

Embedded Systems Interfacing Assignment 08

Due: 31 Ocgtober 2013

Problem 01: Design of an Amplifier

Consider the noninverting amplifier circuit given in Figure 1. The gain of the amplifier is

$$v_{out} = \left(1 + \frac{R_2}{R_1}\right) v_{in}.$$

Suppose it is desired to amplify a signal with a gain K where $1 \leq K \leq 3$. To do this the value of R_1 will be fixed. R_2 will be a digital potentiometer whose value may vary. The wiper settings of the digital potentiometer to be used, the *AD5161BRMZ10 – ND*, are controllable through either an *SPI* or an *I²C* interface.

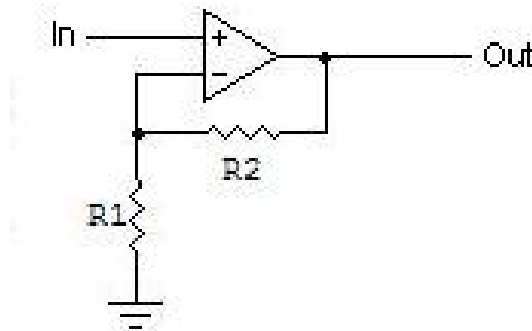


Figure 1: Noninverting Amplifier.

Looking at the data sheet, answer the following questions:

1. Determine the pin numbers on the *AD5161BRMZ10 – ND* for the serial data input/output, the clock, the digital interface select, and the programmable address bit.

2. Determine the pin numbers on the *PIC24FJ128GA010* for the serial data input/output, and the serial clock for using the I²C module.
3. How many bits are used for the device address? What is the device address?
4. What is the maximum I²C clock frequency for the *AD5161BRMZ10 – ND*?

Problem 02: Potentiometer Library

Create functions that do the following:

1. *void initIICResistor(void)* - Create a function that initializes the I²C module using the number of address bits found above, a reasonable clock frequency, and also resets the I²C bus.
2. *int setIICResistance(int N2)* - Create a function that sends the value *N2* to the potentiometer via the I²C port and returns the value of resistance, *R2*, that the potentiometer should have.

Problem 03: Serial Seven-Segment Display

The OpenSegment serial seven segment display (<https://www.sparkfun.com/products/11644>) is a device that will display up to four symbols of data (these symbols can have hexadecimal values). This serial display can be given data over the SPI, I²C, and UART communications protocols. Find the datasheet for this device and determine the following:

1. How many bits are used for the device address? What is the device address?
2. What is the maximum I²C clock frequency for the serial seven-segment display?

Create functions that do the following:

1. *void initIICSevenSeg(void)* - Create a function that initializes I²C module using the number of address bits found above, a reasonable clock frequency, and also resets the I²C bus. This function also ought to enable the SPI port and clear any prior status.
2. *void setIICSevenSeg(char N)* - Create a function that sends the hexadecimal symbol *N* to the seven-segment display using the I²C module.

Write a *main* function that initializes the seven-segment display and in an endless loop clears the display, displays the hexadecimal characters *0x2d40* on the display waits for a second, clears the display, displays the hexadecimal characters *0xd2bf* on the display, and waits for a second. Use any functions that you have previously written to accomplish this task.