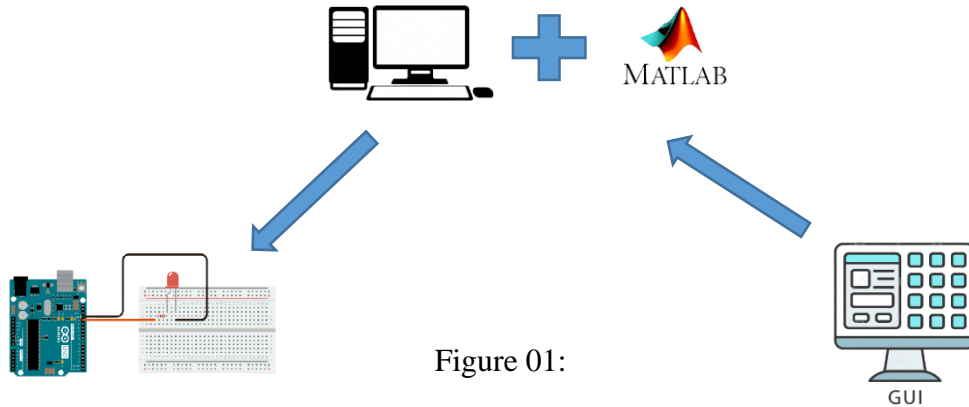
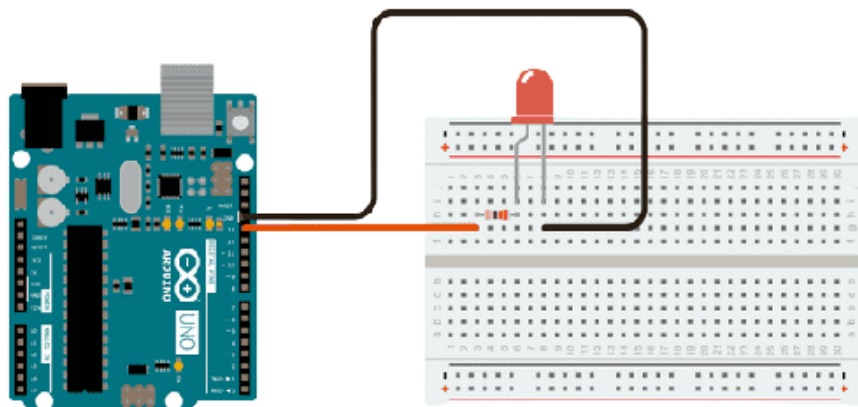


Lab Sheet
IA 3018 – Data Acquisition Systems
Department of Instrumentation and Automation Technology
University of Colombo
Practical 01



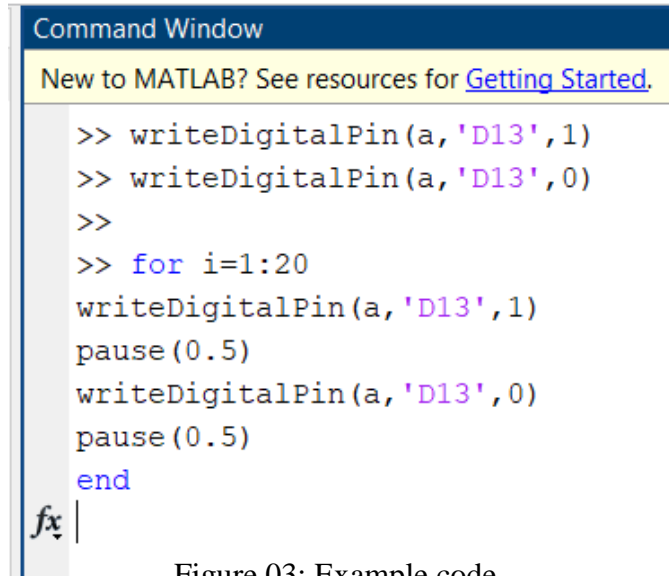
Part 01 -Writing output using MATLAB and Arduino (Blink LED)

1. Add the Arduino support package to MATLAB. (Appendix 01)
2. Connect the Arduino board to computer and record the COM port and board name.(Nano, UNO, Mega, Due)
3. Check the Arduino support package are working in MATLAB using above recorded data in part 2. (Appendix 02)
4. Create the circuit as figure 02.



5. Write a MATLAB code for LED on and off.

6. Write a MATLAB code using loop (For, While) for blink LED ten times with 1S delay.



```
Command Window

New to MATLAB? See resources for Getting Started.

>> writeDigitalPin(a, 'D13', 1)
>> writeDigitalPin(a, 'D13', 0)
>>
>> for i=1:20
writeDigitalPin(a, 'D13', 1)
pause(0.5)
writeDigitalPin(a, 'D13', 0)
pause(0.5)
end
fx |
```

Figure 03: Example code

Part 02 – Create a GUI (Graphical User Interface) for blink led using MATLAB and Arduino.

1. Open GUI window in Matlab.
2. Select blank GUI and save as “Blink_LED”.
3. Add topic for the GUI using “Text” function and change the appearance.
4. Add two push buttons and change the appearance (Name, Color, and Font).

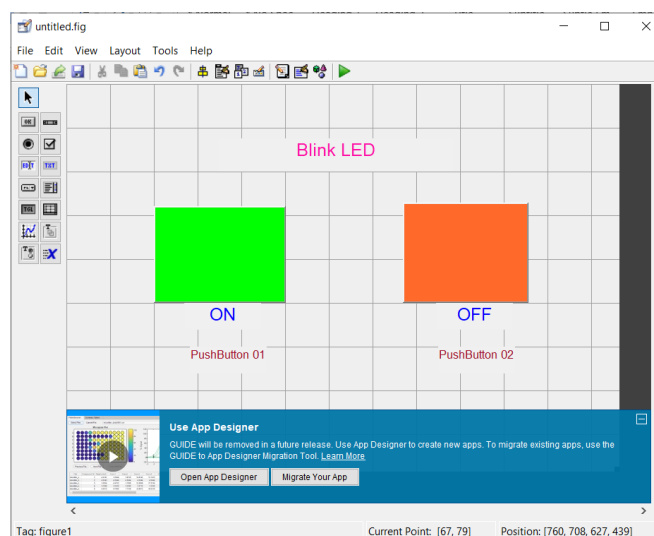


Figure 04: GUI editor

5. Edit the GUI script using bellow (figure 05) commands codes.

```
Editor - E:\Matlab\Example\Practical 01\BlinkLED.m
BlinkLED.m
71
72 % Get default command line output from handles structure
73 varargout{1} = handles.output;
74
75 clear all;
76 global a;
77 a = arduino;
78
79 % --- Executes on button press in pushbutton1.
80 function pushbutton1_Callback(hObject, eventdata, handles)
81 % hObject handle to pushbutton1 (see GCBO)
82 % eventdata reserved - to be defined in a future version of MATLAB
83 % handles structure with handles and user data (see GUIDATA)
84
85 global a;
86 writeDigitalPin(a, 'D13', 1);
87
88 % --- Executes on button press in pushbutton2.
89 function pushbutton2_Callback(hObject, eventdata, handles)
90 % hObject handle to pushbutton2 (see GCBO)
91 % eventdata reserved - to be defined in a future version of MATLAB
92 % handles structure with handles and user data (see GUIDATA)
93
94 global a;
95 writeDigitalPin(a, 'D13', 0);
96
```

Figure 05: GUI sample script

6. Run the script and check the output.

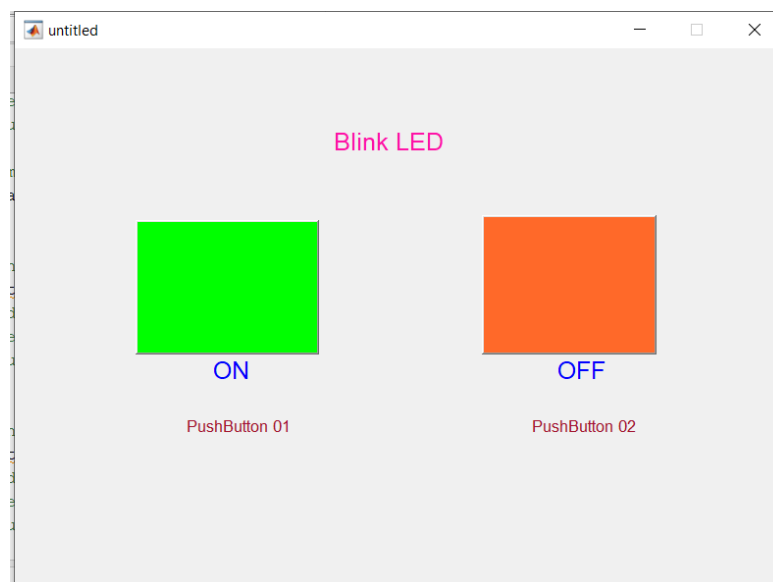
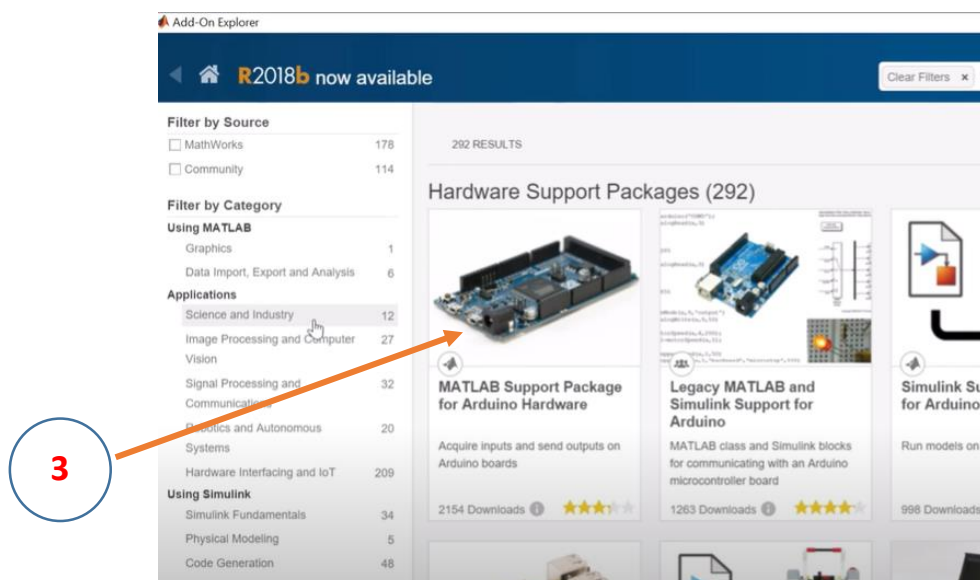
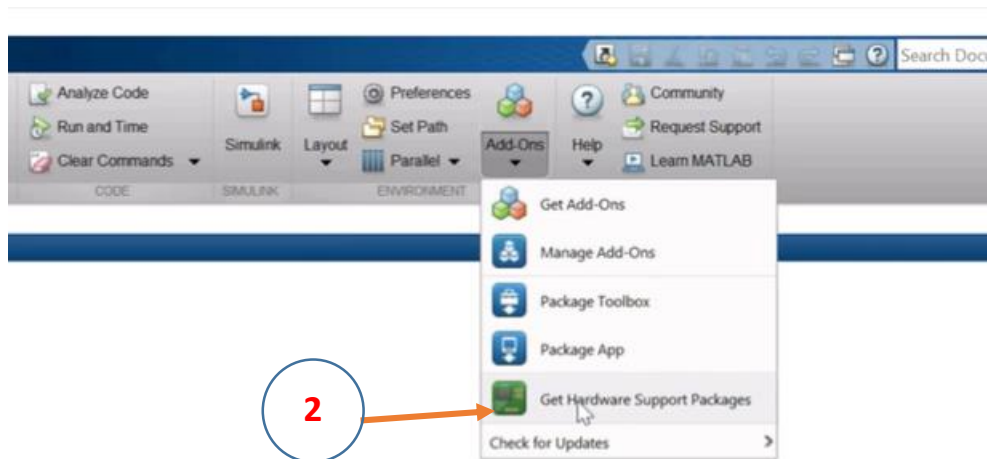
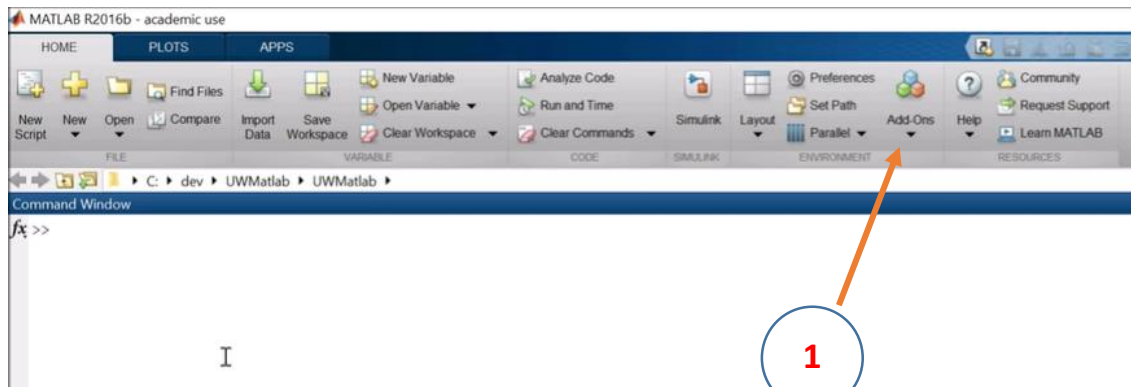
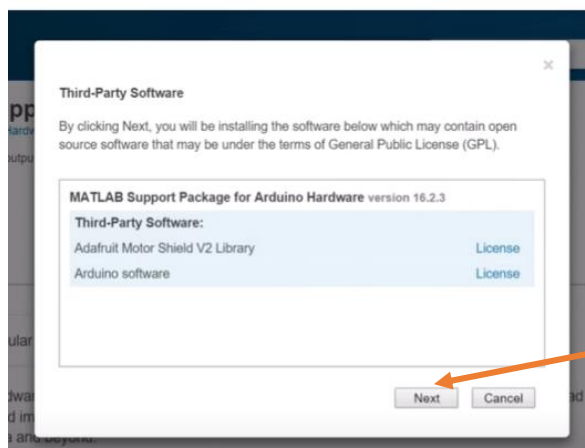
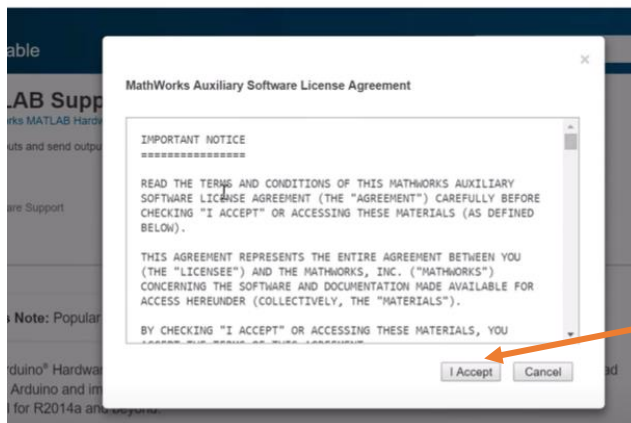
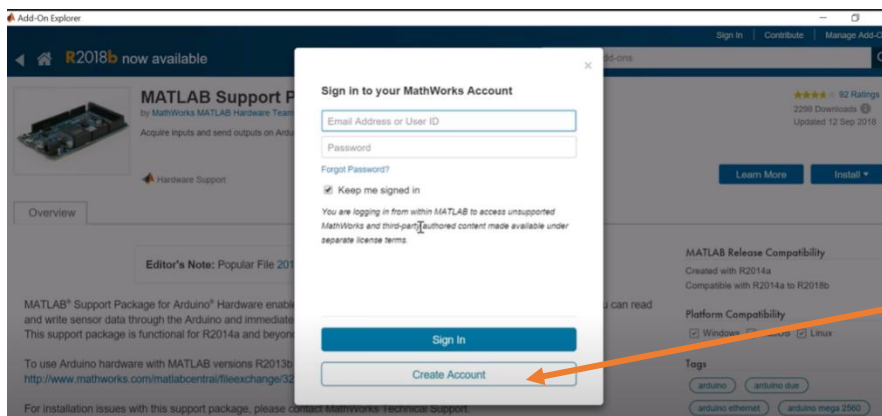
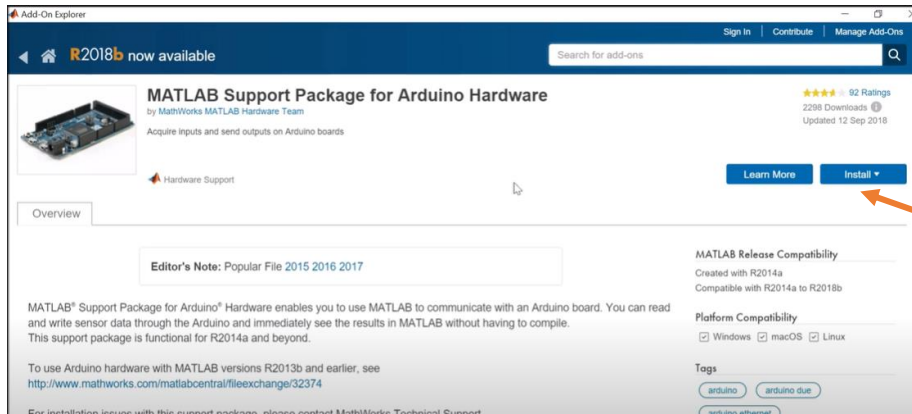


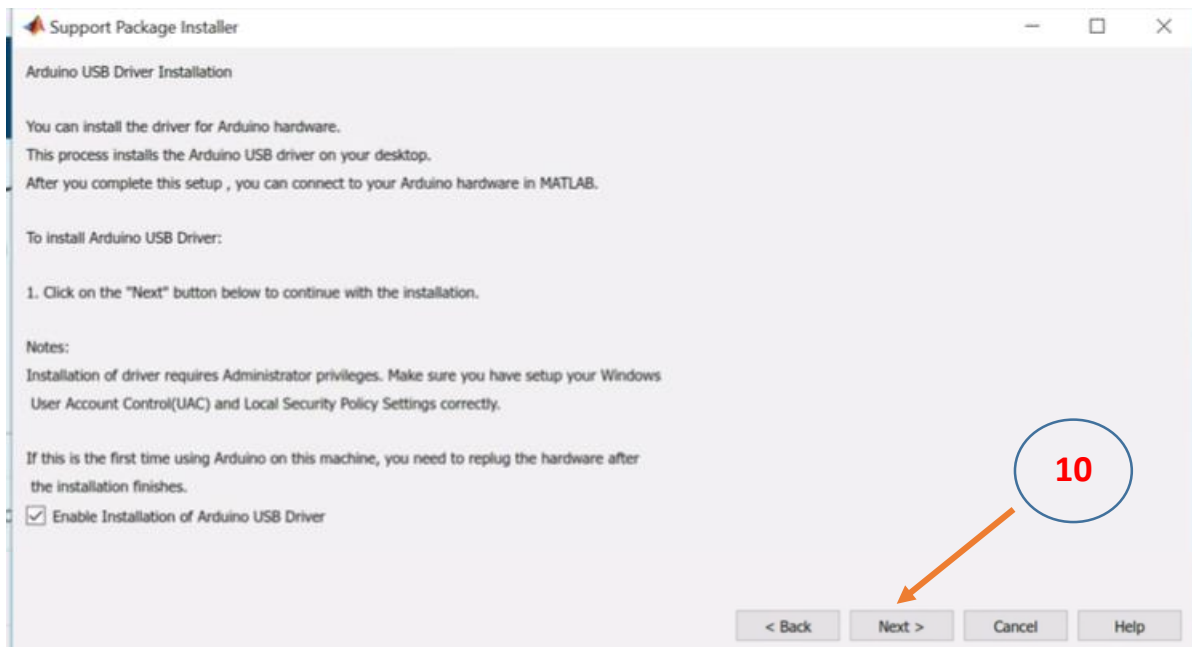
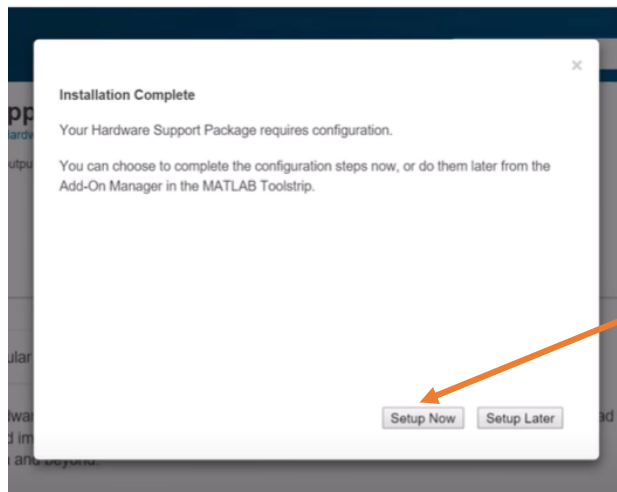
Figure 06: GUI output window

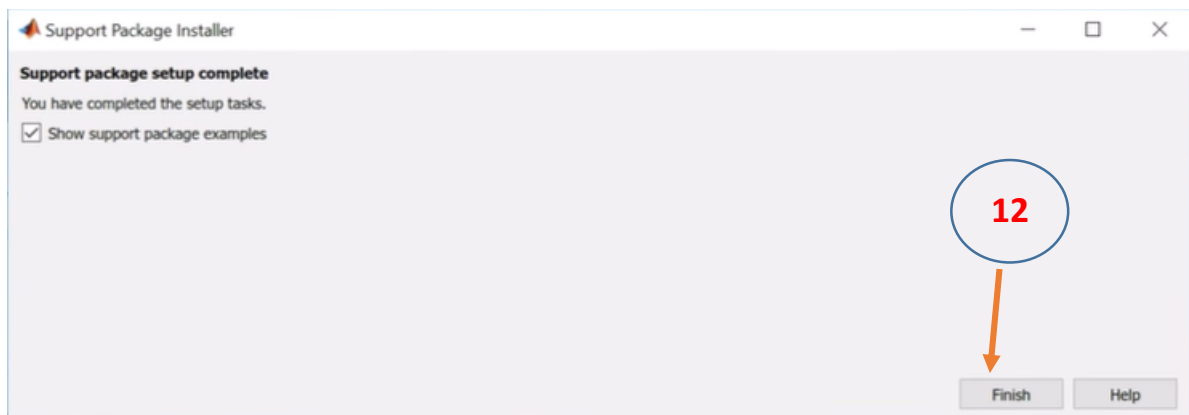
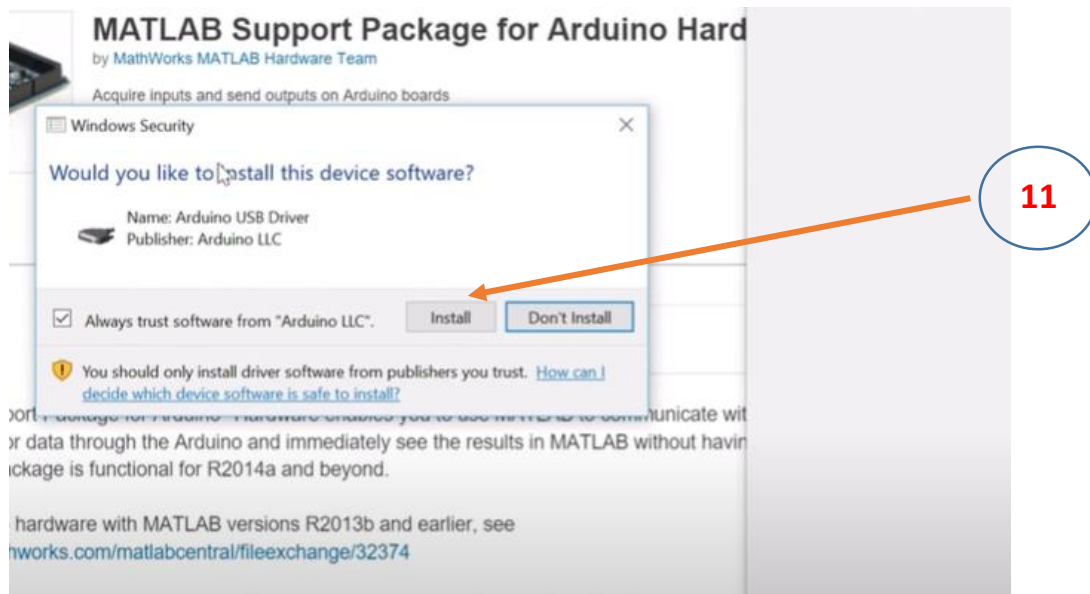
Appendix 01

How to add Arduino support package to MATLAB



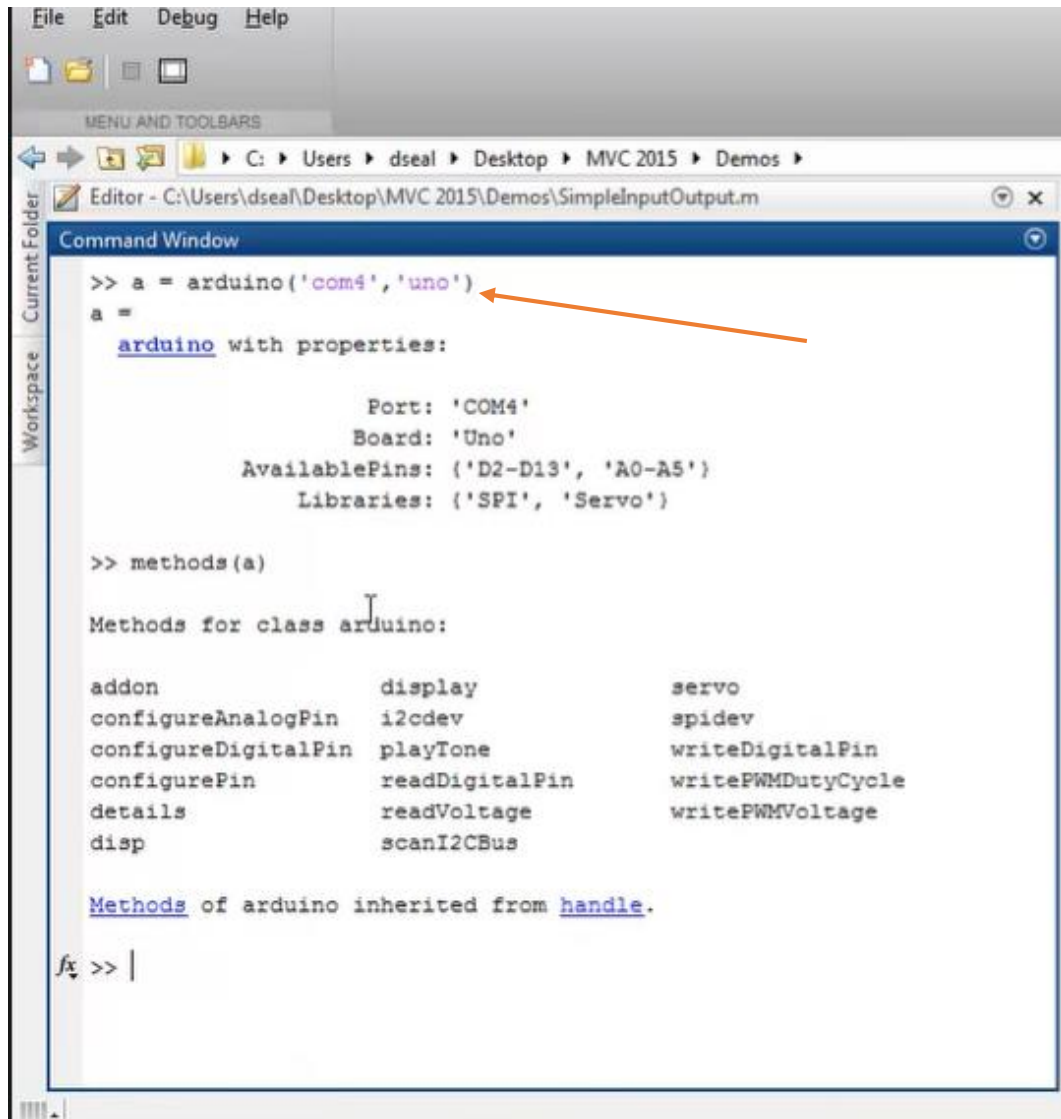






Appendix 02

How to check the Arduino support package are working in MATLAB



The image shows a MATLAB Command Window with the following content:

```
>> a = arduino('com4', 'uno')
a =
  arduino with properties:

      Port: 'COM4'
     Board: 'Uno'
AvailablePins: {'D2-D13', 'A0-A5'}
   Libraries: {'SPI', 'Servo'}

>> methods(a)

Methods for class arduino:

addon          display          servo
configureAnalogPin  i2cdev          spidev
configureDigitalPin playTone         writeDigitalPin
configurePin       readDigitalPin  writePWMDutyCycle
details           readVoltage     writePWMVoltage
disp             scanI2CBus

Methods of arduino inherited from handle.
```

An orange arrow points to the command `a = arduino('com4', 'uno')` in the Command Window.