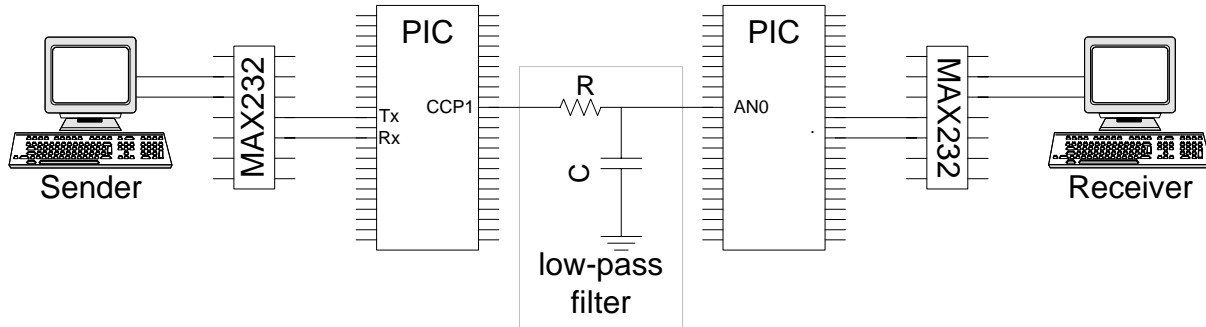


ECE 354 – Lab 4: Analog Communication

This lab uses the A/D and D/A converter that is integrated in the PIC to build an analog communication system. Two PICs will be connected to terminal and each other via a low-pass filter. The PIC shown on the left acts as sender, the one on the right as receiver. After initialization, the user can type a character on the sender terminal. The sender PIC encodes this character into an analog voltage, which is sent out the CCP1 pin. The receiver PIC reads the voltage on the AN0 pin, converts it into a digital value and displays the corresponding character on the receiver.



The setup should operate as follows:

- Both PICs run identical programs. On reset, each PIC should ask if it should operate as sender or receiver. The user can choose by entering a character on the terminal that is connected to each respective PIC.
- Then each PIC asks which coding level should be used. The user has a choice of 1...6. If the chosen coding level is x , then the PIC will allow 2^x different voltage levels on the analog “connection”. The letters that are coded with these voltage levels are ASCII characters $33+2^x$. Example: $x=1$ allows coding of {!,”}, $x=6$ allows coding of {!,”, ...,0, ...,9, ...,,@,A, ...,Z,[,\,],^,_,`}. The voltage ranges corresponding to characters should be evenly spread between 0V and 5V. For $x=1$: !={0V...2.5V}, “={2.5V...5V}. When transmitting a character, you probably want to use the “middle” voltage of the range (e.g., for $x=1$: send A as 1.25V, B as 3.75V).
- Once the coding level as been chosen, the sender will wait for character input from the terminal. When a character is received, it checks if it is in the range of the chosen coding and converts it into an analog voltage. This voltage is maintained until a new character is input by the user.
- The receiver performs an A/D conversion on the AN0 input, translates it into a character and displays it on its terminal. It continues to do this for all new characters (this implies that it cannot detect if the same character is sent twice – you may chose to attempt to fix this (e.g., use 0V as a symbol between characters), but it is not necessary for this lab).
- Both PICs should operate in stand-alone mode and be easily resettable (so we can try different codings).

Tasks

To implement this lab, you should consider the following things:

1. Initialize the UARTs of both PIC correctly.
2. Initialize the PWM and A/D units correctly.
3. For D/A conversion, use the PWM unit with a PWM settings that achieve 10-bit precision.
4. To change the D/A output voltage, modify CCPR1L and CCP1CON<5:4>.
5. On the receiver PIC, perform A/D conversions continuously. Compare A/D output to previous result and print character if it changed. (To allow for slight fluctuations in voltage, you may choose to only print a character if the voltage changes enough to represent a different character.)
6. To convert between ASCII characters and 10-bit digital values, consider using the “rotate” instructions (aka “shift” instructions). If the coding level is x then you basically need to determine the x most significant bits of the 10-bit value. To do that you can take the ASCII value of your character and subtract 65. This leaves you a value in the range of $\{0 \dots 2^x - 1\}$. Then you shift it left by $10 - x$ bits to move it to the most significant bits. Of course, you do not have 10-bit registers, so you will have to do it a bit differently.
Similarly, when you receive a value, shift it to the right by the correct number of bits (depends again on x) and add 65 to get an ASCII character.
7. Consider using an oscilloscope to verify the voltages that you generate.

Note: There is a shortage of serial connectors in the lab. You probably won't be able to continuously use two connectors. Consider sharing your connector with another group. It might also be possible to connect some wires directly to the serial cable from the PC. If you need to in-circuit-debug both PICs, you will need to use two workstations.

Demo

During the demo, your PIC should communicate robustly (for small x) in stand-alone mode. You should set up your demo with an oscilloscope that shows the voltage on the analog wire.

Lab Report

The lab report guidelines are the same as for prior labs. (No logic analyzer printout required.) Include the settings for the PWM and A/D converter that you used and an explanation how you derived them. Also discuss your low pass filter design.