8. Connecting Arduino devices with Wolfram SystemModeler

This video demonstrates a way to observe an integrated system at work by showing how the Wolfram language can connect to open-source Arduino devices so data can be read and written to shields and extensions. It also shows how to integrate Arduino boards to simulations in SystemModeler, enabling the user to visualise the inputs and outputs of the Arduino, and make connections between the two.

Transcript

**Colin:** [00:00:20] In this video, we're going to integrate our Arduino with systems modeller. There are a lot of advantages here because you can actually go and visualise the inputs and outputs to the Arduino and the students can make connections between the two, which is a really nice way for them to be able to show that they have an integrated system.

**Colin:** [00:00:39] The first thing is we need to prepare the Arduino. So as you can see, I've plugged the Arduino into the computer. I now need to go to file, examples. If we scroll down, there will be a section called Firmata. In the Firmata section, we need to go and look for standard Firmata. Click down there. And the IDE produces the code. We upload the code. This Firmata allows the software, which isn't native to the Arduino, to be abble to communicate with it.

**Colin:** [00:01:19] And there it's uploaded. Once we've done that, we don't need to use the Arduino id any longer and we can simply put it away.

**Colin:** [00:01:28] Now we go into SystemModeler. First thing is we need to open our new class, give it a name that makes sense. So in this case, we're just going to give a simple output so I'll call it L.E.D.. Press Okay. And as with their good habits, we need to make sure we save this. And we want to save it on the Wolfram Mathematica. And, there we are.

**Colin:** [00:02:04] Now, some of the components for this are in the model plug library and summary in my Modelica. So for the L.E.D., we need to go to model plug. Open it up. And there are a number of boards. So we're going to pick the Arduino board and drag that across. And I'm going to enlarge it so I can see it.

**Colin:** [00:02:35] The next thing we need to do is to get an input output, so we click on pins and we want a digital output. Drag that across. And again, I'll enlarge that, so it's easier for me to see. Close by. The next thing we need is a source. The source is found under Modelica blocks. And then from blocks we need to go to sources. And we scroll down until we get to a BooleanPulse. We pull the BooleanPulse across and again, I'll enlarge that. So now we have our three components, the BooleanPulse, the output from the Arduino and the Arduino itself.

**Colin:** [00:03:31] We need to wire this up? So we go to the connection tool. And connect the BooleanPulse to the output and then the output to the Arduino and that's done. And of course, we can now verify that. So you go to the validate class and we have a successful validation.

**Colin:** [00:03:55] Now, there are a few things we need to do here. First of all, we need to tell the software how it's communicating to the Arduino. So if you click on that. Arduino there. I'll get the right cursor, there we are. You'll see that there's a field for port. So if we click the little editing tool next to it, it will actually tell us what ports are available and which one is connected to your Arduino. In this case, it's told us it's port number three or comm three. So press Okay. This is the easiest way to make sure you got the correct connection. The next thing to think of that is which pin on the Arduino that we would like to connect their output device. In this case, it will be an LED and ongoing to connect it to pin number eight. And I choose eight because it's the first pin in the second lot of digital blocks and the Arduino. So it's easy to find. I'll just make sure I've got that number correct. Eight.

**Colin:** [00:04:56] Then the final thing that we're going to adjust is the period. Now you can choose the width of the signal in all sorts of things, but the easiest thing to do is simply to put your period under the time and it's in blocks of a second. So I've set all that up. So now I'll build the circuit. Now I said we're going to connect the LED to PIN number eight. And remember, we're going long leg goes into the port, pin number eight, and then into the breadboard like so. We won't be using a resistor for this experiment, although you ordinarily would, we just want to say some time. Then I'm going to connect the short leg of the LED to ground. Plug. And on this shield here, ground is in that set up blocks there, so they are connected my ciruit.

**Colin:** [00:05:56] So now it's time to simulate. So I click on the simulate class button. Again, we can ignore this screen here and just click, simulate. And the screen comes up, you can see that we've got a blank canvas for plotting. Now there are a few things to work with as before with simulations we will go to synchronize with real time. And because we don't want to miss the action, we're going to pick the stop time as being infinite. We can always use a pause button or stop button if we need to. The next thing I'm going to click is plot. Now, this is very useful because later on we're going to find out how we can link the input and output of our projects. And if we use this for our simulations, students can annotate the diagrams and show what's really happening.

**Colin:** [00:06:55] So for this one here, I'm simply going to put in the digital output and I'm going to click the u here, which is the output from the pin. You can see it gives us a nice digital wave. Then I will press simulate. And you can see the LED is blinking and it's blinking in time with the digital pulse on the output from the Mathematica SystemModeler.