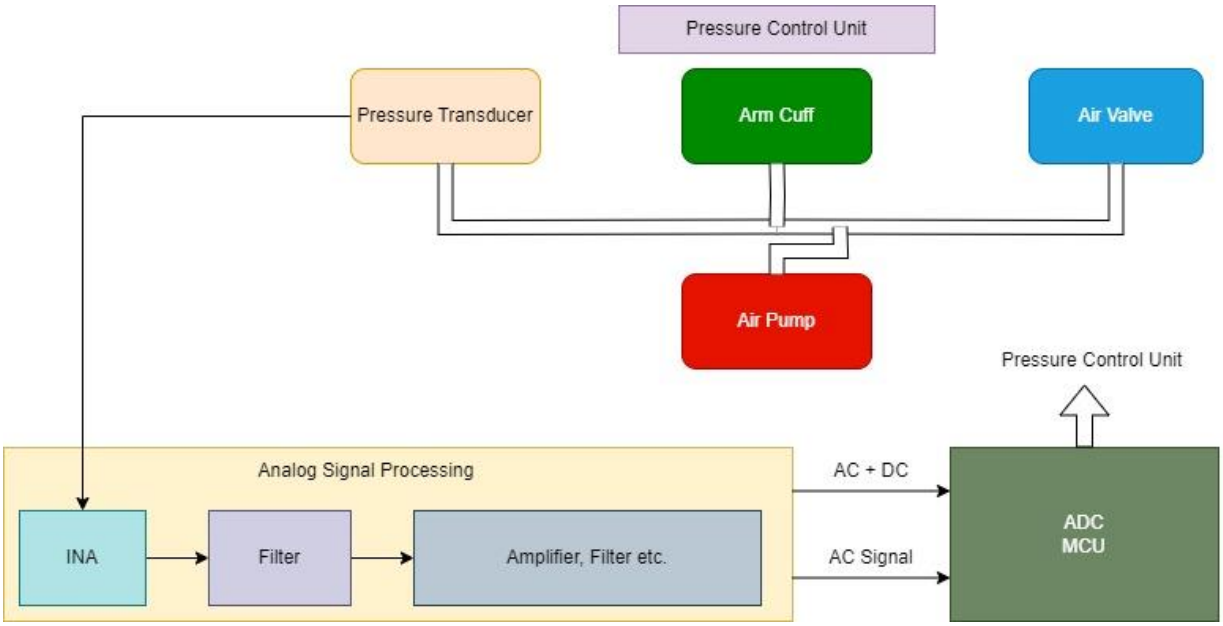


Figure 1: Block Diagram of BPM

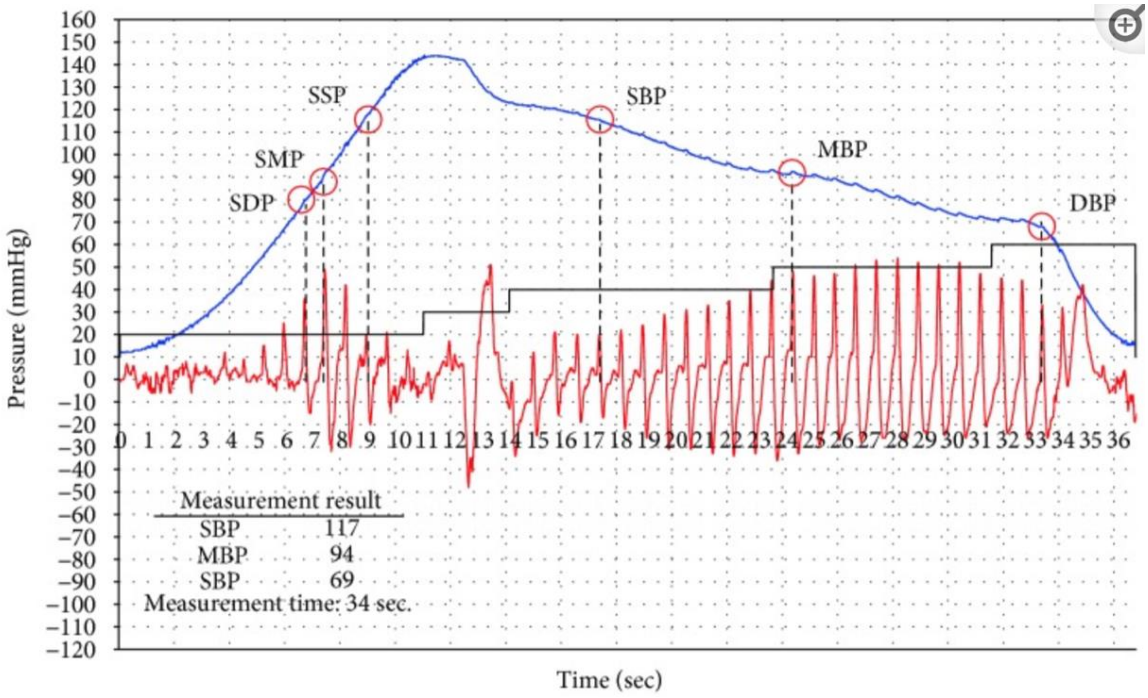


Figure 2: Pressure Signal and Oscillometric signal vs time [2]

You will be provided with an inflatable arm cuff and an air pump to inflate it. Once the cuff has been inflated to the desired pressure, you will trigger an air valve to deflate the cuff. Opening the air valve will deflate the cuff too quickly to capture an oscillometric pulse signal of appropriate

length, so you will be given a small c-clamp so that you can reduce the deflation rate as desired. The pressure will be measured by a pressure sensor. All of these components will be connected by plastic tubing, as shown in Figure 1. The pump and valve are 3V devices, so you cannot drive them directly with 5V.

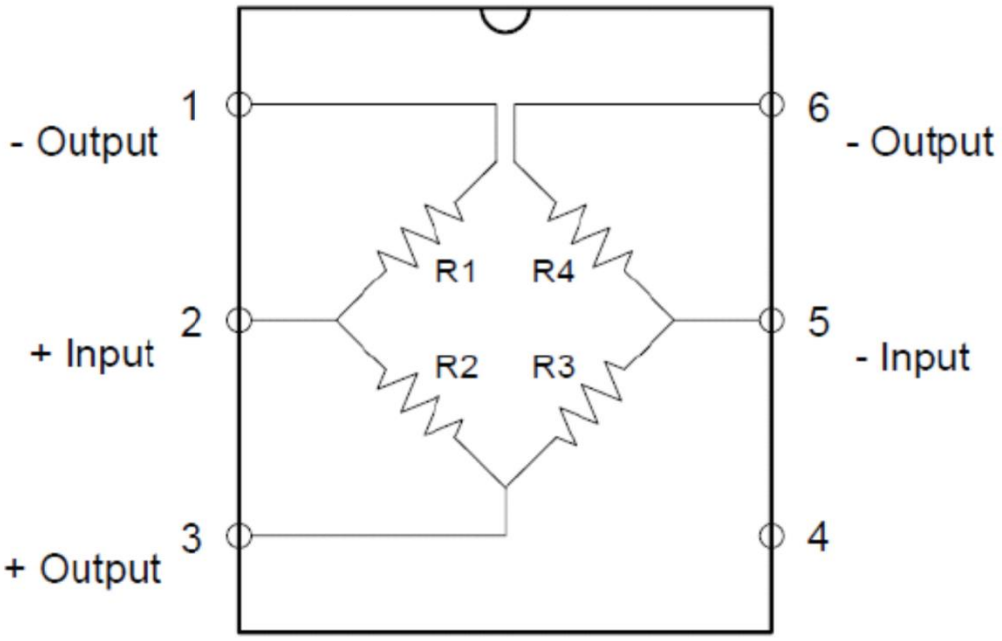


Figure 3: Wheatstone bridge within the pressure transducer [1]

The pressure sensor contains a pressure transducer, which changes its resistance (R1) in response to the applied pressure, configured in a Wheatstone bridge, as shown in Figure 3. The output of the sensor is the voltage difference between the two legs of the bridge. To measure this difference without altering it requires a differential amplifier with a very high input impedance. An instrumentation amplifier meets these requirements: it is a differential amplifier, usually with a programmable gain, that has a huge input impedance, usually in excess of 100 megohms.

The oscillometric pulse signal needs to be extracted from the measured pressure signal in order to compute the systolic and diastolic pressures. It is probably easiest to do this using op amp-based analog signal processing. You already know how to use op-amps as amplifiers and how to design passive first-order filters. If you want to design higher-order filters, you may use Analog Device's filter wizard, since you have not been exposed to this material yet. Analog Device's filter wizard is an open-source online filter design tool where you can design different types of active filters. Some tutorials about basic filtering and how to use this wizard are provided in the resources section.

Milestones & Deliverables:

Milestone 1:

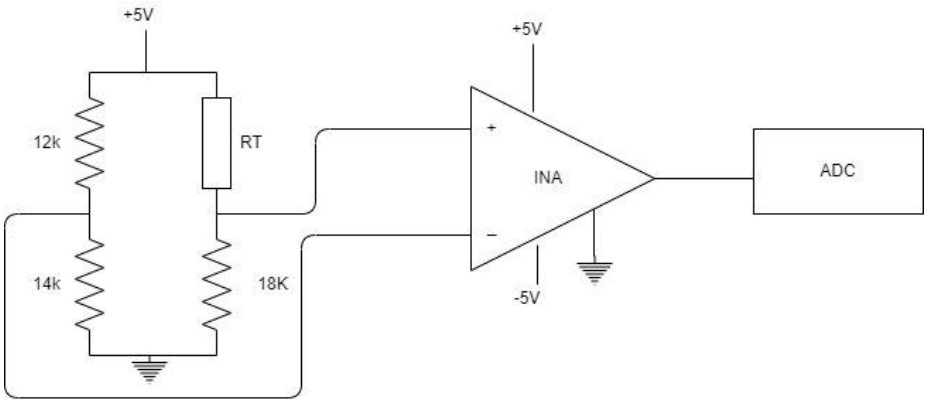
a. In Figure 3, +5V will be supplied to pin 2 and pin 5 will be grounded. The output is the voltage between pins 3 and 1 (pins 1 and 6 will be shorted together). Derive an equation for the output voltage in terms of the resistors. Solve for the unknown resistor R1. Why would you use a Wheatstone bridge to measure an unknown resistance instead of using a simple voltage divider?

b. What are the advantages of using an instrumentation amplifier over a differential amplifier made from a single op amp and resistors? You will be using the INA126P as an instrumentation amplifier in this project. How do you program the gain of this chip? What does the reference voltage of the INA allow you to do? Why might this be useful?

c. A Wheatstone bridge and an amplifier are widely used for precise measurement of physical parameters such as temperature, pressure, light, strain etc. Below is a schematic of a temperature measurement system that operates from 10°C to 40°C. R_T is a thermistor that changes its resistance in response to temperature as given below. The bridge is powered by a +5V supply and the instrumentation amplifier is powered by a +/- 5V supply. The reference voltage pin is grounded, but you can change this if needed. The INA feeds an ADC that operates between 0V and 5V. You want the INA output to cover as much of the 0 to 5V range as possible to increase the precision of the ADC result. Based on INA limitations, you decide that its output should range between 0.1V and 4.1V as the temperature goes from 10°C to 40°C. Find the **amplifier gain and Vref** to achieve this output voltage range. Show all of your work. The thermistor properties are:

- T = 10°C, R_T = 30 Kohm
- T = 25°C, R_T = 20 Kohm
- T = 40°C, R_T = 10 Kohm

  Hint: $V_{meas} = V_{ref} + A_d V_{diff}$



d. Define systolic, diastolic and mean arterial pressure. Describe at least one procedure for determining these pressures from the pressure curve (blue) and the oscillometric pulse (red in figure 2).

e. What type of filtering will you need to produce the oscillometric signal (red in figure 2) from the pressure signal out of the INA? The oscillometric signal will likely be corrupted by 60Hz interference and high frequency noises. Explain where you would put the filter 3dB points and what order of filter you would use.

Milestone 2:

a. Assemble the pressure control unit consisting of the air pump, air valve, arm cuff, and pressure sensor using the supplied tubing and connectors. For your initial work, place the arm cuff around a piece of PVC pipe, both for safety and testing convenience until the system is working correctly. The pump and normally open (NO) valve must be driven from a 3V supply, so you will need to produce a 3V supply from the 5V from the AD2/3 using an LM317 regulator. For safety, you will need to wire a single-pole single-throw kill switch into your system such that turning off the switch will be *guaranteed* to stop power to the pump and air valve, allowing the system to deflate. Check that you can begin inflating the cuff when you apply power to the system and that the kill switch works. Don't inflate the cuff too much. 5 seconds will probably be enough for this testing. (35% of Milestone)

b. Wire up the INA and pressure sensor. Set the gain of the INA so that a 4V output corresponds to 200mmHg. Show all of your calculations. Show a scope trace of the INA output as you apply power to the system and then activate the kill switch when the INA voltage reaches 2V (100mmHg). (20%)

c. Wire your Ardunio, the pushbutton, and any switching transistors (you can't drive motors and valves directly with Arduino pins!) into your system. Write code for the Arduino so that pushing the button starts inflating the cuff and when the cuff reaches 150mmHg the cuff starts deflating. Use the C-clamp to set the deflation time to about 30 seconds. Provide a scope trace of the INA output that captures the entire pressure curve that results from pushing the button to start the system. See the ADC caution below. (35%)

d. Take your blood pressure using the commercial system in your lab, so you get an idea of how it should feel and how high you will need to inflate the cuff to accurately take your blood pressure. Depending on your level of bravery, try your system on your arm or your instructor's arm. Make sure the kill switch is easily accessible. (10%)

Caution: During the arm cuff inflation, ensure that you do not inflate the cuff beyond the desired pressure. For safety, inflate the cuff with the PVC pipe instead of your arm at first and implement a kill switch so you can kill the power quickly.

If you use op-amps with a bipolar (+ and -) supply, then you must ensure that you don't apply a negative voltage below -0.5V to any Arduino pins, as this can destroy the pin circuitry. The simplest way to guarantee this is to connect a Schottky diode between ground and the wire going to the Arduino. The anode should be connected to ground so that any positive signal will be unaffected by the diode, but any negative signal will be clipped to -0.4V.

Milestone 3:

a. Design the necessary filters and amplifiers to extract the oscillometric signal from the total pressure signal (INA output). There will be high frequency noise and interference, including 60Hz hum, on your signal. Design your filters so that the amplitude of the 60 Hz interference does not exceed 1% of the maximum oscillometric signal amplitude. Show scope traces of the total pressure signal and the oscillometric signal, as in Figure 2. Also, provide simulated and measured frequency responses for each filter and list the frequencies of the 3dB points and the amount of suppression at 60Hz. Explain any discrepancies. (50% of Milestone)

b. The Arduino ADC can only handle positive voltages. Design, build and test a circuit that forces the oscillometric pulse signal to stay within the range of 0V to 5V. (20%)

c. Design and explain an algorithm to find the systolic, diastolic and mean arterial pressures. You will implement this algorithm on your Arduino in the next milestone, so ensure that you account for any Arduino constraints, e.g. memory limits, sampling accuracy, etc. How will you calibrate your algorithm? (30%)

Milestone 4:

a. Develop Arduino code for your project to determine the systolic, diastolic and mean arterial pressure from the combined pressure signal and the oscillometric signal. You should display all of these values, as well as the heart rate, via serial monitor. (40% of Milestone)

b. As a first test, record both the pressure and oscillometric signals using your AD2 and calculate the systolic and diastolic pressures, as well as the heart rate, by hand. Compare this calculation with the values output by your Arduino system. (20%)

c. Measure the blood pressure and heart rate of yourself, your partner, and your instructor using both the commercial sensor and your system. Put the results in a table, including the percent error with respect to the commercial monitor for each measurement. Your calibrated system output should not have more than a 4% discrepancy compared with the commercial blood pressure monitor. (40%)

Design Verification:

In your final report, you must verify that your design meets the following specifications:

1. A 4V total pressure signal from the INA corresponds to 200mmHg.
2. The designed, simulated, and measured frequency responses of your filters agree. Provide quantitative data to show this and explain any discrepancies.
3. The amplitude of the 60 Hz hum does not exceed 1% of the maximum amplitude of the oscillometric signal.
4. For a sample blood pressure measurement, show the time domain signal at each stage of your analog processing of the oscillometric signal and discuss why these waveforms look the way they do. These waveforms should not saturate.

Design Validation:

Measure the blood pressure and heart rate of yourself, your partner, and your instructor using both the commercial sensor and your system. Put the results in a table, including the percent error with respect to the commercial monitor for each measurement. Your system output should not have more than a 4% discrepancy compared with the commercial blood pressure monitor. Discuss any significant discrepancies.

Resources:

1. Pressure Sensor:
   https://softroboticstoolkit.com/files/sorotoolkit/files/mps20n0040d-s_datasheet.pdf

   Resources on determining blood pressure from oscillometric signals:
2. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5651164
3. https://ieeexplore.ieee.org/document/7203358
4. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=898494
5. Analog Devices Filter Wizard:
   https://www.analog.com/designtools/en/filterwizard/
6. Some video tutorials about the filter wizard:
   https://www.youtube.com/watch?v=yCb3UvilJKU
   https://www.youtube.com/watch?v=afkZOWc_xbg
7. Instrumentation Amplifier:
   https://www.ti.com/lit/ds/symlink/ina126.pdf?ts=1722368454331

Parts List:

1. Air Pump
2. Air release valve
3. Pressure Sensor Transducer
4. Instrumentation Amplifier (INA 126)
5. Air hose
6. Arm Cuff
7. PVC hose
8. C clamp
9. SPST kill switch
10. Voltage regulator (LM 317)