Extending the Arch Rock Primer Pack with External Sensors and Actuator Devices

Sensor Networks tie together several distinct, but related worlds:

- The *physical world* of space, things, mechanical actions, positions, forces, heat, motion, gravity, acceleration, light, chemical concentrations, magnetism, radio waves, and so on.
- The *analog world* of electronics, with voltage, current, resistance, capacitance, and inductance.
- The *digital world* of Boolean values (true and false), logic, bits, and numbers.
- The *information world* where we give meaning to logical and numerical values, process and manipulate them, visualize information, draw conclusions, and elect to take actions.

We move between these worlds often in every day life; we flip switches and push buttons to turn on lights and gadgets, we read meters and say "its hot", we turn dials to select particular radio stations, we track packages on the web, we run spreadsheets over all sorts of data, and so on. Arch Rock's wireless sensor network offerings allow you to span these worlds in rich and meaningful ways to better understand, perceive, infer and act.

Your Arch Rock Primer Pack has several high quality environmental sensors built-in to each node. They measure temperature, humidity, and light. In fact, it has two light sensors, one that reads the entire visible spectrum, total solar radiation (TSR), and one that only reads the bands that plants care about, photosynthetically active radiation (PAR).

However, literally thousands of different sensors are available for all sorts of different phenomena using all sorts of different sensing mechanisms and many different interfaces. Many of these can be connected to your Primer Pack nodes using its extender terminals and its suite of sensor drivers. Like the built-in sensors, these extensions will be automatically available to you as embedded web services through your Arch Rock server.

This document provides a brief overview of integrating the physical world and the information world with your own choice of sensors and actuators using the Arch Rock Primer Pack expansion suite. The Arch Rock system, networking, and services make this physical information easily accessible through web services and allow you to program actions based on this information as web services. The extension board on the individual nodes is where we take the analog and digital steps in the larger leap between the physical and information worlds.

Binary Devices

We begin with the simplest classes of devices – those that operate in the binary domain of On/Off, Yes/No, True/False, and Black/White – no shades of gray. A huge class of interesting input and output devices fall into this class and are easily integrated into your embedded wireless web.

Switches – analog binary inputs

Probably the simplest "sensor" in every day life is the switch. Physically, the switch is a mechanical device that has an orientation – up/down, in/out – that we think of an On/Off. Electrically, it an analog device with two configurations – open and closed. Viewed from a slightly different perspective, it is a position sensor that has infinite resistance (open circuit) in one position and zero resistance (closed circuit) in the other. Indeed, many switches are used to detect the physical presence of an object that moves the switch – in addition to the more conventional uses of switches to turn electrical devices on and off.

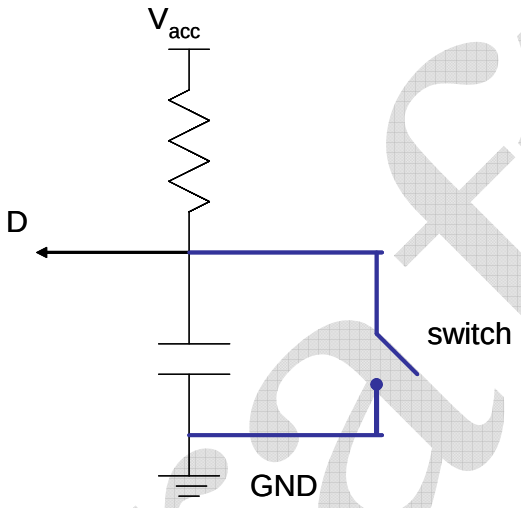

Figure 1. Simple Circuit Converting an Analog Switch into a Binary Digital Signal

The translation from the analog switch (open/closed) to a digital value (true/false) is performed by a very simple circuit, shown in Figure 1, on the Extender. In essence, this is a 1-bit analog-to-digital converter (ADC). When the switch is open, the capacitor charges until it reaches the supply voltage, V_{acc}. At this point, the voltage at D represents a logical 1 or TRUE value. When the switch is closed it creates a path to ground allowing current to flow. The low voltage at D represents a logical 0 or FALSE value. This digital input can be connected directly to a microcontroller through General Purpose IO (GIO) pin, where it can be processed and acted upon. This simple circuit is also a low-pass filter that debounces the switch, so even if the opening and closing of the switch contacts is not perfect, the microcontroller sees a discrete change in the Boolean value.

Although we translated the mechanical switch into a digital input, some analog aspects remain – especially power. When the switch is closed, current flows and power is consumed. For example, if the input voltage from the battery is 3 V and the resistor is 500 K ohms, the current through the resistor and switch is I = V/R = 3/500000 = 6 uA and the power consumed in P = IV =18 uW. While this is a tiny value, it is substantially larger than the power consumption of the TI MSP 430 microcontroller when is sleep state. If our goal was to run for two years on 2 AA batteries, assuming 2,000 mAh battery capacity, with the switch closed all the time it would consume 5.2% of the energy budget.

Thus, it is advantageous that deployments be designed to use "normally open" switches on power-constrained nodes.

Connecting a Switch to the Extender

The Arch Rock Primer Pack Extender provides a user-defined analog switch interface. Simply connect your switch between the SWITCH terminal (6a) and the GND terminal (8b) as indicated in Figure 2.

Although the switch can be sampled at regular intervals, like any other sensor, and the resulting reading communicated over the embedded network. The node can also be configured to generate a notification whenever a switch transition occurs. The change interrupts or awakens the microcontroller, so the Arch Rock embedded system can handle it. The Arch Rock embedded service layer generates a change event which can be communicated over the network. Thus, the switch is physically connected to the extender and logically configured through the web over the mesh network, so its readings and changes can be accessed like other sources of physical information.

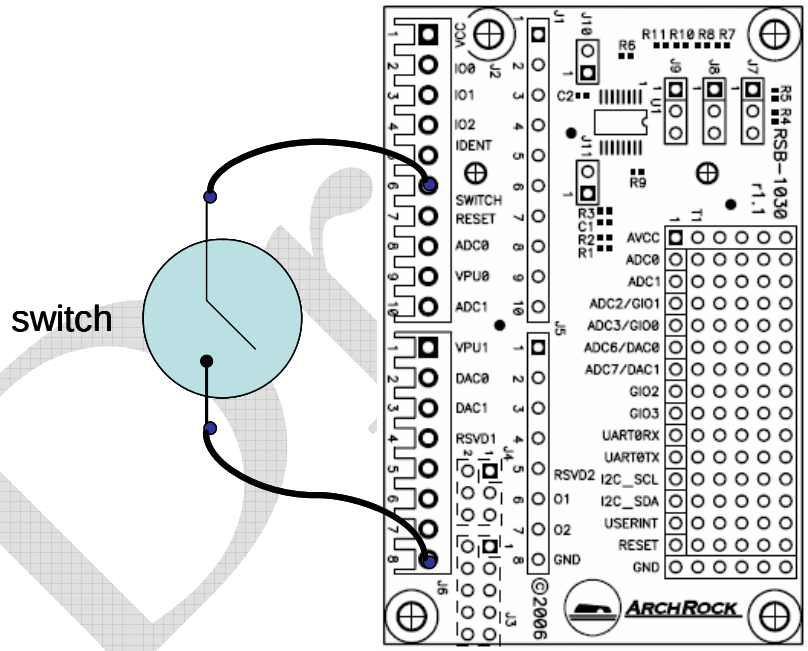

Figure 2. Attaching a Switch to the Arch Rock Extender

Accessing the Switch through the Web Interface

The Arch Rock Primer web services and web-based user interface provide access to these capabilities with no programming required. It is illustrated by Figure 3. Configuring Switch Inputs through the Embedded Web User Interface At the "Sensor/Actuator Devices" page click on the node to which a switch has been attached. It expands to offer a set of device configuration options. Click "Enable External Device" and select "switch". Select port SWITCH. Check the "Periodic Sampling" box, to receiving a reading of the

switch input every sample period (configurable in the box). If you want a reading on each switch transition, check "read on change" box. Often it is natural to select both options, so you have a regular reporting of the switch state, but you also get rapid notification of changes.

The embedded software to support the switch is already in place on the nodes and the server software to make switch events available to you is accessible once you enable such a device on a node. The input value is High when the switch is open and Low when the switch is closed. You may assign a meaningful names to the switch, such as "water tank level," and to its states, such as Full / Empty, using the device configuration page. The data generated by these devices is available on the Data Analysis page, much like all other sensors.

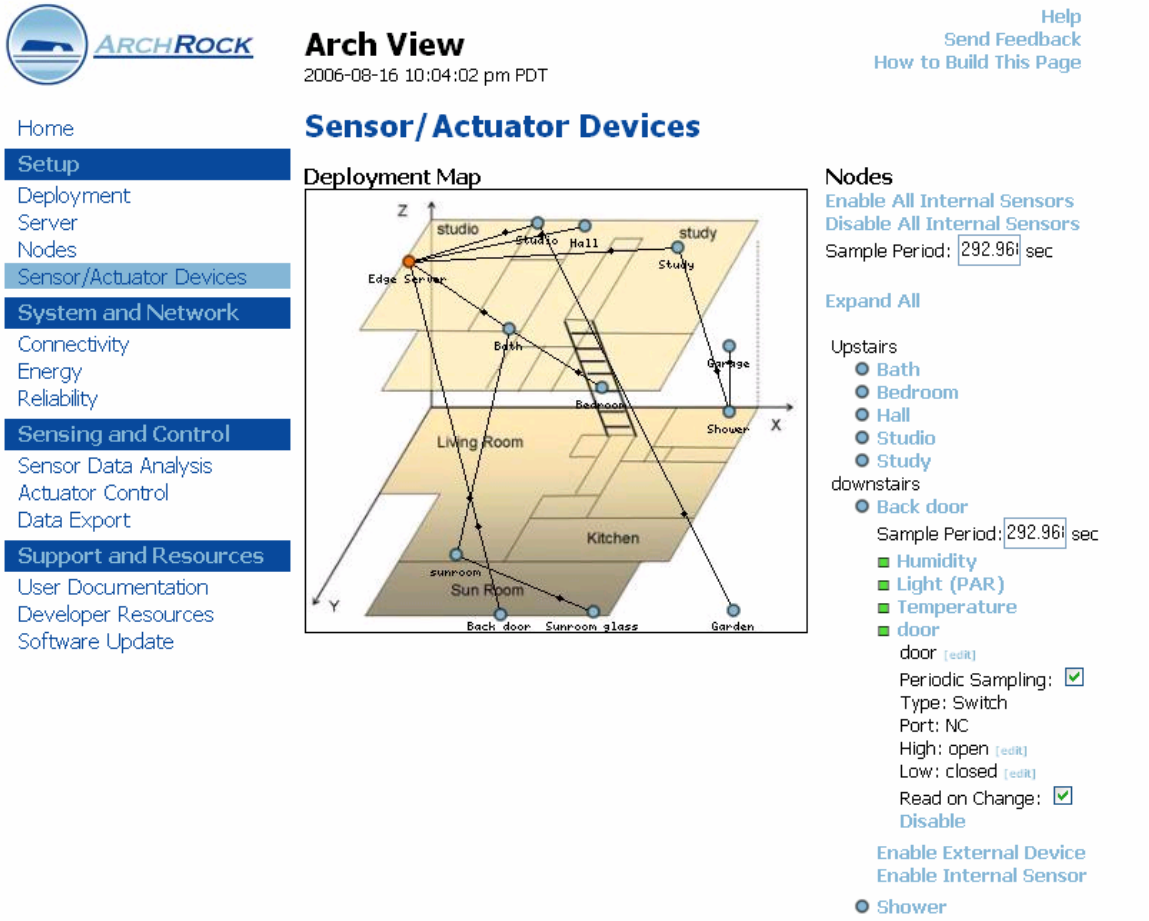

Figure 3. Configuring Switch Inputs through the Embedded Web User Interface

Trip Sensors

We usually think of switches as the mechanical devices that control lights and appliances, but switch-based binary sensors are extremely common. A few examples are illustrated in Figure 4. Many irrigation equipment providers offer rain gauges that open when a configured water level is obtained. Similar devices are used for tank levels, floats and the like. Mechanical vibration and tilt sensors open when a slight tilt causes an internal metal

ball to roll off one of the contacts. The magnetic reed contact sensors used widely in home and building security are magnetically controlled switches. When the magnet attached to the window or door is adjacent to the sensor, the switch is closed; it opens when the magnets are separated. Miniature magnet reed switches are often used on rotating shafts so that a pulse is generated whenever the magnet passes by; such techniques are used for tachometers, flow sensors, weather veins, and many other applications. Pressure, water flow and current trip-sensors cause a circuit to be formed or broken when a certain threshold is obtained.

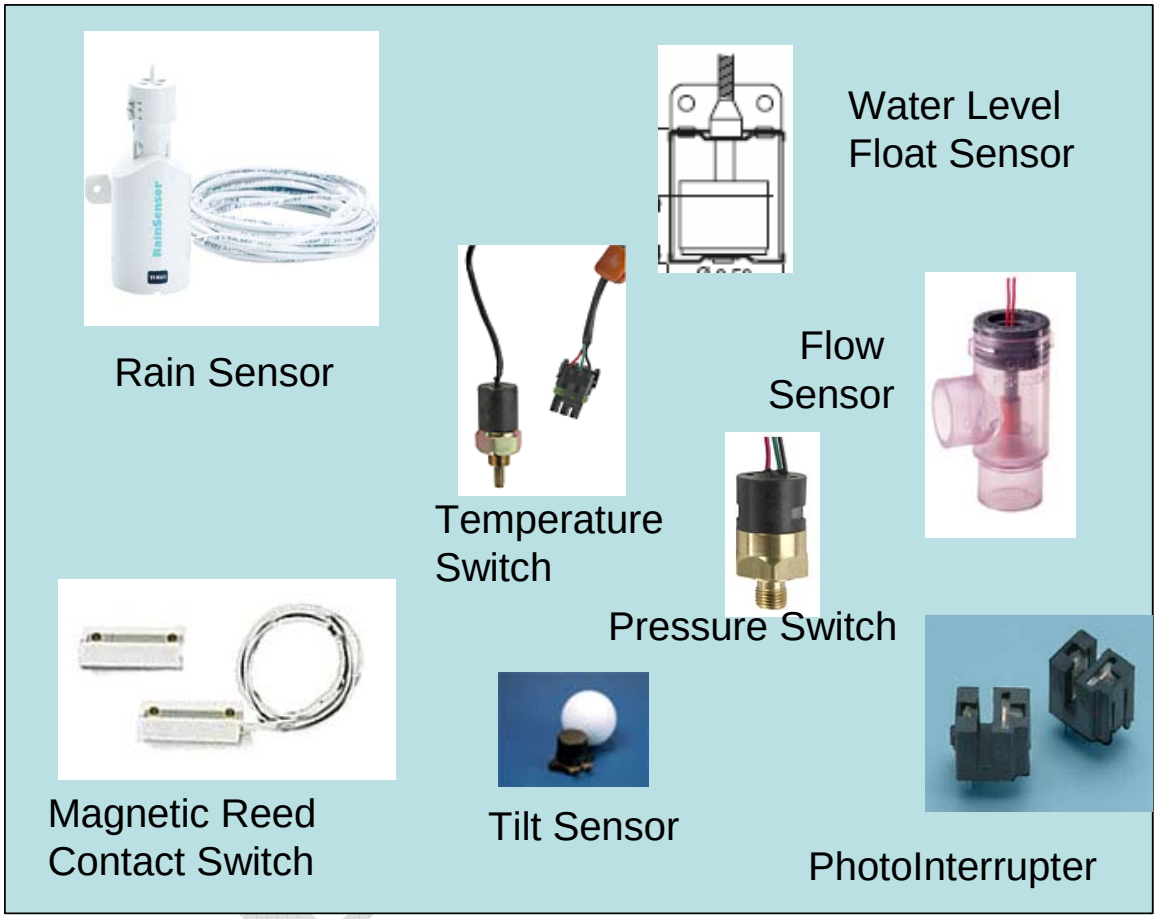

Figure 4 Example Switch-based Sensors

If an application requires connecting multiple switches to an Arch Rock node, three additional general purpose IO terminals are provided on the Extender: IO0, IO1, and IO2. By default, these are intended for digital usage (described below), however, any of them can be adapted for use with analog switches be providing a conversion circuit, such as that in Figure 1. The pull-up resistor is required to raise the input voltage to a logical 1. Space is provided on the Arch Rock Extender for you to place and connect additional electrical components.

Switches as a Web Services

As with every other capability accessible from the Arch Rock Primer Pack web-based user interface, configuration and access of external devices is directly available through

Web Services as well. Web Services exchange information in structured XML over HTTP. They are platform and implementation independent software components that can be *described* using a service description language, p*ublished* to a registry of services, *discovered* through a standard mechanism, i*nvoked* through a declared API, usually over a network, and c*omposed* with other services. In other words, they are the building blocks for constructing sophisticated distributed systems that can evolve as needs and capabilities evolve.

This first step in using a Web Service is to discover the service and determine what operations it provides and how the arguments and results represented. This is all described in XML by a "Web Services Description Language" (WSDL) file. The WSDL for the real-world Web Services provided by an Arch Rock wireless sensor network can be discovered at the network's edge server as follows:

```
http://<server>:5300/V1/?method=gw.getWSDL
```

This odd looking URL is a REST remote procedure call (REST stands for Representation State Transfer, as if that makes anything clearer.) The WSDL file is typically utilized by the Web Services SOAP client to generate a binding to the Web Services server, however, it is also useful to see the suite of available web-based capabilities. All of these are available in either REST or SOAP. You might want to type it into your browser and look at the result.

As an example REST call, the most recent value (or *data event*) from each of the enabled switches can be obtained by typing the following URL directly into your browser:

```
http://<server>:5300/V1/?method=events.readLast&name=SwitchReadEvent
```

The REST call returns an XML object, such as the following, containing time-stamped, typed values for each of the devices with an enabled external switch input.

```
<Results>
 <Result addr="00173b000fed2d51" timestamp="1155792111.022744" seqNo="3822"
name="SwitchReadEvent">
 <Value typeName="nx_bool">0</Value>
 </Result>
 <Result addr="00173b000fed265e" timestamp="1155792103.733275" seqNo="3775"
name="SwitchReadEvent">
  <Value typeName="nx_bool">1</Value>
 </Result>
</Results>
```

To access this information programmatically, you could pass the URL to a utility (eg., curl –u user:passwd "URL") or library routine used for accessing web objects.

In addition to the data event, each switch has configuration and value *attributes* that can be "set" or "get". The following REST call, goes out to each node over the embedded network, obtains the current value of SWITCH and

collects all of values into an XML object. It will obtain the value associated with the SWITCH port, whether that port is enabled on a node or not.

```
http://<server>:5300/V1/?method=attributes.get&name=SwitchValue
```

If you were interested in the value at a particular node, you would add the argument &addr = "EUID" for the node of interest. For example,

http://<server>:5300/V1/?method=attributes.get&&addr=00173b000fed265e&name=SwitchValue

The configuration of the device, which you can set though the External Devices page, is a structured attribute that can be accessed directly as a Web Service. For example,

http://<server>:5300/V1/?method=attributes.get&name=SwitchConfig
returns an object like the following.

```
<?xml version="1.0" encoding="utf-8" ?>
- <Results xmlns="urn:gw:api">
- <Result addr="00173b000fed265e"
    timestamp="1155851249.468720" name="SwitchConfig">
- <Value typeName="nx_pin_config_t">
  <enabled>1</enabled>
  <sample>0</sample>
  <output>0</output>
  <interrupt>1</interrupt>
    </Value>
    </Result>
- <Result addr="00173b000e3304f9"
    timestamp="1155851249.468900" name="SwitchConfig">
- <Value typeName="nx_pin_config_t">
  <enabled>0</enabled>
  <sample>0</sample>
  <output>1</output>
  <interrupt>0</interrupt>
    </Value>
    </Result>
 -
 ...
 </Results>
```

Here the first node has its switch enabled as an input and provides a notification on each change, whereas the second one does not have the switch enabled. (By default, microcontroller pins are configured as outputs to minimize power consumption.)

To configure the first node to report its switch value at regular intervals, in addition to when it changes, the following call could be used.

http://<server>:5300/V1/?method=attributes.set&addr=00173b000fed265e&name=SwitchConfig&value.enabled=1&value.sample=1&value.output=0&value.interrupt=1

These capabilities are available through the more sophisticated SOAP-based Web Service facilities in such languages as Java, C#, Perl, Python, C and C##. The use of SOAP is more fully described in the Arch Rock SOAP guide, but a taste of it follows. In Java, a Web Service "client" generation tool, such as WSDL2Java, is used to construct a Java class that represents the Web Service and interacts with it over XML-based remote procedure calls. That class binds to a particular Web Service server when is runs. For example, we might gain access to the Arch Rock edge server SOAP web services using the following, where all the GW associated methods are auto-generated from the WSDL file.

```
URL portURL = new URL("http://" + gateway + "/soap");
GWServiceLocator locator = new GWServiceLocator();
proxy = locator.getGWPort(portURL);
```

Then the most recent switch reading on all nodes looks like a familiar method invocation

GW__eventsRead_Result = proxy.eventsReadLast("SwitchReadEvent", "FFFFFFFFFFFFFFFF");

Similarly, all the other REST capabilities are available through the SOAP interface. Modern Integrated Development Environments, such as Eclipse, Visual Studio, and NetBeans, provide extensive support for programming on SOAP-based Web Services.

Logic gates – digital binary inputs

Many kinds of electronic devices can function as switch-based sensors, generating a logical output of Zero or One when some particular status is obtained or some event detected. These digital signals can be connected directly to GIO pins of the microcontroller using the IO0, IO1, IO2 terminals at the upper part of the terminal block shown in Figure 2, along with their GND, and configuring those ports as inputs.

The Arch Rock Primer web services and web-based user interface also provide access to these capabilities without user programming. At the "Sensor/Actuator Devices" page, click on the node that has been augmented with the external binary input. Click "Enable External Device" and select switch. Select the IO port to which you have connected the binary signal. Once you have configured the port as a switch, the server will instruct the individual node to utilize that port as an input. In addition, for port IO0 you can request that notification be generated on all changes. When used as inputs, the sampling and event generation of these devices can be configured on the Device Configuration page and the resulting data can be accessed through the Data Analysis page.

Just as with analog switches, all the capabilities presented in the web-based user interface are available to you as web services. For example, the following REST call could be used to get the most recent readings from IO1 port.

```
http://<server>:5300/V1/?method=events.readLast&name=IO1ReadEvent
```

```
The corresponding SOAP call in JAVA would be
```

GW__eventsRead_Result = proxy.eventsReadLast("IO1ReadEvent", "FFFFFFFFFFFFFFFF");

To obtain the configuration of all the IO1 ports, you could use the following

http://<server>:5300/V1/?method=attributes.get&name=IO1Config

Whereas to actively get the value on this input, the following REST call could be used

http://<server>:5300/V1/?method=attributes.get&name=IO1Value

With digital inputs, one analog issue that often arises is the voltage level associated with a logical high value. With the TI MSP running with Vcc = 3 V, its positive-going input threshold voltage is 2.0 volts and its negative going input threshold is 1.3 volts. These are typical CMOS levels. TTL interfaces typically operate at 0 - 5V. Thus, a level conversion should be applied to TTL inputs to bring it down to CMOS levels. For inputs, this can be accomplished by a simple voltage divider. More generally, a level conversion chip should be used.

Relays and Set Points - Binary outputs

The IO ports can also be configured as outputs to control and actuate external devices. The classical use of this is a relay, in which an electronic device causes a analog switch to open and close. Of course, binary outputs are also used to light LEDs and perform other functions.

The Arch Rock Primer web services and web-based user interface provide access to these capabilities at the Sensor/Actuator Devices page selecting the Device Type to be Relay and selecting the port to which the relay or actuation device is attached. The value transmitted to this device can then be set using the Actuator Control page.

These capabilities are available through web services as well. For example, the following REST call would set the value output on port IO2 to be High.

http://<server>:5300/V1/?method=attributes.set&&addr="00173b000fed2d51"&name=IO2Value&value=1

The analog precaution that must be observed is the current drawn by the external device. No device should draw more than 6 mA from a pin and the total current drawn from all microcontroller outputs combined should not exceed 48 mA. This is particularly an issue when the external device is analog, such as an LED. The effective resistance should be large enough that the current draw through the output pin is within tolerance. For binary devices, current draw is seldom a problem, but attention must be paid to voltage level. The TI MSP microcontroller operates at CMOS levels, below 3.3 volts. Many digital devices operate at TTL levels 0/5 V. To connect the extender output to a TTL input will typically require a level conversion.

As a little exercise, how might we support a switched sensor which consumed very little power, even though it was normally closed? Say we have a float switch based water level detector in a tank and we only need to know if the level is below threshold at 10 minute intervals. We can construct our simple pull-up circuit as in Figure 1, but instead

of using V_{acc} as the voltage source, we use an IO port configured as an output, say IO2. Asserting a logical 1 on the output port enables the sensor, otherwise it reads as low and draws no power. We might enable the sensor for a few seconds every few minutes to take the reading.

Range-Valued Devices

Many sensors and actuators utilize a range of values, rather than binary options – Open/Close, On/Off, True/False. They make use of the shades of gray. The space of possible sensors is immense and they have widely varying interfaces. The Arch Rock Primer extender support the common, relatively basic cases directly. More specialized and more sophisticated forms of use require additional external circuitry.

Resistive Sensors

One important class of range-values is resistive sensors, which includes thermistors, proto-resistors, strain gauges, flex strips, and some thermocouples. Electrically, these devices are passive circuit elements with a variable resistance that changes in response to physical conditions. For example, Figure 5 shows how the resistance of a particular thermistor (Panasonic ERTJ 4500K) decreases with increasing temperature. At 0 °C, it has a resistance of 18 MΩ, at 25 °C it obtains its rated value of 4500 KΩ, whereas at 100 °C it has 220 KΩ. The datasheets for the particular sensor give the conversion functions in detail. To obtain the reading in physical units, we must observe the resistance and then compute the corresponding temperature, or whatever physical phenomenon is being sensed.

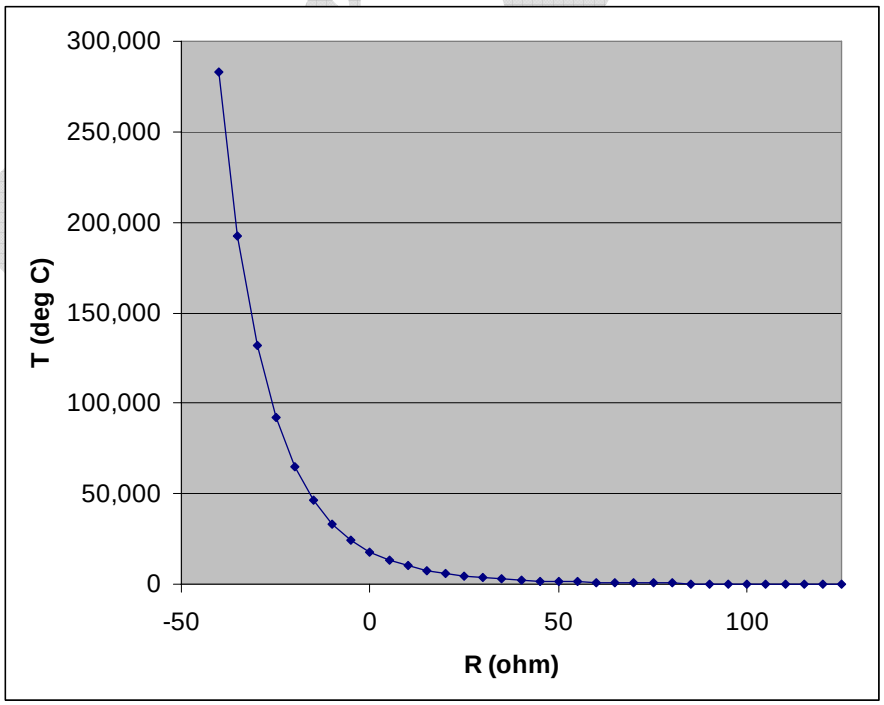

Figure 5 Characteristic Function of a Thermistor showing Resistance as a function of Temperature

While a switch has either infinite resistance (open) or zero resistance (closed) these devices take on a range of resistances in between. By placing them in a voltage

divider circuit, as illustrated in Figure 6, this variation in resistance is converted into variation in voltage, which can be measured by an ADC to get a digital reading for voltages between zero (V=GND) and D_{max} ($V = V_{ref}$).

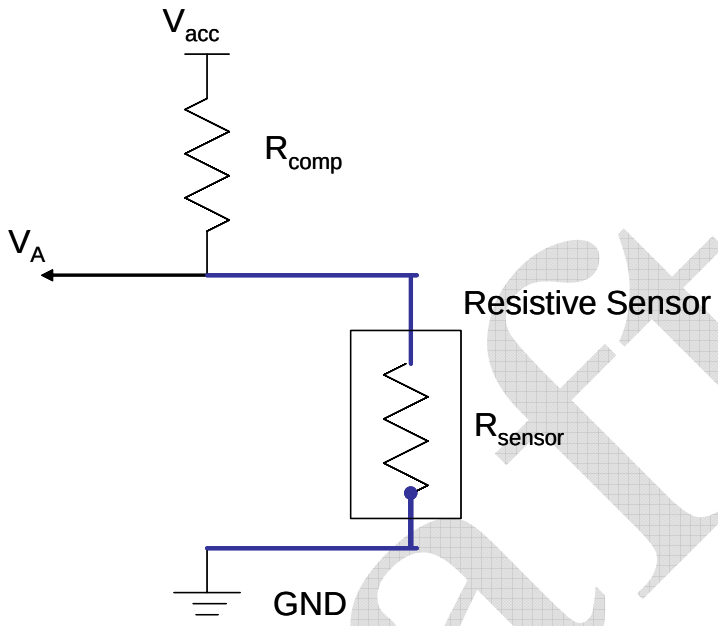

Figure 6 Conversion of a Resistive Sensor to an Analog Voltage to a Digital Value

The voltage, V_A, at point A in Figure 6 is equal to

$$V_A = V_{acc} * R_{sensor} / (R_{sensor} + R_{comp}),$$

where V_{acc} is the analog supply voltage, R_{comp} is the resistance of the pull-up, and R_{sensor} is the effective resistance of the sensor.

This analog input, V_A,is sampled by the ADC in the microcontroller and compared to a reference voltage, V_{ref}, to produce a digital value D between zero and D_{max} such that

$$D = \min(D_{max} , D_{max} * V_A / V_{ref}).$$

In the Arch Rock Primer Pack, a 12-bit ADC is provided by the TI MSP microcontroller, so $D_{max} = 4095$ (or 0xFFF). The reference voltage, V_{ref}, can be set to 1.5 V, 2.5 V, or Vcc.

In many situations a constant, regulated voltage is most useful as a reference. However, for simple resistive sensors it is convenient to have the reference voltage be the same as the supply voltage to the sensor, $V_{ref} = V_{acc} = V_{vcc,}$ even when this is an unregulated battery voltage. This configuration is called a *ratiometric* sensor; all we need to determine the physical condition being sensed is the ratio of the resistances. For example, when $R_{sensor} = R_{comp,}$ we will get a reading of $D = D_{max} / 2$, independent of the supply voltage. Using the lookup function from the sensor datasheet, we can convert this to physical units.

In general, for a ratiometric sensor we will get a reading of

$$D = D_{max} * R_{sensor} / (R_{sensor} + R_{comp}).$$

Solving for R_{sensor} we get

$$R_{sensor} = R_{comp} * \rho / (1- \rho), \text{ where } \rho = D/ D_{max}.$$

In practice, with a ratiometric sensor, you cannot use the full range of the ADC, because the output voltage will be significantly less than the reference voltage. If a small R_{comp} were used so that the output voltage would be close to the reference when the sensor resistance is large, then at the other range of the scale both resistances would be small and too much current would be drawn through the voltage divider. If R_{comp} is chosen to be roughly equal to the effective sensor resistance at the high end its useful operating regime, half of the ADC range is used. (The most significant bit will be zero.) A smaller R_{comp} can be used, increasing the range of readings, as long as the current does not exceed the microcontroller tolerances and power consumption fits within the application power budget.

Alternatively, if a fixed voltage reference is used for the ADC, we can determine V_A from the observed input D as

$$V_A = V_{ref} * D / D_{max},$$

as long as $V_A < V_{ref}$. V_{ref} will typically be smaller than the supply V_{acc}, so in this case, the value of R_{comp} should chosen be so that at the end of the desired sensing range with the largest resistance, $V_A \leq V_{ref}$.

The effective resistance of the sensor is calculated as

$$R_{sensor} = R_{comp} * \alpha / (1 - \alpha), \text{ where } \alpha = V_A / V_{acc}.$$

Your Arch Rock Primer Pack extender provides connections and preprogrammed support for two external ADC channels. A circuit like that described in Figure 6 is constructed by connecting the resistive sensor between ADC*i* and GND. In addition, connect together the VPU*i* and ADC*i* terminals, as illustrated in Figure 7.

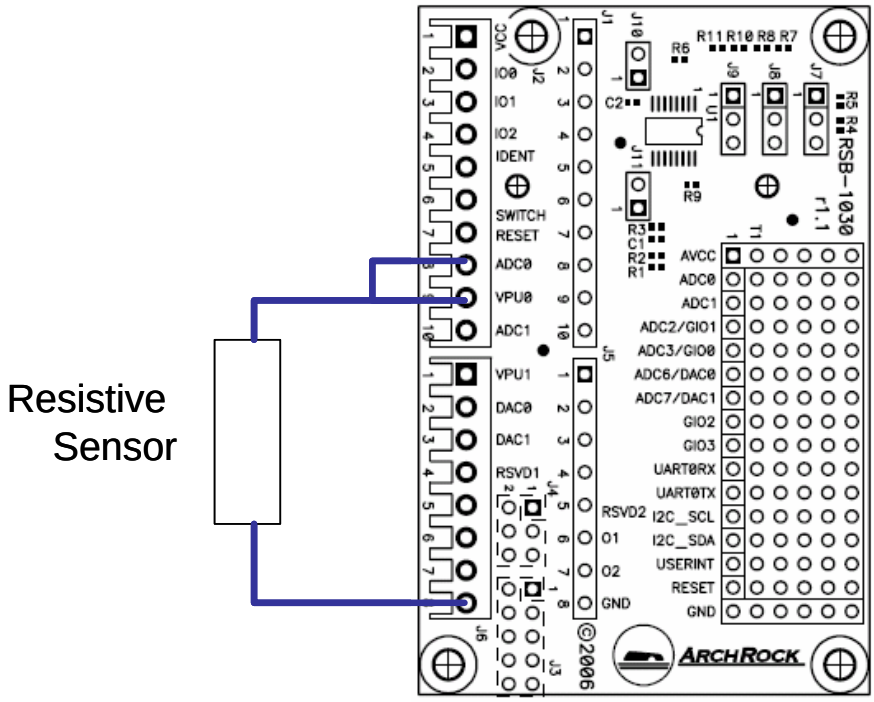

Figure 7 Extender Connections for a Resistive Sensor on ADC Channel 0

The extender contains a digital potentiometer so that R_{comp} can be set under program control for each channel to appropriately match the sensor. For example, a typical configuration for a fixed voltage reference might be $R_{comp} = R_{sensor\_max} (V_{Acc} / V_{ref} - 1)$. Thus, if $V_{Acc} = 3.3$ V, $V_{ref} = 1.5$ V, and $R_{sensor\_max} = 10{,}000$ ohm, we would want to set the digital potentiometer so that $R_{comp} > 12{,}000$ ohms. This would use a setting of $POT_{ctrl} = 256 * R_{comp} / R_{max} = 256 * 12{,}000 / 100{,}000 \approx 31$.

The Arch Rock Primer web services and web-based user interface also provides access to external sensors through the built-in ADC without user programming. On the Sensor/Actuator Devices page, click on the node to which you have attached the external sensor. It expands to show you the ports that are currently configured. Under Device Type, select Sensor. Select the appropriate ADC port (ADC0 or ADC1) and click Enable. This will provide additional configuration detail for the ADC port. You can enter the desired resistance for the pull up; the software will determine the closest setting for the digital potentiometer on that channel. Select the desired reference voltage.

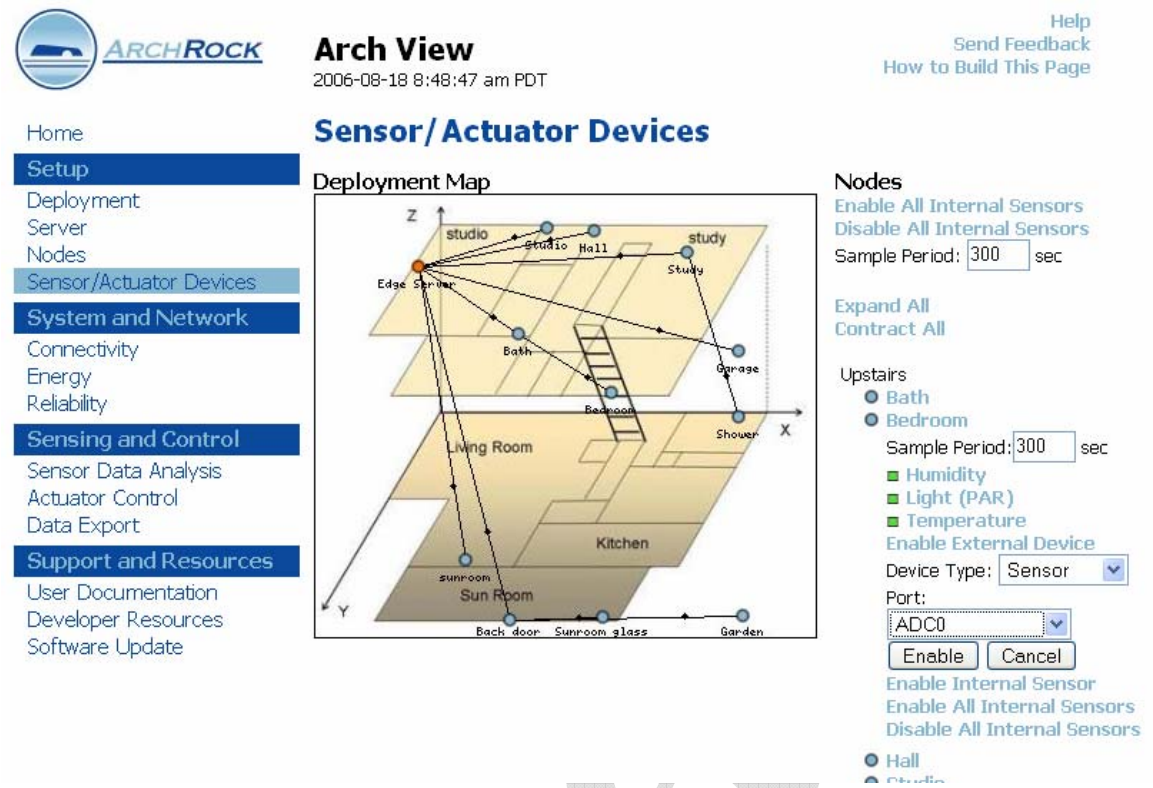

Figure 8 Configuring External Sensors through the Web Interface

Voltage is supplied to the external sensor from an IO pin through the pull-up resistor. You can select when you want this provided – all the time, just in advance of sampling the sensor, or for a warm-up period before the sample. Some sensors require such a warm up in order for the reading to stabilize.

Finally, you can select whether you want the sensor sample at regular intervals, or on demand when a sample is requested.

The REST and SOAP calls to get and set the ADC0Config and ADC1Config attributes allow these configurations to be queried or established programmatically. The data can be obtained as with any of the built-in sensors using the ADC0ReadEvent and ADC1ReadEvent.

Our thermistor example raises a useful tidbit. Although thermistors are resistors that are designed specifically to vary their resistance with temperature, in practice all resistors are temperature dependent to some degree. In fact, many different kinds of sensors exhibit some temperature dependency. The datasheets will describe how to do the temperature compensation in the conversion to physical units. Some sensors will provide built-in temperature compensation – especially high quality digital sensors. In many real applications, the data analysis will utilize multiple sensor nodes to gain high quality physical information.

Voltage Source Sensors

Another important class of sensors is those that are essentially voltage sources. Consider for a moment a generator – when the armature spins it generates current. Tachometers, speedometers, anemometers (wind speed meter), and many other devices are similar. The sensor need not have moving parts to be a voltage source; it may have its own independent source of power. In any case, it becomes an active element in the circuit.

For these sensors, when the voltage source peaks at CMOS levels, the connection to the Arch Rock Primer Pack is even simpler. The digital potentiometer is disengaged, so that VPUi simply indicates that the sensor is being sampled (or it can be a source of power to the sensor). The reference voltage will normally be set to a regulated constant. The output of the sensor is connected to ADCi. No connection is made between VPUi and ADCi.

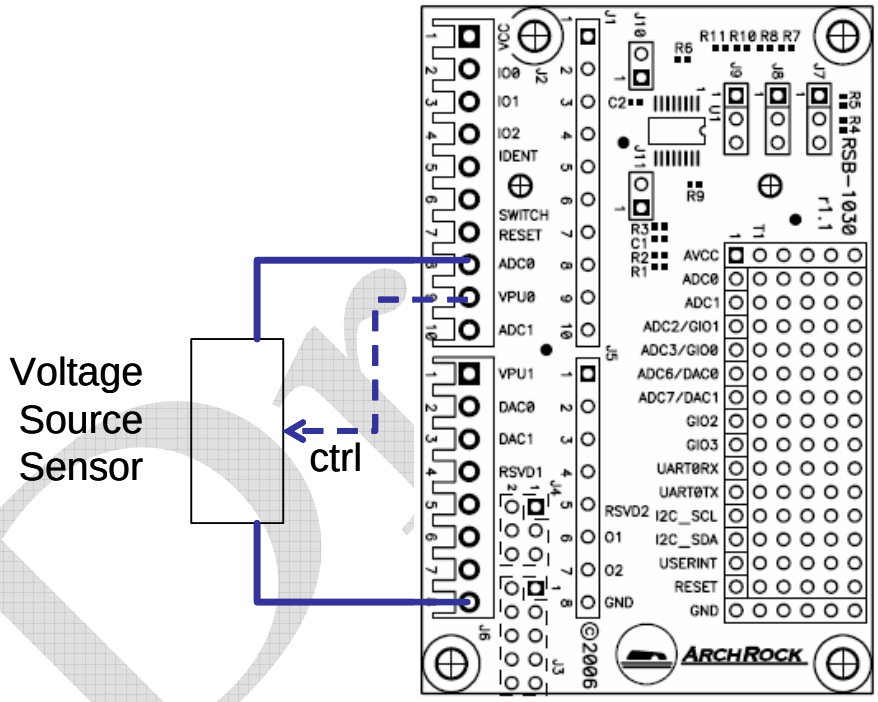

Figure 9 Extender Connection for Voltage Source Sensor

Although the wiring on the Extender differs from resistive sensors, the web service configuration is the same.

The main complication that arises in this case is when the maximum useful output voltage does not match well enough with the available references on the MSP. Some external sensors have very small voltage swings – millivolts in various cases. In this case, operational amplifiers and other signal condition is used to match to the reference.

In some cases, the output voltage is higher than the available ADC reference voltages, say 5V or 12V so a level conversion is required. The digital potentiometer on the Primer Pack Extend can be used as a programmable voltage divider as shown in Figure 10. Each

ADC channel has a jumper (J10 for ADC0 and J11 for ADC1) that converts the variable resistor into a voltage divider. The sensor signal output is connected to VPUi, rather than ADCi. The resistance setting, R_{comp}, is the upper portion of the voltage divider. The lower portion is 100 KΩ - R_{comp}.

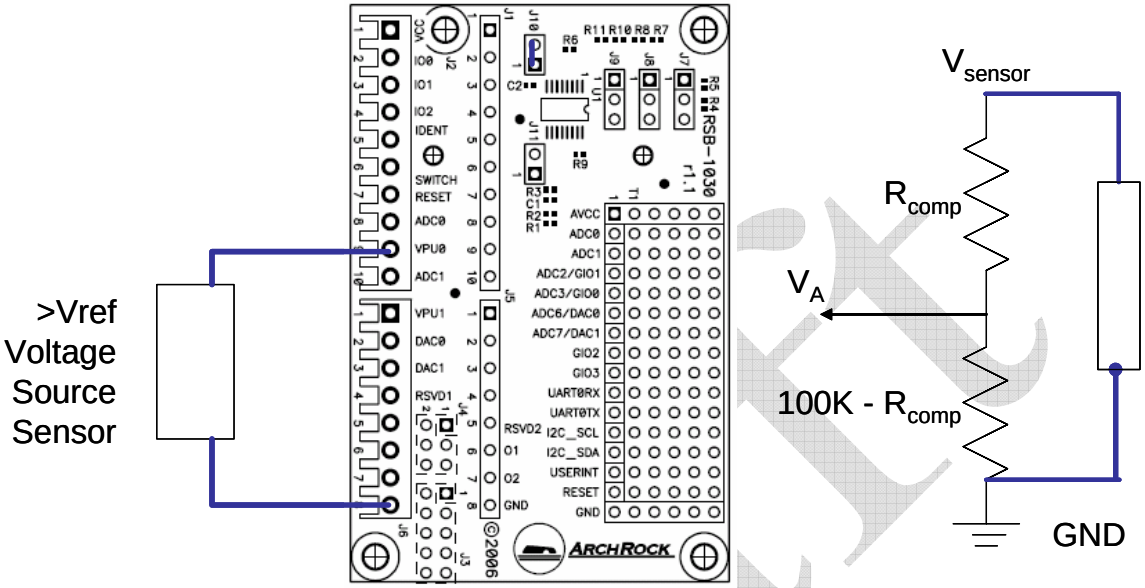

Figure 10 Connection and Circuit for External Sensors Signalling at Greater than Vref

General Analog Sensors

With the thousands of different kinds of sensors that exist and the many interfaces, no fixed set of interfaces can cover them all without external circuitry. For example, many moisture sensor, humidity sensors, and touch sensors are capacitive, rather than resistive. The changing physical conditions change the capacitance of the device. Additional circuitry transforms this change into a change in voltage, which can be measured with the ADC. Many industrial sensors use "current loops" with 4-20 mA signaling where the sensor modulates the flow of current, rather than the resistance or the voltage. Putting a variable current across a fixed resistor creates a variable voltage, which can be provided to an ADC. These kinds of devices often require a specific excitation voltage, they use various signaling voltages, and some provide differential outputs. External signal conditioning circuit should adapt these inputs to a form similar to voltage-source sensors, described above. These sensors can be connected to the Extender as in Figure 9 or Figure 10, as appropriate, through additional circuitry on an external board or on the prototyping area. Typically, they will require a separate voltage supply. Additional IO ports may be used to configure or control these devices.

General Digital Sensors

Increasingly, many sensors incorporate the analog-to-digital conversion internally or directly produce a digital output. These have a range of interfaces, mostly serial, including RS232, UART, I2C, SPI, and 1-wire. The internal humidity and temperature sensors on the Primer Pack nodes are such digital sensor, using I2C. The Telos-based

version of the Arch Rock Primer pack does not support external serial external devices in general.