Notes for Instructors

The Scratch for Arduino module is designed to introduce non-experienced programmers to creating interactive prototypes. Participants who have no programming experience will quickly become familiar with basic concepts and those with experience should readily be able to take up the language.

The module starts by introducing all the sensor (inputs) and actuators (outputs) that are available, and describes them in real-world terms. For example, it easier to describe a photocell as something that measures the amount of light, or an accelerometer as being able to measure rotation. Also, terms like “Analog”, “Digital”, “Input”, and “Output” should be discussed and explained to the participants.

After introducing key terms, the remainder of the module is walking through examples of different sensors and actuators, which increase in difficulty as they go on. The first demo is to simply turn on the on-board LED on the Arduino. Once they have accomplished this, the idea of delays and loops can be added to make the LED blink. Later demos can read in physical values from sensors and store them in variables. After introducing a sensor (e.g. range detector) and an actuator (e.g. servo motor), the next example is to connect the two together such that one controls the other. At this point the participants have covered a sufficient number of topics such that they can being to explore on their own.

For the module to be successful, a moderate amount of preparation is required. First, to ease the burden on the participants, it is helpful to wire difficult sensors and actuators before the workshop. Sensors that need voltage dividers, such as a photocell or force resistor, or have unlabeled pins, such as a LED, may be difficult for participants to breadboard during the workshop. However, after they are more familiar with the hardware and software, participants should be confident to wire in their own sensors near the end of the workshop.

Second, the drivers for the Arduino need to be install before S4A will work. This can be a huge headache if unanticipated. This is a problem on Windows systems as the drivers cannot be installed until an Arduino is physically plugged in. Mac drivers can be installed beforehand and Linux drivers are likely already installed. Depending on the number of participants, this will cause a significant delay in starting the workshop. Instructions for installing the drivers are include in the bill of materials document. The best practice we have found is for everyone to download the drivers to their desktop, regardless of the system in use. Sometimes Windows Update will automatically find the drivers on the internet, or it’s possible they’re already in place. Regardless, if the software is located on the participant’s desktop, is it easier to find and install before the workshop. Likely as people come in the drivers can be installed before beginning the demonstration. Additionally, if they are working in groups only one computer per group needs to drivers.

Wiring sensors and dealing with the driver install are the two great hurdles before the workshop. If these items can be handled, then the remainder is simply to have sufficient number of examples and at an appropriate difficulty. Not all sensor or actuators will be compatible with S4A, and in some cases the S4A firmware may need to be modified. For the servo motors we selected, the delays programmed into the firmware did not match our motor, such that it was still moving when it should be still. By manually changing these delays, we were able to overcome this problem. Finally, the S4A language is written in Smalltalk and is therefore difficult to modify. Currently there is not direct access to external ports such as the I2C bus or UART port on the Arduino. Sensors that require the interfaces cannot currently be supported.