Connecting Sensors and Actuators

Adwait Dongare & Ian Hartwig

This is a quick guide to help with picking, interfacing & connecting various sensors & actuators with your micro-controller. It's slightly different and expanded from the post by Ian on Piazza

---

Datasheets

Every device you will use will have a datasheet associated with it (some of those parts may be very common any may have multiple datasheets from different manufacturers). This is the first document you should grab for any part since it would completely describe all the relevant characteristics. Always keep the data sheet of all the parts on your PCB handy (including the micro-controller)

Here are some examples of what datasheets for processors, sensors or actuators look like:

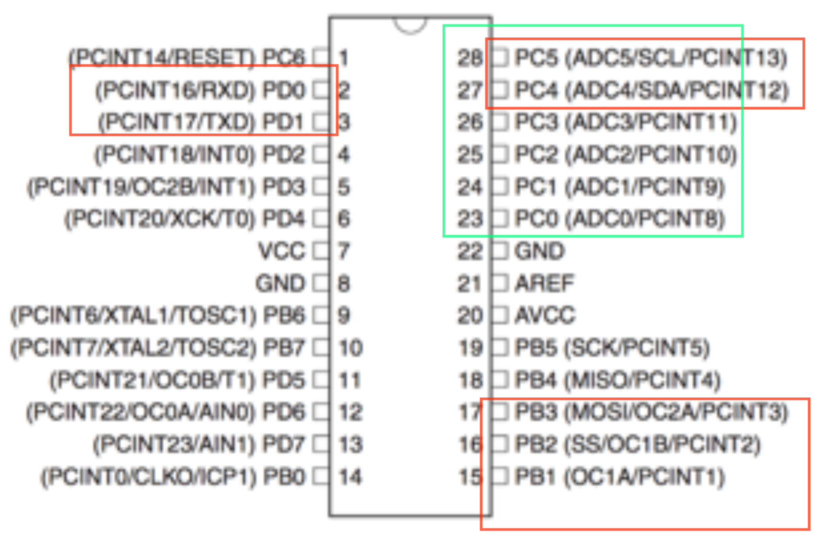
- Atmega 328 (Make sure to keep a PDF of this with you): http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet.pdf
- CO sensor: https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf
- A relay switch: http://www.omron.com/ecb/products/pdf/en-g5le.pdf
- 24-bit ADC: http://www.ti.com/lit/ds/symlink/ads1232.pdf
- Servo motors: http://www.parallax.com/sites/default/files/downloads/900-00005-Standard-Servo-Product-Documentation-v2.2.pdf
- Voltage regulator: http://www.micrel.com/_PDF/mic5209.pdf

If in doubt, always refer to the datasheet. Sometimes there are corrections in data sheets for parts which have been in production for a long time (though major changes are unlikely). Check that the manufacturer hasn't released any errata documents or newer revisions than the one you found. Datasheets on manufacturer websites tend to be up-to-date and provide relevant links to errata.

Interface

There are a number of interfaces used to communicate with sensors, actuators and other micro-controllers (ATmega 328 pins used are mentioned in brackets):

- Serial or UART (RX/TX)
- SPI (MISO/MOSI/CLK/SS)
- I2C or 2-wire interface (SDA/SCL)
- Analog (Any ADC pin for input to microcontroller. You need to do some



engineering if a device needs analog inputs)
- GPIO (any pin on your micro-controller)
- Pulse Width Modulation (PWM) (the OC* pins or any other pin if you write some good code)
- etc

HELP! My actuator need an analog input
Since the ATmega328 cannot provide analog outputs, this problem could be solved in a couple of ways:
- Hook up an appropriate DAC IC and control the DAC digitally.
- If your inputs don't need to change very rapidly (around 10 times/sec), you could also use a low-pass filter hooked to PWM outputs to generate an approximately steady analog output.

Note that it is always a good idea to use an analog low-pass filter even with a DAC to smoothen out your output waveform.

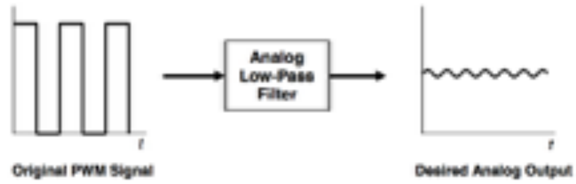


Figure 2. Analog Filtering of PWM Signal

In case of some sensors like photodiodes, you would need to provide additional circuits around your sensor to ensure it behaves properly under varying conditions. eg. Photodiodes are generally supported with various op-amp amplifiers to ensure they don't change their characteristics as a current starts flowing thru them. Some ways to design such circuits are mentioned here:
http://www.ti.com/lit/an/sboa035/sboa035.pdf
and
http://dkc1.digikey.com/us/en/tod/ADI/Common-Photodiode_NoAudio/Common-Photodiode_NoAudio.html

---

Voltage levels

For Power
Different devices may require power supply at different voltage levels.

e.g. The grab from the relay datasheet mentions that the relay can work from 5-24V

■Ratings

●Coil

Rated voltage	Rated current (mA)	Coil resistance (Ω)	Must operate voltage (V)	Must release voltage (V)	Max. voltage (V)	Power consumption (mW)
			% of rated voltage			
5 VDC	79.4	63	75% max.	10% min.	170% at 23°C	Approx. 400
12 VDC	33.3	360				
24 VDC	16.7	1,440				

Note 1. The rated current and coil resistance are measured at a coil temperature of 23°C with a tolerance of ±10%.
2. The operating characteristics are measured at a coil temperature of 23°C.
3. The "Max. voltage" is the maximum voltage that can be applied to the relay coil.

(atleast it's been tested & calibrated for those)

On the other hand, another grab from the 24 bit ADC datasheet shows it operates between 2.7-5.3V

Both datasheets also mention how much current they would draw at those voltages. Keep an eye on the total current load and ensure you aren't over-stressing it (this can cause your power supply IC to shut down at best or burn out itself and some other ICs at worst)

SBAS350F – JUNE 2005 – REVISED FEBRUARY 2008 www.ti.com

ELECTRICAL CHARACTERISTICS (continued)

All specifications at T_A = –40°C to +105°C, AVDD = DVDD = VREFP = +5V, and VREFN = AGND, unless otherwise noted.

PARAMETER	CONDITIONS	ADS1232, ADS1234 MIN	TYP	MAX	UNIT
Digital					
Logic Levels					
V_{IH}		0.7 DVDD		DVDD + 0.1	V
V_{IL}		DGND		0.2 DVDD	V
V_{OH}	I_{OH} = 1mA	DVDD – 0.4			V
V_{OL}	I_{OL} = 1mA			0.2 DVDD	V
Input Leakage	$0 < V_{IN} < DVDD$			±10	µA
External Clock Input Frequency (f_{CLKIN})		0.2	4.9152	8	MHz
Serial Clock Input Frequency (f_{SCLK})				5	MHz
Power Supply					
Power Supply Voltage (AVDD, DVDD)		2.7		5.3	V
Analog Supply Current	Normal Mode, AVDD = 3V, Gain = 1, 2		800	1300	µA
	Normal Mode, AVDD = 3V, Gain = 64, 128		1350	2600	µA
	Normal Mode, AVDD = 5V, Gain = 1, 2		650	1300	µA
	Normal Mode, AVDD = 5V, Gain = 64, 128		1350	2500	µA
	Standby Mode		0.1	1	µA
	Power-Down		0.1	1	µA

Part Number	Voltage	Junction Temperature Range	Package	Pb-Free
MIC5209-2.5YS	2.5V	–40°C to +125°C	SOT-223	X
MIC5209-3.0YS	3.0V	–40°C to +125°C	SOT-223	X
MIC5209-3.3YS	3.3V	–40°C to +125°C	SOT-223	X
MIC5209-3.6YS	3.6V	–40°C to +125°C	SOT-223	X
MIC5209-4.2YS	4.2V	–40°C to +125°C	SOT-223	X
MIC5209-5.0YS	5.0V	–40°C to +125°C	SOT-223	X
MIC5209-1.8YM[1]	1.8V	0°C to +125°C	SOIC-8	X
MIC5209-2.5YM	2.5V	–40°C to +125°C	SOIC-8	X
MIC5209-3.0YM	3.0V	–40°C to +125°C	SOIC-8	X
MIC5209-3.3YM	3.3V	–40°C to +125°C	SOIC-8	X
MIC5209-3.6YM	3.6V	–40°C to +125°C	SOIC-8	X
MIC5209-5.0YM	5.0V	–40°C to +125°C	SOIC-8	X
MIC5209YM	Adjustable (2.5V – 15.0V)	–40°C to +125°C	SOIC-8	X
	Adjustable (1.8V – 2.5V)	0°C to +125°C		
MIC5209-1.8YU[1]	1.8V	0°C to +125°C	TO-263-5	X
MIC5209-2.5YU	2.5V	–40°C to +125°C	TO-263-5	X
MIC5209-3.0YU	3.0V	–40°C to +125°C	TO-263-5	X
MIC5209-3.3YU	3.3V	–40°C to +125°C	TO-263-5	X
MIC5209-3.6YU	3.6V	–40°C to +125°C	TO-263-5	X
MIC5209-5.0YU	5.0V	–40°C to +125°C	TO-263-5	X
MIC5209YU	Adjustable (2.5V – 15.0V)	–40°C to +125°C	TO-263-5	X
	Adjustable (1.8V – 2.5V)	0°C to +125°C		
MIC5209YML	Adjustable (2.5V – 15.0V)	–40°C to +125°C	8-Pin DFN	X
	Adjustable (1.8V – 2.5V)	0°C to +125°C		

Note:

1. Contact Micrel for availability.

For communication interface
The ATmega 328 can work on a voltage level in 1.8-5.5V. To minimize the need for additional circuits on your PCB, try to operate it at a voltage level which matches with that of the other devices you will use. In the above example we can choose to operate everything at 5V. However, this may not always be possible and you may have to communicate with devices at different voltage levels.

Here are some ways you can control/read devices at different voltage levels:
- Voltage dividers for unidirectional communications
- Level Shifters for bi-directional communication (http://dlnmh9ip6v2uc.cloudfront.net/datasheets/BreakoutBoards/Logic_Level_Bidirectional.pdf)
- Transistor or MOSFET amplifiers can be uni or bi-directional
- Tons of other ways
- Some ICs designed for level shifting (like TXB0104PWRSOIC14 (digikey 296-21928-1-ND) for bi-directional communication). If you use a level-shifting IC, make sure to ground unused connections.

Power Consumption

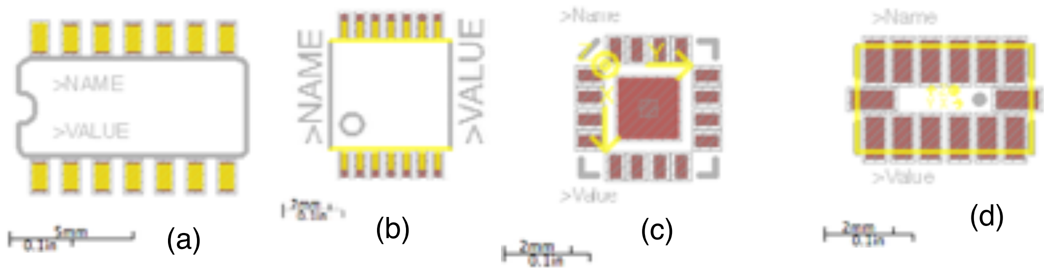
Your voltage regulator can only provide a limited amount of power (0.5A in case of MIC5209). Add the approximate currents consumed by all devices connected to that regulator and make sure they are within this limit (with a margin of at least 10-20%). If the power requirement turns out to be higher than the supply limits, add more voltage regulators and distribute the current loads. Make sure you also add consumption due to LEDs, resistor bridges etc which you may have designed in addition to the ICs & devices.

If the power regulator is stressed, they either shut down (and turn off the rest of your circuit) or blow up taking along a few other ICs with them.

Finally, most voltage regulators you will use are linear regulators. If you had an input supply of 12V and your whole circuit required 100mA at 5V then a linear voltage regulator would need at least 0.1A as input at 12V and it would dissipate at least (12-5)*0.1 = 0.7 W of power as heat (yes, that’s 200mW more than what your circuit consumes). So make sure you have a large enough heat sink area to dissipate whatever heat your regulator might generate. The large pin on the top of linear regulators like MIC5209 is generally used for heat dissipation.

Packages

The eagle library viewer shows you the symbol (left picture) and package (right picture) of the part when you're browsing. What kinds of footprints you use could definitely change how much work it takes to bring up your board. It is easiest to solder thru-hole parts (called DIP), like the ATmega 328 you are using, however the variety of DIP parts available is generally smaller and they also tend to be unnecessarily large. In general, go for the larger SMD components (SOIC, SOT) and avoid anything with a pad that cannot be soldered from the outside. In the symbol, the red polygons are the pads that need solder applied with an iron. You are free to pick anything off digikey, but we would recommend using something already available in the SparkFun library since they have make all the footprints/symbols for us already.



In these examples, (a) and (b) are the same part (level shifter) and (c) and (d) are similar parts (accelerometers).
Package (a) is preferable for our boards since the pitch is much higher, which leaves fewer chances for solder bridges. Both of these parts have leads extending out of the package, which we know singe the red rectangles are outside of the grey rectangle (the part body).
Of the accelerometers, neither package is great for us, but (c) would be almost impossible to solder with an iron. This part would generally need reflow or hot air to be put on a board. (d) is also not great, since the leads and pads don't extend out of the part body, but it is doable. **Our recommendation here is to make your life easier by editing the package to extend the pads away from the part body by 2 or so pad widths.**

Here is an Adafruit tutorial on editing packages in libraries:
https://learn.adafruit.com/eagle-tutorial-how-to-add-a-new-package-to-a-component/package-adjustments

---

Multi-Bit?

We want you to work with something more complicated than a binary input/output. This does not mean you have to use a complicated/custom multi-pin chip. We just want to see more than 1 bit worth of state information transferred. Examples:

- Serial (bytes)
- Analog (10 bit input)
- PWM
- etc.

(A LED/Relay turning ON/OFF won’t be enough)

This is messy :-(How do I go about making my PCB design?

Make a chart like this and figure out what you need on your PCBs:

Property	Sensor	Actuator
Complete Name or Part No.		
Package		
What does it do?		

Supply Voltage Range (Your operation voltage)		
Required Supply Current		
Need additional power circuits or regulators? If yes, which one?		
How does it communicate with the microcontroller?		
Interface voltage levels		
Interface current		
Additional interface circuit required?		

Happy Designing!