Temp Sensor Data collection Board

Overview:

The purpose of this project is to create a data collection tool for many temperature sensors. When we thermally test our satellite we will need to place temperature sensors on many points of interest, however each sensor has three wires so sometimes it's not possible to rout every wire to an Arduino. Therefore this board will implement an Analog to Digital Converter (ADC) which takes the analog voltages of the temperature sensors as an input, and stores each measurement in a register, then communicates to an Arduino over a two wire, I2C, interface.

Deliverables:

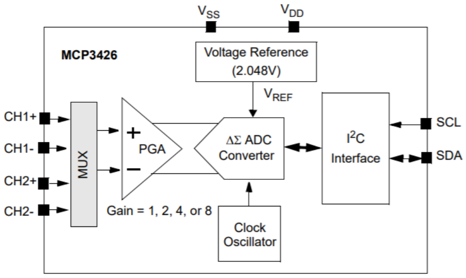
1. PCB board which implements an ADC, has at least three temperature sensor inputs, and an I2C output.
2. Wrapper (C library) which implements the needed functions of the ADC, converts voltage measurements to temperature, and can be used by an Arduino
3. (if time) Arduino code which uses the Board and Wrapper to demonstrate temperature sensing.

Given Temp sensor will be: http://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf

PCB Board:

Eagle resources: Sparkfun has good tutorials: https://learn.sparkfun.com/tutorials/using-eagle-schematic#introduction

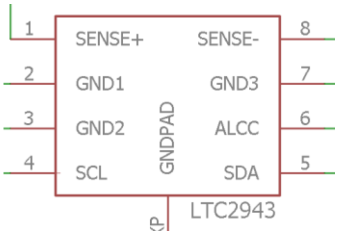
- Step 1: Choose an ADC
  Example of ADC internal workings:

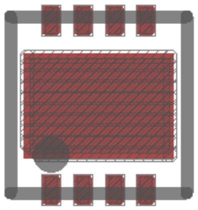


  An analog to digital converter compares an input voltage to an internal reference voltage and assigns a register value to that percentage.

  - Required features:
    - At least 3 inputs
    - 5v power
    - I2C output
    - More than 8 bit register size (resolution)
    - Surface mount component.
- Step 2: Create an Eagle Component for your chosen ADC. (also download eagle if you haven't already)

    - A “component” in eagle is a combination of a schematic symbol and an associated footprint. A schematic symbol has pins which you connect to other components in the schematic, just like normal circuit diagrams. However a footprint is what you see on the actual board. It is the copper pads which the pins of your surface mount component will connect to. Information about the footprint will be on the last few pages of the datasheet.
    - Example: Schematic: Footprint:





- Step 3: Create Circuit schematic
    - The schematic defines how component pins are connected to each other through “nets”
    - Follow the application circuit in the datasheet of your ADC.
    - READ THE DATASHEET CARFULLY
- Step 4: Layout your board
    - A PCB board is a Layer based design. For example there will be a layer called “Top” which represents copper on the top of the board and a layer called “bottom” which represents copper on the bottom of the board.
    - Lookup good PCB design practices to follow, or/and ask Sean to review your layout.
- Step 5: Prepare your board for order.
    - See Sean or Paul.
- Keep the board as small as possible.

Wrapper

A wrapper, also called a Library, is a file of pre-written commands, or “functions”, that implement features on a chip. Anyone using the board won’t have to understand the details of the chip, they would only have to call the function “ADC.readTemperature(1)” in their Arduino code to find out the voltage on that input of the ADC.

A good resource for writing a Library for Arduino is here:
https://www.arduino.cc/en/Hacking/LibraryTutorial

The wrapper should have at least the following functions.

- insertNameOfADCHere(…Configuration variables);
    - This function creates an object and configures the ADC with any settings outlined in the datasheet.
- double readTemperature(int input#)
    - This function reads the voltage on a given input and converts that voltage to a Temperature.

You can include helper functions that make the above functions easier to implement. For example a function “readRegister(int reg)” would probably be helpful.