SYNTHIAM

synthiam.com

Setting Servo speeds and Initialization Script Tutorial

This tutorial will explain what an initialization script is and how to use them in your EZ-Builder projects. Initialization scripts, sometimes referred to as "Init" scripts are used to prime controls, devices, or existing scripts within your EZ-Builder projects, and sets them ready for use. Sometimes, you may see it referred to as a "Run Once" script. In the next few steps, I will go through how to set an "Init" script up, and go through some of the main uses for this function, that includes servos speeds and setting variables.

Last Updated: 10/16/2015

Step 1. Servo Speed Init, Part 1.

Some of you who are new to EZ-Builder and use servos with your robotic projects, may notice that if you have your servo speeds set to go slower speeds when your robot performs an action, after connecting you robots EZ-B to your EZ-Builder project and you move any of the servos connected, they will move at full speed the first time they move, then move at the speeds you have set them at. This is not a fault or an error, but simply this is the way most servos are designed.
Quote:
Taken from the EZ-Script menu.
ServoSpeed (servoPort, speed)
Set the speed of servo or PWM. This is the speed to move between positions.
The servo speed is a number between 0 (fastest) and 10 (slowest).
*Note: To initialize the ServoSpeed() at first use, set a Servo() position before using the ServoSpeed() command. If there is no previous position (such as during power-on), the software assumes the position is 0 and will cause issues with your robot.
*Note: Once the ServoSpeed() has been initialized the first time, specify the ServoSpeed() before specifying the Servo() position.
Example: ServoSpeed(D14, 25)
Once power is supplied to the servo for the first time and the EZ-B starts to communicate with that servo, the servo does not know what it's speed is set at or its position to begin with. Depending on the design of you robot, be it a revolution to out or one you have built yourself, this initial fast sudden movement could possibly cause damage to your robot, so what is needed is a way to slow these servos down when they are first moved. This is where an "Init" script comes in. This will initiate the servo speed and position, then prime the servos for the speeds you wish to set.
So how is this done? There are two ways to do this and I will explain both. In the script examples, 7 servos will be used.
The first way. [b]
[b]1.) Open up your EZ-Builder project on your computer, and click on the little gear icon next to your EZ-B's IP address on the "**Connection control**" and this will open up the script editor.

2.) In the script editor, write or copy/paste the following script...
NB: *Any body of writing in the following, or any script, with a **#** in front of it, is just text for explanation purposes and can be removed or changed if necessary.*

```
# This will clear the servo speeds for initialization.
ServoSpeed(D1, 0)
ServoSpeed(D2, 0)
ServoSpeed(D3, 0)
ServoSpeed(D4, 0)
ServoSpeed(D5, 0)
ServoSpeed(D6, 0)
ServoSpeed(D7, 0)
```

The following will move the servos into desired starting positions.

```
Servo(D1, 90)
Servo(D2, 40)
Servo(D3, 90)
```

```
Servo(D4, 90)
Servo(D5, 38)
Servo(D6, 82)
Servo(D7, 40)

# The following will now set the servo speed for these servos which will be

#used from now on.
ServoSpeed(D1, 2)
ServoSpeed(D2, 3)
ServoSpeed(D3, 1)
ServoSpeed(D4, 2)
ServoSpeed(D5, 1)
ServoSpeed(D6, 3)
ServoSpeed(D7, 2)
```

3.) Click on "**Save**" to save the script and close the script edition, the click "**Save**" again to close the "Connection Control", menu.

There is one thing to keep in mind when using this servo speed "Init" script for servo speed and position. I'll give you a working example...
Your robot has a servo controlled head and you have it set at "90" in your Init script. When you power off your EZ-B, maybe the head servo is facing at "180". When you power on and connect your EZ-B to the EZ Builder project again, that servo will still go at full speed to the position back to the "90" you have set up as in your "Init" script.
One way to combat this is to do a reverse "Init" script. By this I mean a kind of Off button in EZ Builder that would be pressed when you are finished with your robot. This could be an EZ-Script control, or a button on the mobile interface. This is where you would use the same servo speed and position "Init" script to move the servos to their starting position, so when you turn it on again, the servo don't move.

Step 2. Servo Speed Init, Part 2.

You may have several "Init" commands for different scripts and/or controls in you project that all need to start when you connect your EZ-B to EZ-Builder. Some of these "Init" scripts you may wish to call again while your project is already running, for example, the Off button I mentioned in the last step. So instead of having one large script in the "Connection Control" containing all of your "Init" scripts, add a "**Script Manager**" control to your project, and place each of your "Init" commands in individual script controls. So, using our servo speed example...
1.) On the EZ Builder menu ribbon, click on "**Project**", then "**Add Controls**", then click "**Scripting**", and then "**Script Manager**" to add the control to your project.

2.) This step is optional, but it is recommended. On the bottom of the "Script Manager" control, click on the gear icon and rename the control to something of your choosing, something like "Init Manager", then hit "**Save**".

3.) Again, on the bottom of the Script Manager control, click "**Add Script**". The script editor will automatically open.

4.) In the script editor, write in your servo speed "Init" script.

5.) This is quite an important part, and this is to name your script. On the top left of the script editor, click on the text input field (it may already say something like "Script 1") and rename your new script to something like "Servo Speed Init", and then press "**Save**" to close the script editor.

A final couple of steps are needed now to add the command in the "Connection control" to automatically start your servo speed "Init" script.
6.) Click on the gear icon next to your EZ-B's IP address on the "Connection Control" to open the script editor.

7.) In the script editor, right click your mouse/trackpad button to bring up the list of cheat sheet commands, find the name of your script manager (in our example, "Init Manager") and click to expand the command list. Click on "**ControlCommand,Init Manager,Start**", then Click "**Save**" to close the editor, and "**Save**" again to close the "Connection Control" menu.

8.) Now all that remains, is to save these changes to your EZ Builder project by hitting "**File**" and then "**Save**" in the menu ribbon.

Step 3. Servo Speed Init with Multiple Boards

You may have multiple EZ-B's connected to your EZ Builder project, with servos connected to each board. In this case, some slight changes need to made to the servo speed and position "Init" script. To use an example, we will say that we have two EZ-B's connected to "Board 0" and "Board 2" in one EZ Builder project.
1.) Follow one of the previous steps to create an "Init" script, and open it's script editor.
2.) For servos connected to "Board 0", no change in the original "Init" script is needed.
3.) Any script lines that are controlling servos connected to "Board 2" will need to be started with a **2.** (a number two and a period), so the script should look something like this...

```
 #Servo speed initialization.
## Servos connected to board 0.
ServoSpeed(D1, 0)
ServoSpeed(D2, 0)
ServoSpeed(D3, 0)
## Servos connected to board 2.
ServoSpeed(2.D4, 0)
ServoSpeed(2.D5, 0)
ServoSpeed(2.D6, 0)
ServoSpeed(2.D7, 0)
## Desired starting positions.
## Servos connected to board 0.
Servo(D1, 90)
Servo(D2, 40)
Servo(D3, 90)
## Servos connected to board 2.
Servo(2.D4, 90)
Servo(2.D5, 38)
Servo(2.D6, 82)
Servo(2.D7, 40)
## Newly set servo speeds
## Servos connected to board 0.
ServoSpeed(D1, 2)
ServoSpeed(D2, 3)
ServoSpeed(D3, 1)
## Servos connected to board 2.
```

```
ServoSpeed(2.D4, 2)
ServoSpeed(2.D5, 1)
ServoSpeed(2.D6, 3)
ServoSpeed(2.D7, 2)
```

In the next step, I will explain further uses for an "Init" script with a couple of examples.

Step 4. Init scripts using varables.

Some scripts use variables, and some of these scripts need a primer to set the variable to be used. The following example sets a variable to be used with the Mobile Interface Control. This variable allows you to have two uses for one button which is great for saving space on the mobile control, as you only need one button instead of having two. The following example can be used to drive a robot forwards then to stop it from moving, and assumes you have a movement control panel already added to your project.
1.) In a "Connection" script, either in the "Connection Control" script editor, or in a connection script in a script manager like was explained in steps 1 and 2, enter the following "Init" script and save it...

```
$pressed = 0
```

2.) From the EZ-Builder menu ribbon, add a "Mobile interface" control by clicking on **Project** then **Add Controls**, **Mobile**, then **Interface Builder**.

3.) Click on the gear icon on the mobile control to open the configuration menu.

4.) Add a "Forwards" button to the mobile control.

5.) Click on the "Forwards" button, then on the lower right of the mobile control, click on "**Edit Script** tab...

and copy in the following script...

```
if (!$pressed)
Forward()
$pressed = 1
ELSE
Stop()
$pressed = 0
endif
```

6.) Click "**Save**" on the script editor, then click "**Save**" on the mobile control.

7.) Power on your EZ-B and connect it to your EZ Builder project. When the connection is made, the "Init" script will automaticly run and the "!$pressed" varable will be set. When you press the "Forwards" button on your mobile interface control, the motors on your robot should move. When you press the same button again, the motors will stop.

Step 5. Another Init script example.

In this example, we use another variable to set EZ-Builder to tell you the current date using your computers clock.
1.) As before, in either a "Connection" script in the "Connection Control" script editor, or in a connection script in a script manager, enter the following "Init" script and save it...

```
$SayDate = 0
$lastDay = $day
```

2.) From the EZ Builder menu ribbon, either add an EZ-Script control by clicking on "**Project**" then "**Add Controls**", "**Scripting**", then "**EZ-Script**" and click on the gear icon to open the script editor.
2A.) or click on "**Project**" then "**Add Controls**", "**Scripting**", then "**EZ-Script Manager**". On the script manager, click on "**Add Script**" and click on "**Edit**" to open the script editor.
3.) Copy the following script in to the script editor then hit **Save** when you're done...

```
# Script courtesy of Rich Pyke.
if ($lastDay != $day)
goto(sayDay)
endif
if ($SayDate  = 1)
goto(toldYou)
endif
```

Say day, date, month

```
:sayDay
SayWait("Today is " + $dayName + " , the " + $day  + " of, " + $monthName )
$lastDay = $day
$SayDate  = 1
halt()
```

Repeat day, date, month

```
:toldYou
SayWait("Haven't I already told you, it is " + $dayName + " , the " + $day  + " of, " +
$monthName )
$SayDate  = 0
halt()
```

4.) Power on your EZ-B and connect it to your EZ Builder project. When the connection is made, the "Init" script will automatically run and set the vairabe. When You press "**Start**" on your date script, you computer will tell you the current date. Pressing "**Start**" again, you computer will tell you that you already asked for the date, and tell you it again.

Step 6. UARTInit() Commands.

Quote:
[/b]Taken form the EZ-Script menu.[/b]
UARTInit(boardIndex, port, baudRate)
Initialize the Peripheral UART on the EZ-B v4 with the specified baud rate. The UART will stay initialized until the EZ-B v4 is power cycled, and therefore this command only needs to be called once.
The Board Index is the EZ-B index starting at 0.
The port can be 0, 1 or 2.
The baud rate can be between 1 and 3750000 bps. The UART receive buffers on the EZ-B v4 are 5,000 bytes.
Look at the UART Port section in the EZ-Script menu for the EZ-B Pin's associated with each UART Port.
Example: UARTInit(0, 0, 9600)
When using the UART0 port it always has to be initialized the first time you use it after power up by using the UARTInit() command. Once initialized, you don't have to send that command again untill the next time you power on the EZ-B.
Breakdown of the UARTInit() command.

```
UARTInit(0, 0, 9600 )
```

.) In the "UARTInit" command above, the first number is for defining the EZ-B v4's board number (the board number that the EZ-B connects to in the connection control),
.) The second number is for defining the UART port number (on the EZ-B),
.) And the last number is the baud rate, which must match the same baud rate number on the device you are going to use (Roomba, Sabertooth, Kangaroo ect).
There are lots of other reasons to use "Init" scripts, such as using an iRobot Roomba vacuum cleaners with an EZ-B. These require an "Init" script to initialize the controls. There are projects in the EZ-Cloud that use a Roomba movement panel which has a built in "Init" button, but it can be written as a script as well. Such an "Init script could look like the following...

```
#Init script for "Roomba 530".
UARTInit(0, 1, 115200)
```

Another example for a UARTInit() command, is for the SSC-32 servo controller. There is a dedicated control for the SSC-32 found in EZ-Builder, found by selecting "**Project**", "**Add controls**", "**Servo**", "**SSC-32**". But if you wish to use a script to control servos connected to the SSC-32, you will need a UARTInit() script first...

```
#Init script for "SSC-32" that has a choice
#of baud rate baud rate.
UARTInit(0,0,9600)
UARTInit(0,0,38400)
UARTInit(0,0,115200)
```

These types of "Init" scripts can be placed in a connection script like has been mentioned previously, or can be put in to a script, and run when needed.

Final Thoughts.

There are a few controls found in EZ-Buider that have "Init" command tabs built in to the controls, as seen below...
HT16K33 8x8 Animator.

MPU9150 4 in 1 sensor.

These "Init" commands can be run straight from the control by clicking the "**Init**" tabs when needed, but they can also be run using the controls "Cheat Sheet" commands found in script editors...

Hopefully, the examples given in this tutorial will give you a good idea of what initialization scripts do, what they can be used for, and how to use them.
Happy building.
Tutorial created on 3rd October 2015