SYNTHIAM

synthiam.com

Using ADC (Analog to Digital) Ports, Commands and Controls.

This tutorial will cover some basics on using the ADC (Analog to Digital Converter) ports on the EZ-B v4. Included will be some information about the ADC ports, mention some of the peripherals that can be used with them, the EZ-Builder ADC controls you can use, and some script examples for you to look at and try.

Last Updated: 3/23/2016

Step 1. About the ADC ports.

There are 8 ADC Ports on the EZ-B v4 numbered "A0" to "A7" that you can connect devices like analog sensors. Each ADC port contains three pins...
Ground: (output) located towards the centre of the board.
VDC: (output) The centre pin, regulated at 3.3 volts.
Signal: (input) The outermost pin from the centre of the board which reads the voltage between 0 and 3.3 volts.
The ADC ports are Configurable at 8-bit (0-255) or 12-bit (0-4095) in software.
In an ADC control found in the EZ-Builder software, such as "**Read ADC**", ADC reading intervals can be selected from 100 ms to 60000 millisecond analog values.
Analog readings are from 1 to 255 (0v to 3.3v) in 8 bit mode, or 0 to 4,095 in 12 bit mode.

To make sure you get reliable ADC readings using the EZ-B connected to the EZ-Builder software, the following list contains some popular troubleshooting techniques...
1) Make sure that you are supplying the correct power, to power any sensor that you want read (Remembering that the power output from an ADC VDC port is 3.3v),
2) That they are connected to the ADC ports correctly. Keep in mind that the layout of pins on some sensors may not match the same order on the EZ/B, so using an extension cable may not be adequate, and single jumper cables my need to be used.
3) The correct port is selected in the ADC control configuration menu,
4) Make sure that the sensors and cables are in good condition,
5) Adjust the "interval ms" in an ADC control (explained in the next step), to suit you and your sensors needs,
6) Use appropriate "Pull up" or "Pull down" resistors on your sensors where necessary,
7) Adjust any potentiometers that may be on the sensor itself, to dial in any sensitivity that helps get correct readings.
8) Slow ADC readings could be a result of network saturation, wireless interference, or could be due to your computers processor not working as fast as it should.
Something to keep in mind, is the more sensors you use simultaneously, and have the ADC reads set as maximum speed, this will use more processing power and could flood the WiFi network which will slow down response times.
The next step will explain the three ADC controls found in EZ Builder.

Step 2. EZ-Builder ADC controls.

With you EZ-Builder project open on your computer, clicking on "**Project**", "**Add controls**", then "**ADC**" from the menu ribbon, you will be presented with three ADC controls to add to your project.

ADC Graph.

This control will show the ADC voltage in a historical graph. The input voltage can be between 0 volts and 3.3v. A demonstration of this control can be found in step 6.
ADC Graph Configuration Menu.

Name: Where you can rename your "ADC Graph" control.
Board #: Select board number your EZ-B (found in the "Connection Control" configuration menu).
Port: What EZ-B analog port you wish to use.
Interval: Select the amount of time in milliseconds between sensor reads (100 being the quickest, and 60000 being the slowest).
Colour: Select the colour of the graph lines displayed in the control window. (Current choices are Red, Green, or Blue).

ADC Meter.

Displays the ADC (Analog Digital Converter) voltage in a VU (volume unit) meter. The voltage can be between 0 and 3.3v. A demonstration of this control can be found in step 7.
ADC Meter Configuration Menu.

Name: Where you can rename your "ADC Meter" control.
Board #: Select board number your EZ-B (found in the "Connection Control" configuration menu).
Port: What EZ-B analog port you wish to use.
Interval: Select the amount of time in milliseconds between sensor reads. Settings are 100 being the quickest for fast sensor detection (more processing power), and 60000 being the slowest for slower sensor readings (uses less processing power).
Colour: Select the colour of the VU meter displayed in the control window. (Current choices are Red, Green, or Blue).

ADC Value.

This ADC control will display a numeric value and voltage from the selected ADC port chosen that is chosen from the configuration menu. A demonstration of this control can be found in step 5.
ADC Value Configuration Menu.

Name: Where you can rename your "ADC value" control.
Board #: Select board number your EZ-B (found in the "Connection Control" configuration menu).
Port: What EZ-B analog port you wish to use.
Unit Description: Change the name of the unit description displayed in the "ADC Value" control window.
Unit Multiplier: This is used by the value of the ADC reading to calculate the desired scale. If you are using an EZ-B v3, and wish to display voltage, the multiplier value would be 0.0196. If using an EZ-B v4, the multiplier value would be 0.0128.

Interval: Select the amount of time in milliseconds between sensor reads. Settings are 100 being the quickest for fast sensor detection (more processing power), and 60000 being the slowest for slower sensor readings (uses less processing power).

Step 3. ADC peripherals.

A number of sensors can be connected to an ADC port. Some sensors rated between 3.3v and 5v will work straight off the ports, and some will not. Possible sensors you can use are...

Smoke sensor modules

Can be used to detect smoke and other gasses, such as methane and butane gas.

Sound sensors

Adding these to a robot can give it the ability to hear what direction sound is coming from. (See step 7 for an example).

Pressure sensors

These sensors allow you to detect physical pressure, and can be used on things such as a robot claw to detect pressure and stop the claw servo from moving. (Pull up or Pull down resistors mat be required. See the Flex Sensor Tutorial for details).

Passive Infrared (PIR) sensors

These little sensors measure infrared light, and can be used for motion detection. Some PIR sensor modules have the option to work on analog ports, or on digital ports as well.

Gas detection sensor modules

Similar the the smoke detection modules mentioned above, these do exactly what their name says... measures a range of different gasses.

Flex/bend sensors

Much like the pressure sensors mentioned above, these can measure resistance and increase analog readings the further they are bent. What a look at the Flex Sensor Tutorial for more details and a video example.

Sharp infrared range sensors

These can be used for object avoidance, much like the ultrasonic detection sensors, but use infrared. There are "Sharp Infrared Sensor" controls that can be found in EZ-Builder.

Temperature sensors

Use these to measure ambient temperature levels.

Photosensitive diode sensors

These detect "Light" and can be used for many applications. One example is to have one of these plugged in to your EZ-B, along with a low level spotlight. When the sensor levels change when in a dark room, this will trigger the spotlight to come on. (You can adjust the ADC values so the light from the spotlight doesn't cause the sensor to turn it off again). And there are many more sensors you can use. Some sensors that are 5v can be used on

the 3.3v analog pins without any additional hardware or resistors, but others will require some additional help to work properly. Please bare in mind though, that not all sensors available will work straight out of the box on an analog port. Also, research the sensors you wish to use on an analog port, as some are digital or serial. If you're unsure on anything, please feel free to ask in the Community Forum, and where possible, post a link, quote, or picture of the relevant data sheets for the sensor your asking about.

Step 4. ADC Script Commands.

In the EZ-Builder script menu, you will find some script commands that can be used with the ADC ports.

ADC_Wait (ADC Port number, higher/lower/equals, ADC value, [delay ms])

This will wait until an ADC ports value in your script is either "higher" or "lower" than specified value you have chosen. The "Delay MS" command is the millisecond delay for checking and works out the delay between value reads. This command is optional.

EZ-Script menu examples using ADC port "A0".

ADC_Wait(ADC0, HIGHER, 50)

ADC_Wait(ADC0, HIGHER, 50, 50)

ADC_Wait_Between (ADC Port number, low, high, [delay ms])

This command will pause a script until the ADC port reading is in between two values (a high value, and a low value). Once met, the script will begin to run. As before, the "Delay MS" command is the millisecond delay for checking and works out the delay between value reads. This is also is optional.

EZ-Script menu examples using ADC port "A0".

ADC_Wait_Between(ADC0, 20, 50)

ADC_Wait_Between(ADC0, 20, 50, 50)

GetADC(Port)

This will return the 8 Bit ADC value of the ADC port you have chosen.

EZ-Script menu example using ADC port "A0".

$x = GetADC(adc0)

Another example of this script command can be found being used, in step 8.

GetADC12(Port)

Returns the 12 Bit ADC value of the specified port on the EZ-B v4

EZ-Script menu example using ADC port "A0".

Example: $x = GetADC12(adc0)

Step 5. Step 4. Controlling servos with sensors and ADC ports.

One possible use for ADC ports, is to utilize them to control servos connected to digital ports respectively, using pressure or light sensors. In the following example, a servo and Photosensitive sensor will be used. The servo will move depending on how much light the sensor is detecting. This will also show an "ADC Value" control in action. It is not necessary to use this control for this example, it is there for demonstration purposes.
So, here is an example of how to do it...
1.) Open your copy of EZ-Builder, and click "**Project**".

2.) Click on "**Add Controls**", then "**Scripting**" and select "**EZ-Script**".

3.) Now click on the "Gear" icon to open the script editor.

4.) Copy/paste or wright in the following script...
Servo(d0, getadc(adc0))
goto(loop)[/code]
6.) Now click "**Save**".
7.) Next, Connect the sensor to an analog port on your EZ-B, then a servo to a digital port.

8.) Power on your EZ-B, and connect it to EZ-Builder. Now press "**Start**" on the script control, and the servo will now move when the sensor detects different brightness of light which change the analog readings. As an optional extra, you can add an ADC control like the "ADC Value" control seen in the video below, to see the sensor readings change as they happen. Check out the short video demo of this in action...
One thing to note, because most servo positions only go up to 180, and ADC values go up to 255, you may need to add a multiplier to the **GetADC()** script command for each of the positions for better accuracy. The resister values on the Photosensitive sensor determine the accuracy and voltage being returned. An explanation of a multiplier can be found in the step 7.

Step 6. ADC Example with Smoke Sensor.

This is an example using a smoke detection sensor module and an **ADC Graph** control, and a script that responds to specific ADC readings.
1.) Connect the smoke sensor module to an analog port on the EZ-B (This example will use "A0"). Then power up the EZ-B.
2.) Open EZ-Builder, then from the menu ribbon, click "**Project**", "**Add controls**", "**ADC**", then "**ADC Graph**" to add this to your project. Do the same, and add an "**ADC Value**" control as well.

3.) On the "ADC Graph" and "Value" controls, click on the gear icon to open the configuration menu, and select the port the sensor is connected to, then click "**Save**".

4.) Now use a source to make smoke, and wave the smoke fumes in front of the sensor. You should now see the values change in the control windows. (Please take care with whatever source you are using to make the smoke with. Remember the old saying, "*There's no smoke without fire*").
5.) The more smoke that is detected, the higher the ADC values will reach. Make a note of the readings from the "ADC Value" control with a small amount of smoke, and a larger amount of smoke.
Now to make the smoke detector useful.
6.) From the menu ribbon, press "**Project**", "**Add controls**", "**Scripting**", then "**EZ-Script**" to add this to your project.

7.) Open the script configuration menu, and write or copy/paste the following script...
#Read's the value on EZ-B port A0
$smoke=getADC(adc0)
#Will detect LOW level smoke over 40
#Change the values if necessary.
if($smoke>40)
saywait("I am now sensing smoke, or gas fumes in the area.")
Sleep(10000)
#Will detect HIGH level smoke over 60
elseif($smoke>60)
saywait("Warning. I am now sensing an increased level of smoke, or gas fumes in the area.")
endif
goto(loop)[/code]
You can adjust the script above, and add a "greater than" 40 and "less then" 59 in the first half of the script, so only low levels of smoke will be detected. Anything over 60, and the second half will run. (Unable to post this example as using the greater than and less than symbols in this document. See screenshot below)...

Below, is a video demonstration showing the smoke sensor using both the "ADC Read" and "Script" controls in action...
And below is another video demonstration of the smoke sensor being used on a robot...
The next step will guide you through using sound sensors on an analog port. sound sensor

Step 7. ADC Example with Sound Sensors.

This step will guide you though setting up and using a Sound Sensors and make it turn an LED on and off. It will also show the "ADC Meter" control in use.
1.) Connect the sound sensor to an analog port on the EZ-B (This example will use "A0"). Now connect up a single 3 volt LED to the "Signal" and "Ground" pins on a digital port (D0 in this example), ensuring that the "Anode" (+) goes to the signal pin, and "Cathode" (-) goes to the ground pin. Then power up the EZ-B.

2.) Open EZ-Builder, then from the menu ribbon, click "**Project**", "**Add controls**", "**ADC**", then "**ADC Meter**" to add this to your project.

3.) On the "ADC Meter" control, click on the gear icon to open the configuration menu, and select the port the sensor is connected to, then click "**Save**".

4.) Now connect the EZ-B to Ez-Builder, and make some noise... literally. The louder the sound is detected, the higher the visual display will go in the control window. You may need to adjust the potentiometer on the sound sensor to adjust the sensitivity.
5.) Make a note of the readings when quiet, and when sound is heard. You will want to use an "ADC Read" control for this.
Now to make the sound detector turn on an LED when sound is heard.
6.) From the menu ribbon, press "**Project**", "**Add controls**", "**Scripting**", then "**EZ-Script**" to add this to your project.

7.) Open the script configuration menu, and write or copy/paste the following script...
```
#Read's the value on EZ-B port A0
$sound=getADC(adc0)
#Will turn LED on when sensor reading is above 5.
#Change the values if necessary.
if($sound>5)
set(d0,On)
#Will turn LED off when sensor reading is below 4.
elseif($sound<4)
set(d0,Off)
endif
goto(loop)[/code]
```
Below, is a short video demo of the script in action...
Yoy may have noticed in the video, that the visual output on the "ADC Graph" didn't go very high. The link below, is a more extensive script using two sound sensors that also uses a multiplier to increase sensor readings. Basically by multiplying the sound values, you are magnifying the sound sensor values and you filter out random background noises that the sensor may pick up. A fellow community, @mtiberia, member did exactly this using two sensors, and his thread explains and demonstrates this very well. He also has a great video demo, and also shares his script. Click on the link below to have a look.
EZ-B can figure out what side sound is coming from.

Final Thoughts.

So hopefully you will see how useful the analog ports and ADC controls are, and how versatile these can be.
People hear the word "Analog" and think "*old fashioned*", but when you look in to it, analog can do some pretty cool stuff. There are members here that have customized servos and motors to help with position feedback, and even made their own pressure sensors just using some wire connected to an analog port.
With a better understanding of what you can use analog for, and how to add and use the available ADC controls, get creative and share your projects with the EZ-Robot community.
Happy building.
Tutorial created on 7th October 2015.