PURDUE UNIVERSITY DEPARTMENT OF PHYSICS

Physics 53600

Electronics Techniques for Research

Now in PowerPoint!

Spring 2020 Semester

Prof. Matthew Jones

The usual ANNOUNCEMENT

- Obvious changes to the course:
  - No in-person lectures: you'll have to read the lecture notes yourself
  - No more labs: don't worry about it – your grade will be based on work done so far
  - Remaining assignments will try to cover topics that would have been explored in the lab
  - Second mid-term: simplest to cancel it
  - Final exam: I'm not sure what to do about this yet, but I'll figure something out.
- Changes to grading scheme:
  - Old scheme: Assignments (30%) exams (40%) lab (30%)
  - New scheme: Assignments (50%) exams (25%) lab (25%)

The usual ANNOUNCEMENT

- Because there won't be any in-person lectures, you will have to read the lecture notes yourself.
- To demonstrate that you have read them, you will be required to answer *one or two simple questions* before the next lecture is posted.
- The question will be somewhere (like maybe at the end?) and you just have to e-mail me the answer

  mjones@physics.purdue.edu
- To make this easy, please make your subject look like this:

  "PHYS53600 Lecture xx questions Your Name"
- These will be part of your assignment grade, maybe contributing 10% of your total grade.

More ANNOUNCEMENTS

- Feel free to send me questions about the lecture material if there is anything you don't understand. I'm happy to give more explanation (and I'm soooo bored.)
- Send me e-mail if you think it would be useful to arrange a time as a class to have a time where you can ask questions by video.

LECTURE 21 QUESTION #1

What are some advantages and disadvantages of SPI compared with I^2C serial communication?

Digital Data Transfer

- So far we have considered two types of data transfer:
  - Parallel data transfer where all the data is provided at one time
  - Serial data transfer (SPI) where the data is provided one bit at a time
- Both schemes use one signal (STROBE, SCLK, etc…) to synchronize data transfer.
  - Asserting this signal means that it is safe to sample the data signals.
  - Important to respect the setup and hold times at both ends of the data transfer

Disadvantage of SPI

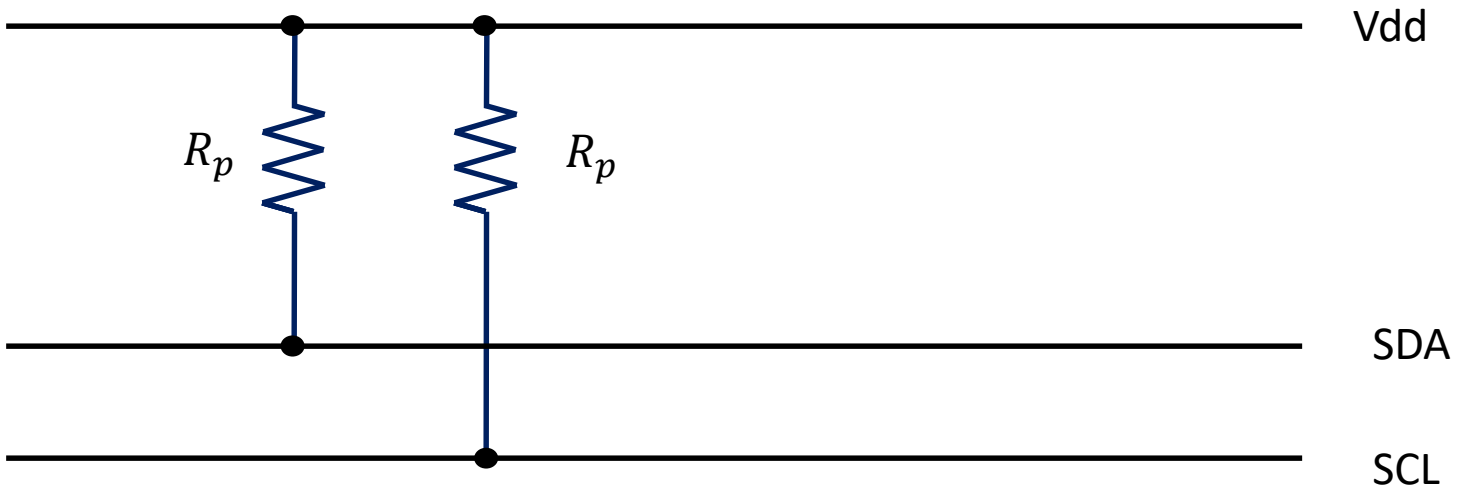
- The SPI interface is simple and convenient if you need to communicate with a small number of peripheral components.
- Each peripheral component requires a dedicated SCS signal to enable it.
- The number of signals needed is 3 + n where n is the number of SCS signals/number of components.
- The I^2C (or IIC) interface only uses two signals, independent of the number of peripheral components.

The Inter Integrated Circuit Interface

- The IIC (or I^2C) interface uses two signals:
  - SCL is like a clock signal
  - SDA is a bidirectional data signal
- Both signals have open collector/open drain drivers
  - A component will either be in a high impedance state, or pull the signal to ground (logic 0)
  - Nothing bad happens when two components drive the signals simultaneously (although data might be corrupted)
  - Both signals require a pull-up resistor that will pull them to Vcc (3.3 volts or 5 volts) when all components are in a high impedance state.

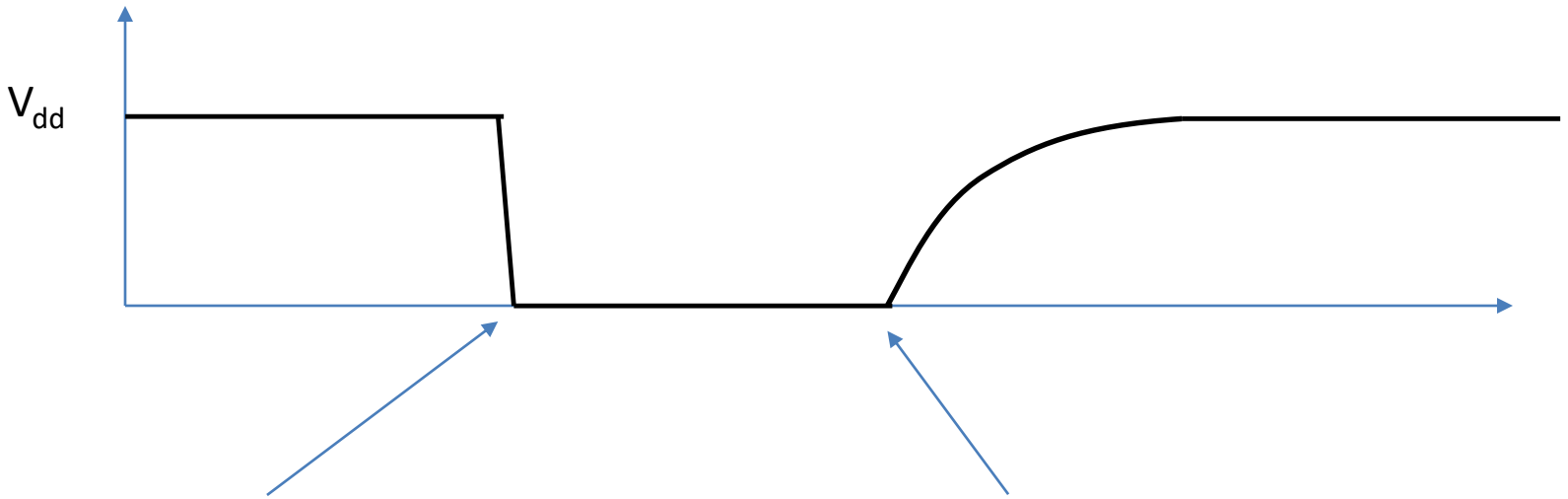
IIC Signals

- Both SDA and SCL signals need to be pulled up to the positive power supply voltage by resistors:



IIC Signals

- Timing diagram:



One of the components pulls the signal low. This creates a low-impedance path to ground and the signal quickly goes to a logic '0'.

When the component goes back to a high-impedance state, the signal charges back up to Vdd. The time constant is $R_p C$ where C is the total capacitance (sum of the input capacitance plus parasitic capacitance of wires or PCB traces)

IIC Signals

- What value of R_p should you use?
  - It is not all that critical…
  - A lower value will make the signal charge up to V_{dd} faster, but will dissipate more power when the signal is pulled low
  - A few kΩ is typical (maybe 2 kΩ to 10 kΩ)
- Typical time constant:
  - If the parasitic capacitance is, say, 400 pF, then the time constant is $\tau = R_p C = 2\ \mu s$
  - The maximum frequency on either signal must be less than about $1/\tau$.
  - Typically, the maximum frequency is 100 kHz (standard mode) or 400 kHz (fast mode).
  - There is no lower limit on the frequency

IIC Signals

- The signals are used for communication in the following way:
  - One component acts as the “master” and drives the SCL signal.
  - Another component acts as a “slave” and receives the SCL signal.
  - Initially, the master drives the SDA signal...

IIC Signals

- Initially, all the slaves on the IIC bus are in an idle state.
  - In this state, all drivers are in a high-impedance state, so both SDA and SCL are pulled high
- They wake up when the detect a “start condition”:
  - The start condition is defined as SDA being pulled low, while SCL remains high.
- They go back to sleep when they detect a “stop condition”:
  - The stop condition is defined as SDA going high when SCL is high.
- Otherwise, SDA should only change when SCL is in a low state.

IIC Signals

- Timing diagram:

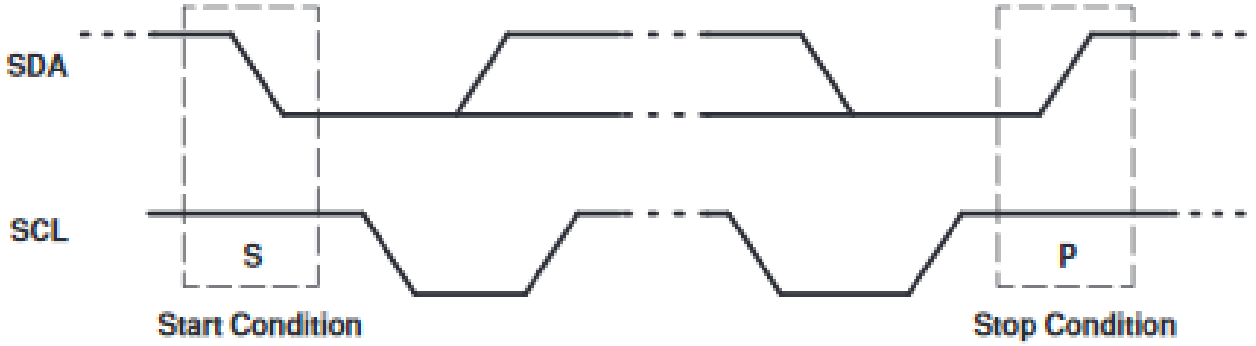


Figure 8. Definition of Start and Stop Conditions

IIC Signals

- Each IIC component has a 7-bit address associated with it.
  - This is usually built into the component but some of the lower-order bits can be changed by pulling pins on the device high or low (or left floating).
- The master first transmits the 7-bit address of the device it wants to talk with, followed by a R/$\overline{W}$ bit.
- The master then releases the SDA signal and looks for an acknowledgement from the slave.
- If a slave recognizes its address, it pulls SDA low (ACK) which the master samples on the next rising edge of SCL.
- If nobody recognizes the address then the SDA bus is pulled high (NACK).

IIC Signals

- Acknowledgment:

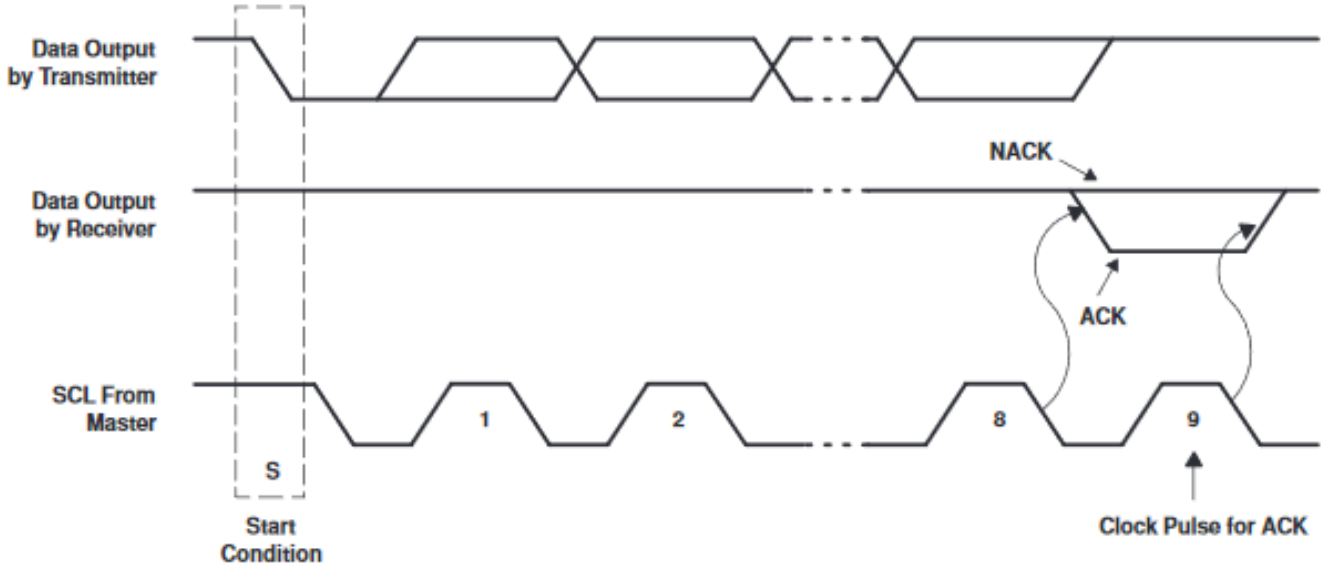


Figure 10. Acknowledgment on the I^2C Bus

- The ACK/NACK is asserted while SCL is low and is sampled by the master on the rising edge of SCL.

IIC Signals

- Here is an example where the master writes data to the slave (for example, a control register):

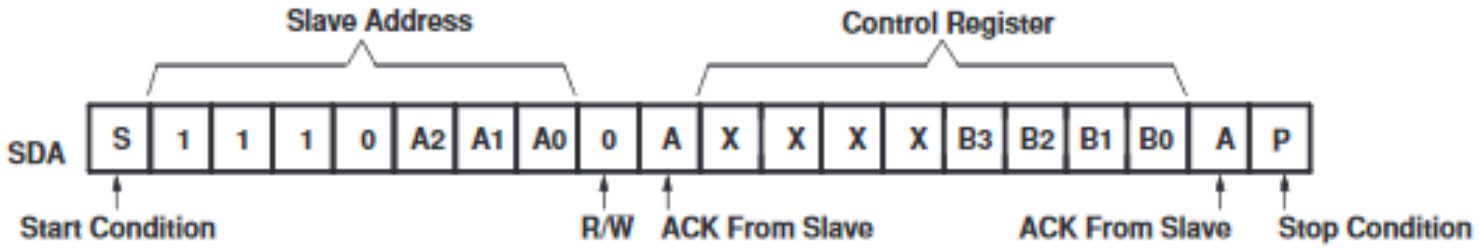


Figure 11. Write Control Register

- After receiving the data, the slave sends another ACK signal.
- The master then terminates the transaction with a stop condition.

IIC Signals

- Here is an example where the master reads data from the slave. In this case, $R/\overline{W} = 1$.

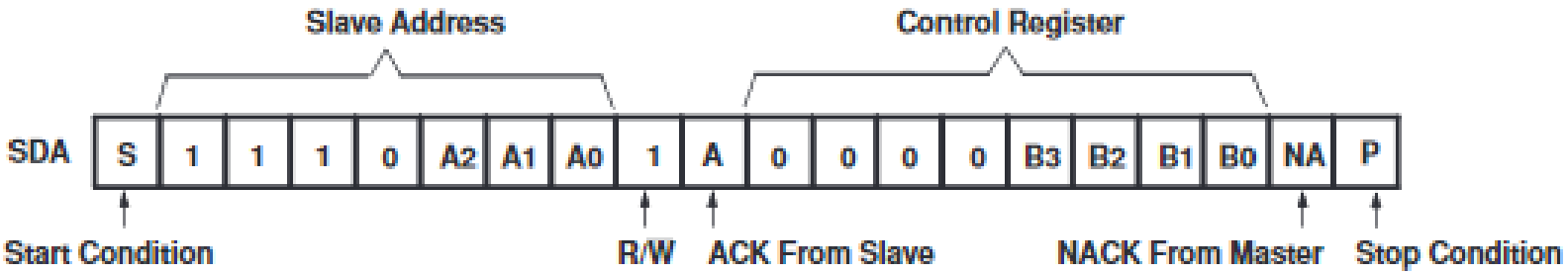


Figure 12. Read Control Register

- The master issues an ACK if it wants to read more data, or a NACK if it is finished.

IIC Protocols

- These examples show how a master and slave can communicate, but the details are often specific to the components being used.
- Let’s look at some examples…

A network of temperature sensors

- Suppose you needed to measure temperatures at several places in an experiment.
- An example of a temperature sensor that communicates using IIC: The LM76 (Texas Instruments)

LM76

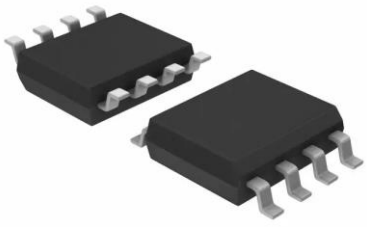
www.ti.com

SNIS109E – JANUARY 2000 – REVISED MARCH 2013

LM76 ±0.5°C, ±1°C, 12-Bit + Sign Digital Temperature Sensor and Thermal Window Comparator with Two-Wire Interface

Temperature Sensor with IIC interface

- The data sheet describes how it works:



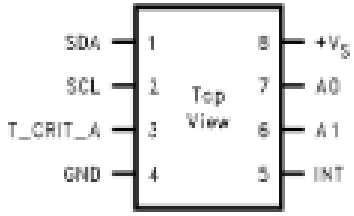
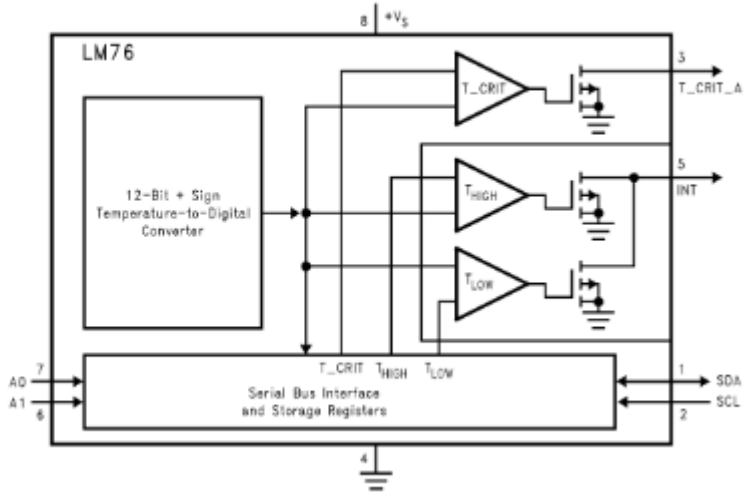


Figure 1. SOIC-8
LM76 See Package Number D

Simplified Block Diagram



PIN DESCRIPTIONS

Label	Pin #	Function	Typical Connection
SDA	1	Serial Bi-Directional Data Line, Open Drain Output, CMOS Logic Level	Pull Up Resistor, Controller I^2C Data Line
SCL	2	Serial Bus Clock Input, CMOS Logic Level	From Controller I^2C Clock Line
T_CRIT_A	3	Critical Temperature Alarm, Open Drain Output	Pull Up Resistor, Controller Interrupt Line or System Hardware Shutdown
GND	4	Power Supply Ground	Ground
INT	5	Interrupt, Open Drain Output	Pull Up Resistor, Controller Interrupt Line
A0–A1	7, 6	User-Set Address Inputs, TTL Logic Level	Ground (Low, "0") or $+V_S$ (High, "1")
$+V_S$	8	Positive Supply Voltage Input	DC Voltage from 3.3V power supply or 5V.

Temperature Sensor Example

- The device has 8 internal 16-bit registers.
  - Register 0 is the actual temperature reading
- Which register is being read/written is specified by setting the “pointer register”:

POINTER REGISTER

(Selects which registers will be read from or written to):

P7	P6	P5	P4	P3	P2	P1	P0
0	0	0	0	0	Register Select		

P0–P2: Register Select:

P2	P1	P0	Register
0	0	0	Temperature (Read only) (Power-up default)-
0	0	1	Configuration (Read/Write)
0	1	0	T_{HYST} (Read/Write)
0	1	1	T_CRIT (Read/Write)
1	0	0	T_{LOW} (Read/Write)
1	0	1	T_{HIGH} (Read/Write)

P3–P7: Must be kept zero.

Temperature Sensor Example

- The data sheet describes how to read the temperature:

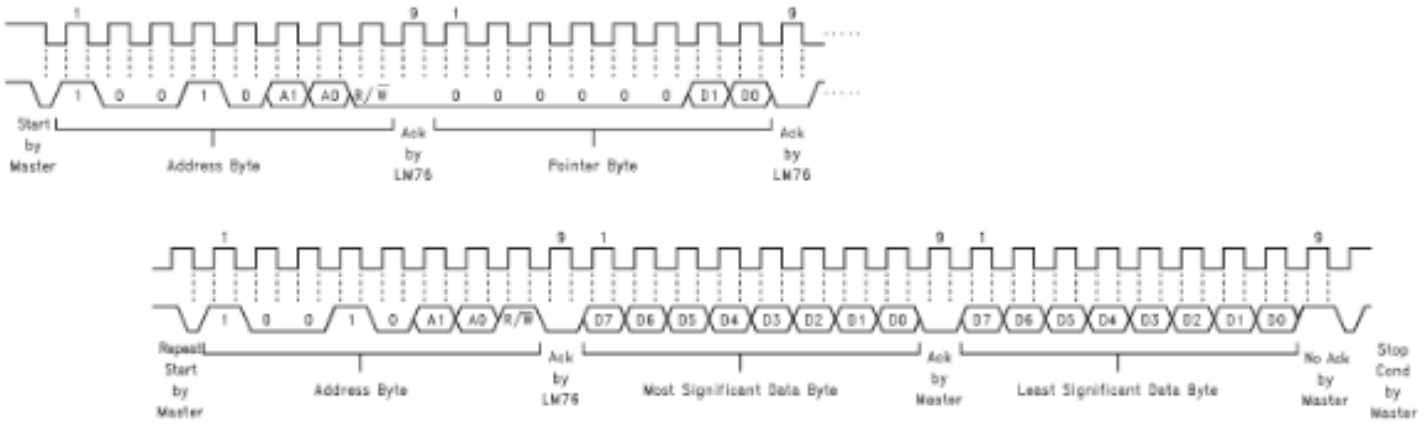


Figure 9. Typical Pointer Set Followed by Immediate Read for 2-Byte Register such as Temp or Comparison Registers

Temperature Sensor Example

- The format of the data is specified in the data sheet:

TEMPERATURE REGISTER

Table 1. (Read Only):

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Sign	MSB	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	CRIT	HIGH	LOW
													Status Bits		

D0–D2: Status Bits

D3–D15: Temperature Data. One LSB = 0.0625°C. Two's complement format.

TEMPERATURE DATA FORMAT

Temperature data can be read from the Temperature and Set Point registers; and written to the Set Point registers. Temperature data can be read at any time, although reading faster than the conversion time of the LM76 will prevent data from being updated. Temperature data is represented by a 13-bit, two's complement word with an LSB (Least Significant Bit) equal to 0.0625°C:

Temperature	Digital Output	
	Binary	Hex
+130°C	0 1000 0 010 0000	08 20h
+125°C	0 0111 1101 0000	07 D0h
+80°C	0 0101 1010 0000	05 90h
+64°C	0 0100 0000 0000	04 00h
+25°C	0 0001 1001 0000	01 90h
+10°C	0 0000 1010 0000	00 A0h
+2°C	0 0000 0010 0000	00 20h
+0.0625°C	0 0000 0000 0001	00 01h
0°C	00 0000 0000	00 00h
−0.0625°C	1 1111 1111 1111	1F FFh
−25°C	1 1110 0111 0000	1E 70h
−55°C	1 1100 1001 0000	1C 90h

Temperature Sensor Example

- Suppose you needed to immediately report that a temperature exceeds some critical value…
- You can do this by programming the other registers.
- When the temperature exceeds the value in the T_CRIT register, the device will pull the T_CRIT_A output low.
- How you use this signal is up to you… maybe you want to make it turn on a light or something.

Complete Design Example