RENSSELAER MECHATRONICS

Blinking the LED

Lab Objectives

- Install and verify Arduino software package for Simulink using a digital output to light a LED
- Communicate with the target board (Arduino) using external mode by changing the brightness of an LED with PWM
- Use overrun detection to determine how fast you can run your code

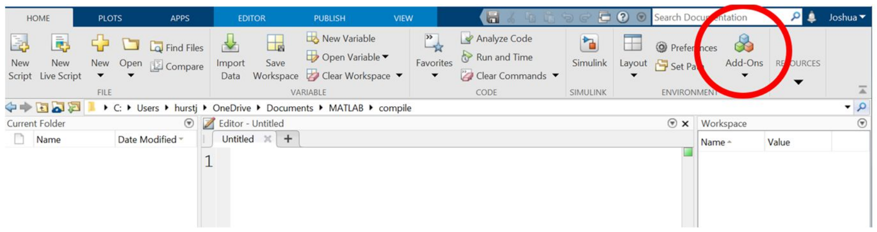
Part 1: Arduino Toolbox Installation for Simulink

Objective:

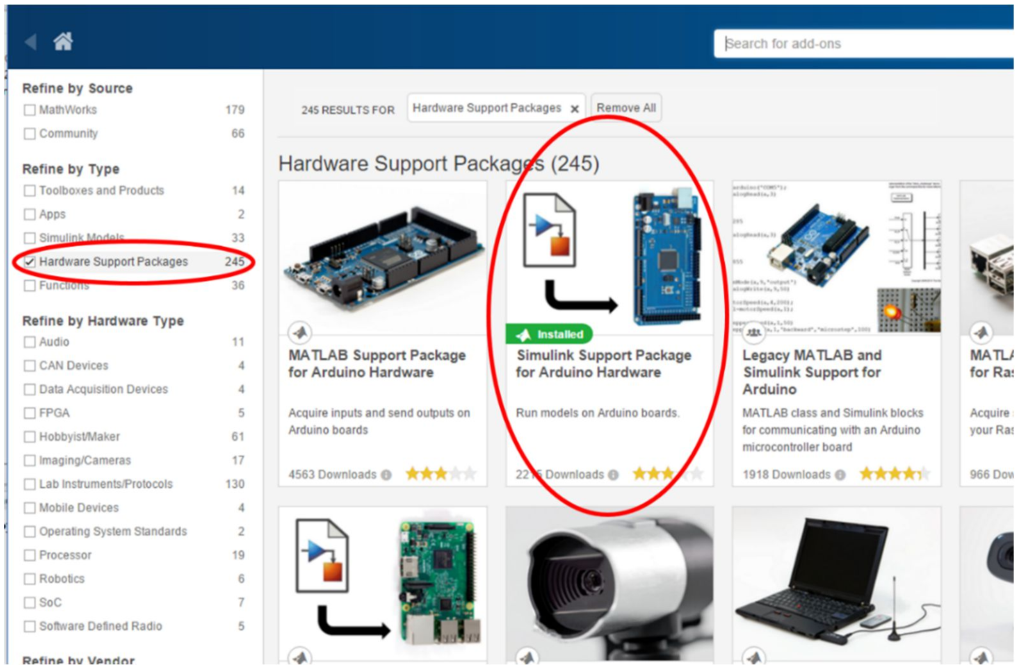
- Install and verify Arduino software package for Simulink using a digital output to light a LED with MATLAB/Simulink 2018b or later.

Simulink Arduino Library Installation:

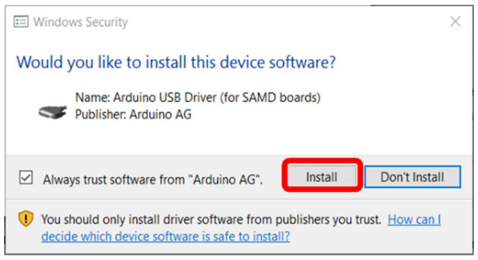
- Open MATLAB by right clicking the icon and selecting “Run as Administrator”
- Click Add-Ons (under the “HOME” tab):



- The add on explorer window will appear. Under “Filter by type” select “Hardware Support Packages” then “Simulink Support for Arduino Hardware”: (you can also type “Simulink Support for Arduino Hardware” in the ‘search for addons’ box if you have trouble finding it.


Search for add-ons
Refine by Source
MathWorks 179
Community 66
Refine by Type
Toolboxes and Products 14
Apps 2
Simulink Models 33
Hardware Support Packages 245
Functions 36
Refine by Hardware Type
Audio 11
CAN Devices 4
Data Acquisition Devices 4
FPGA 5
Hobbyist/Maker 61
Imaging/Cameras 17
Lab Instruments/Protocols 130
Mobile Devices 4
Operating System Standards 2
Processor 19
Robotics 6
SoC 7
Software Defined Radio 5
Refine by Vendor
245 RESULTS FOR
Hardware Support Packages
Remove All
Hardware Support Packages (245)
MATLAB Support Package for Arduino Hardware
Acquire inputs and send outputs on Arduino boards
4563 Downloads
Installed
Simulink Support Package for Arduino Hardware
Run models on Arduino boards.
22 Downloads
Legacy MATLAB and Simulink Support for Arduino
MATLAB class and Simulink blocks for communicating with an Arduino microcontroller board
1918 Downloads
966 Dow

- When prompted, log in using your MathWorks credentials. If you do not have an account you will need to create one.
- Follow the onscreen prompts to install the library.
- During this process, install the drivers if prompted:



Windows Security

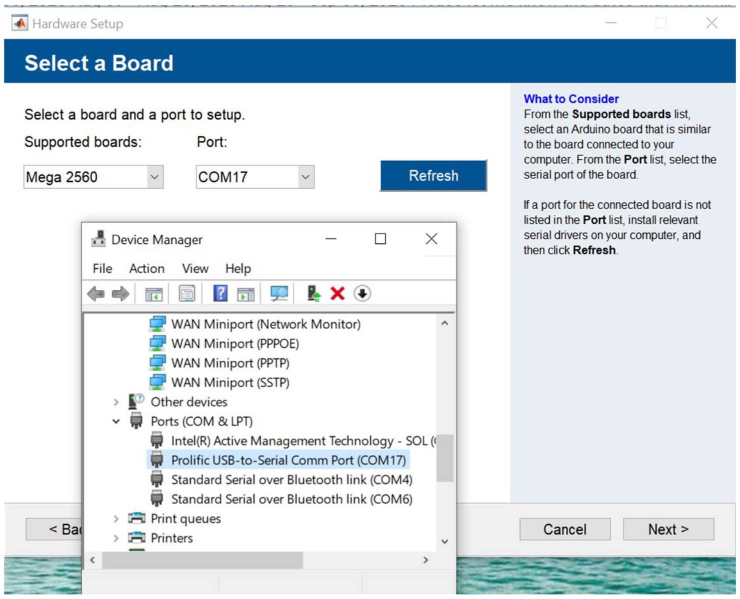
Would you like to install this device software?

Name: Arduino USB Driver (for SAMD boards)
Publisher: Arduino AG

☑ Always trust software from "Arduino AG".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

- During Hardware Setup, it might not recognize your board. In this case, manually select the board; use Device Manager to find the COM port for your board and click next.


Hardware Setup

Select a Board

Select a board and a port to setup.

Supported boards: Mega 2560

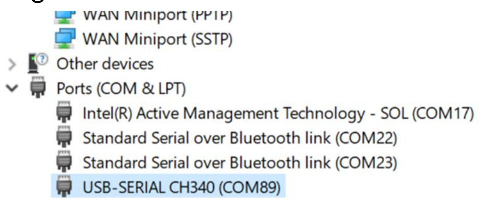
Port: COM17

Refresh
Device Manager
File
Action
View
Help
WAN Miniport (Network Monitor)
WAN Miniport (PPPOE)
WAN Miniport (PPTP)
WAN Miniport (SSTP)
Other devices
Ports (COM & LPT)
Intel(R) Active Management Technology - SOL (
Prolific USB-to-Serial Comm Port (COM17)
Standard Serial over Bluetooth link (COM4)
Standard Serial over Bluetooth link (COM6)
Print queues
Printers
< Ba
Cancel
Next >

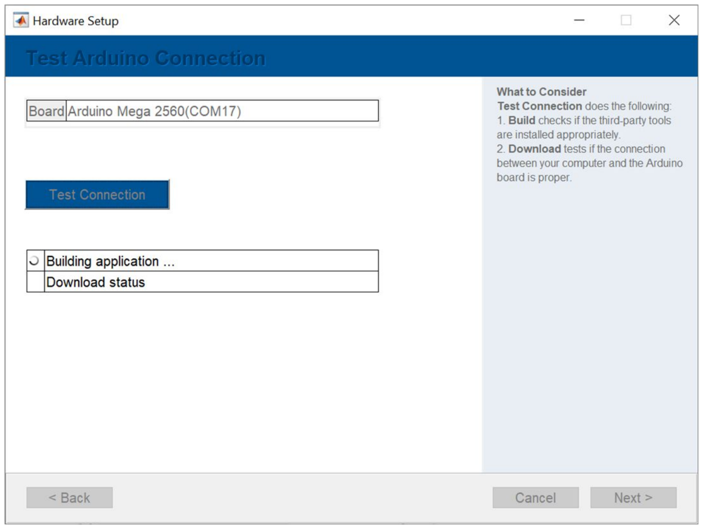
What to Consider
From the **Supported boards** list, select an Arduino board that is similar to the board connected to your computer. From the **Port** list, select the serial port of the board.

If a port for the connected board is not listed in the **Port** list, install relevant serial drivers on your computer, and then click **Refresh**.

It might read “Prolific USB-to-Serial Comm Port” Or “USB-SERIAL CH340”.


WAN Miniport (PPTP)
WAN Miniport (SSTP)
Other devices
Ports (COM & LPT)
Intel(R) Active Management Technology - SOL (COM17)
Standard Serial over Bluetooth link (COM22)
Standard Serial over Bluetooth link (COM23)
USB-SERIAL CH340 (COM89)

- Click Verify Setup to make sure it can communicate with your board

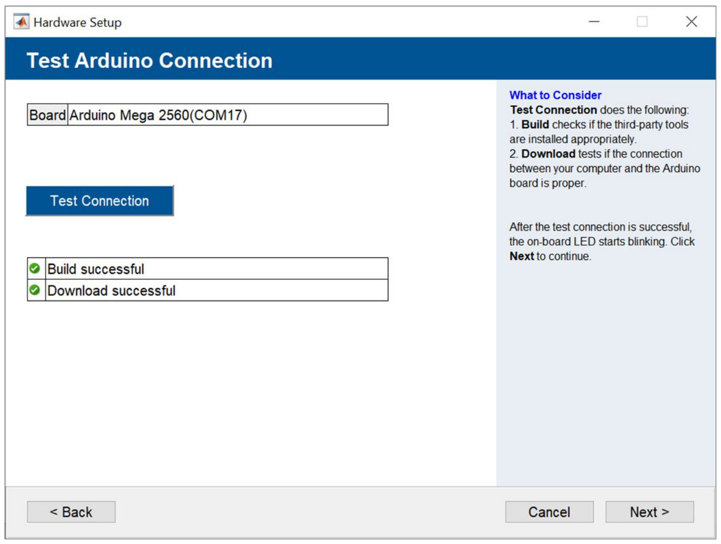

Hardware Setup

Test Arduino Connection

Board Arduino Mega 2560(COM17)

What to Consider
Test Connection does the following:
1. **Build** checks if the third-party tools are installed appropriately.
2. **Download** tests if the connection between your computer and the Arduino board is proper.

- It should indicate that it has downloaded test code that will make Led 13 blink on your board


Hardware Setup

Test Arduino Connection

Board: Arduino Mega 2560(COM17)

Test Connection

✓ Build successful

✓ Download successful

What to Consider

Test Connection does the following:

1. **Build** checks if the third-party tools are installed appropriately.
2. **Download** tests if the connection between your computer and the Arduino board is proper.

After the test connection is successful, the on-board LED starts blinking. Click **Next** to continue.

< Back | Cancel | Next >

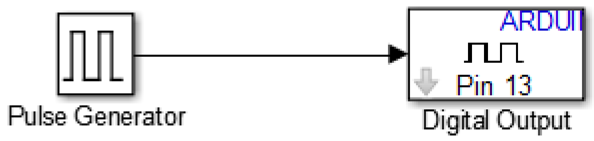
- Click Next, then Finish.

How do I find the COM port for my Arduino Board:

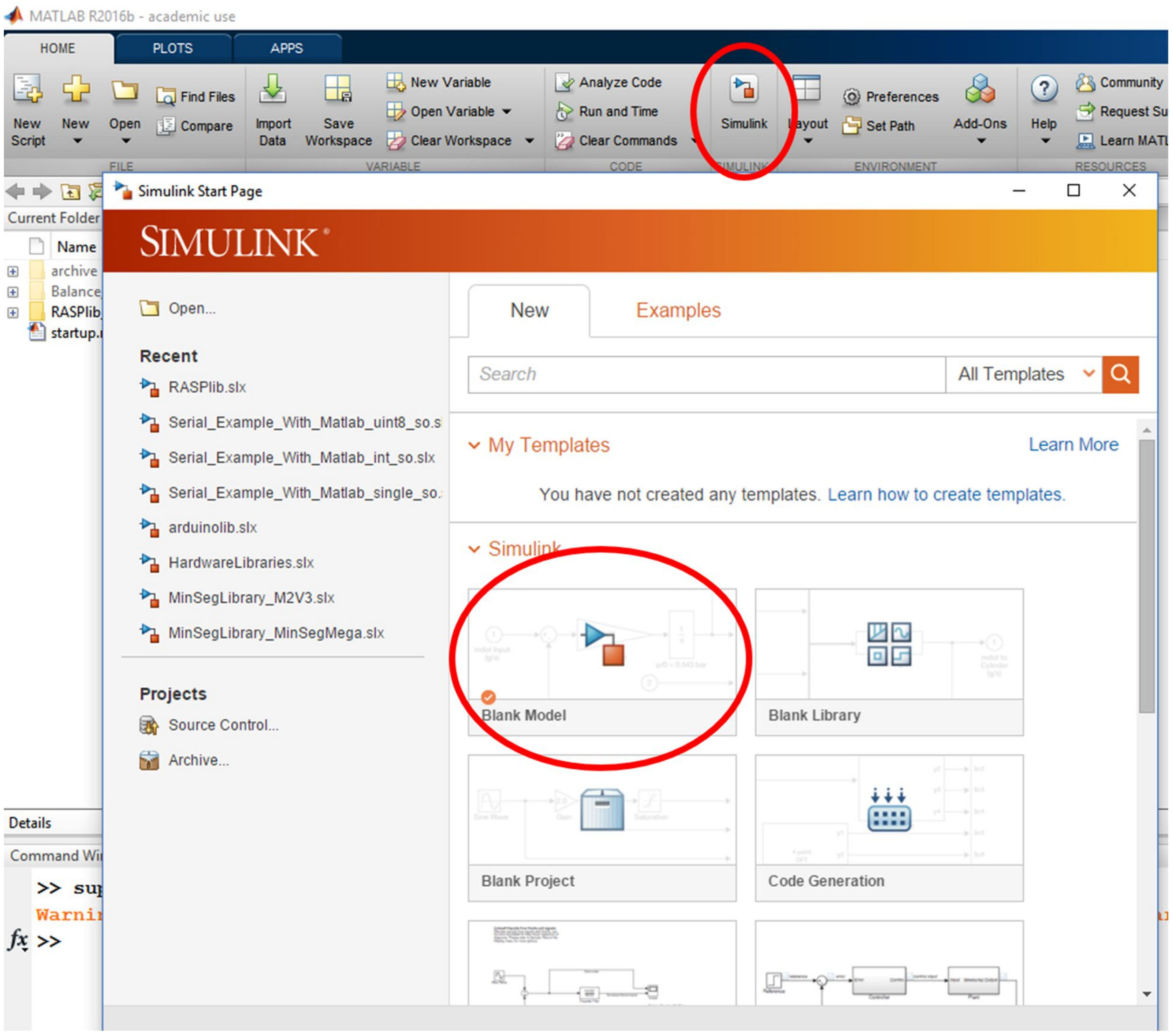
- Go to Device Manager and look for your Arduino Mega board (Arduino Mega 2560). Your Arduino board will be under Ports (COM & LPT).
- It may appear as “Arduino Mega”, “Prolific USB-to-Serial Comm..”, or “USB-SERIAL CH340” depending on your board. With the device manager open, unplug and plug in the USB cable to find which device appears.

Simulink Setup:

In order to ensure that your computer can properly communicate with the Arduino board, build and run the following Simulink diagram using the steps below:

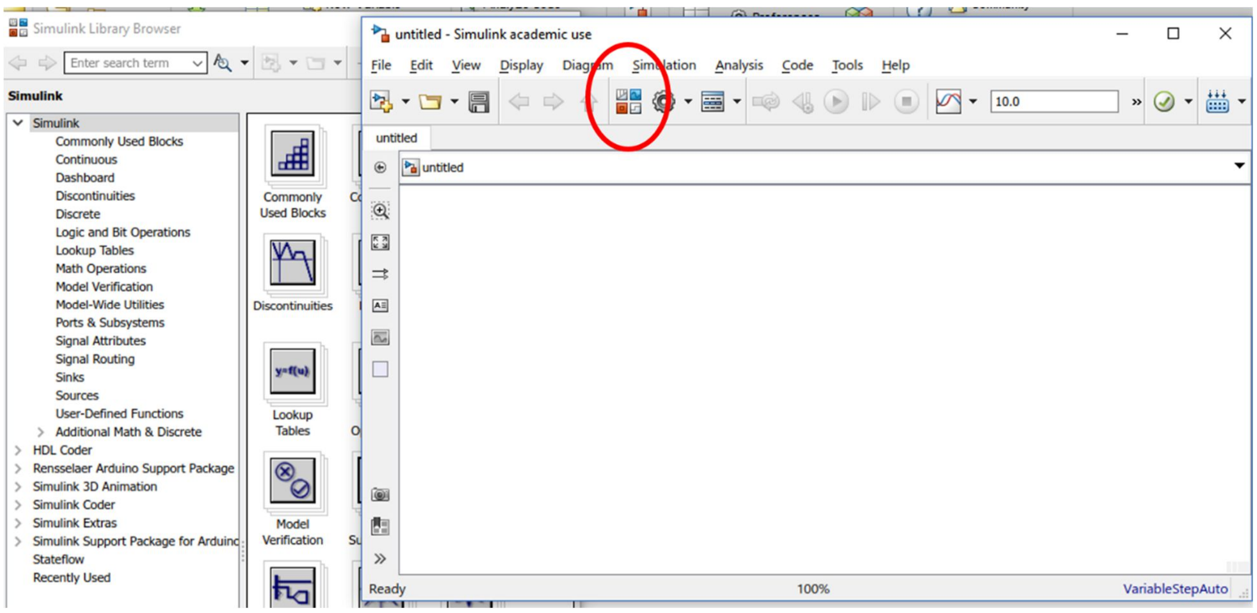


1) Open MATLAB then open a new Simulink model by clicking the Simulink icon at the top of the screen, then the “Blank Model” in the new window:



```
>> su
Warni
fx >>
```

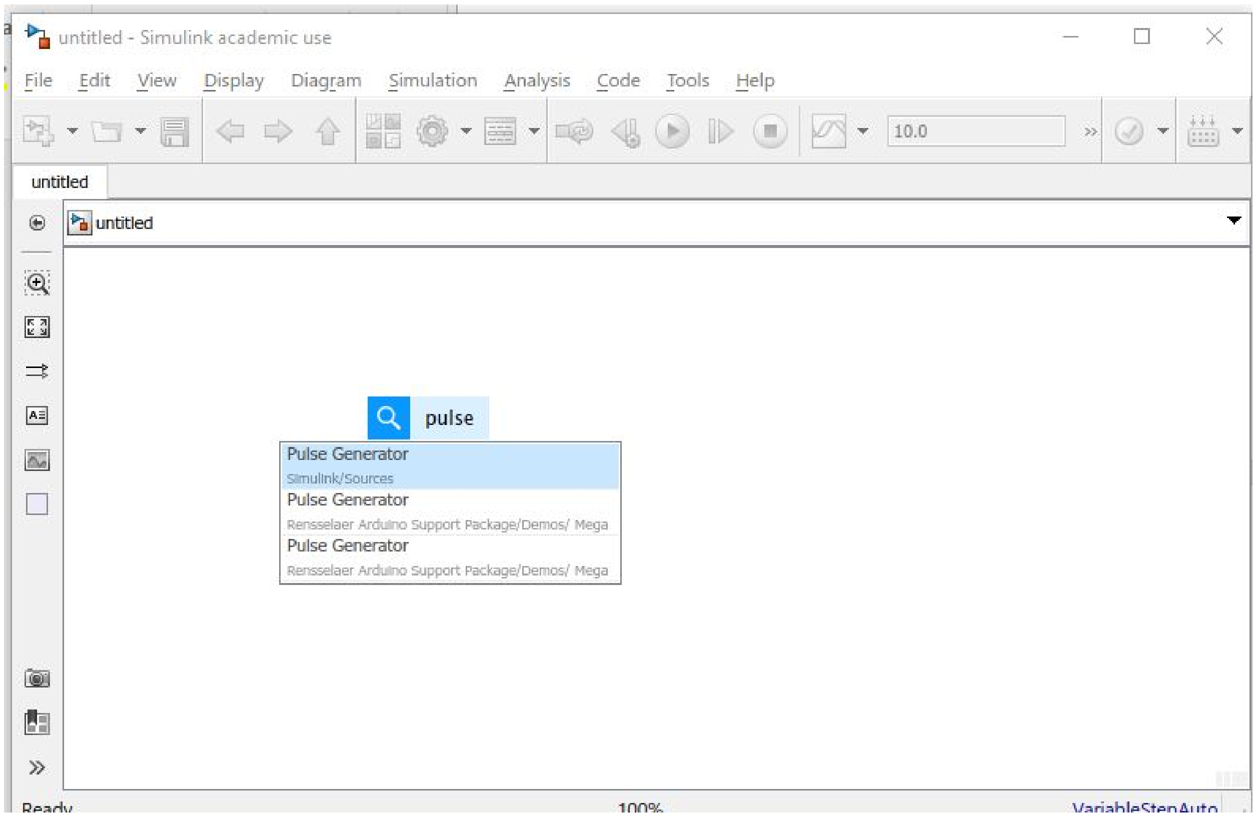
Open the Simulink browser from your new diagram:



2) Find and assemble the required blocks.

- Pulse Generator is under View->Library Browser->Simulink->Sources
- Digital Output is under View->Library Browser->Simulink Support Package for Arduino Hardware->Common

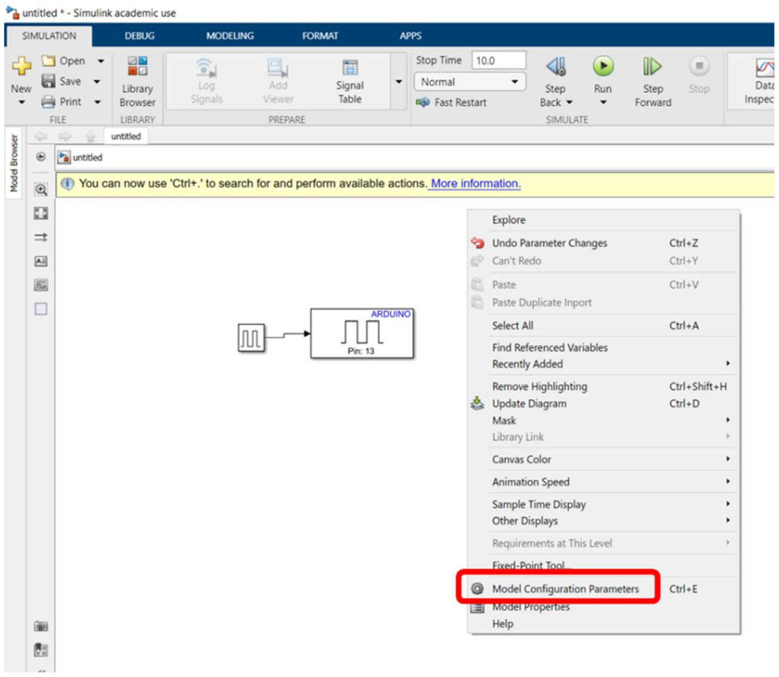
You can also search for the blocks by starting to type in the empty diagram for the block your looking for:


untitled - Simulink academic use
File Edit View Display Diagram Simulation Analysis Code Tools Help
10.0
untitled
pulse
Pulse Generator
Simulink/Sources
Pulse Generator
Rensselaer Arduino Support Package/Demos/ Mega
Pulse Generator
Rensselaer Arduino Support Package/Demos/ Mega
Ready
100%
VariableStepAuto

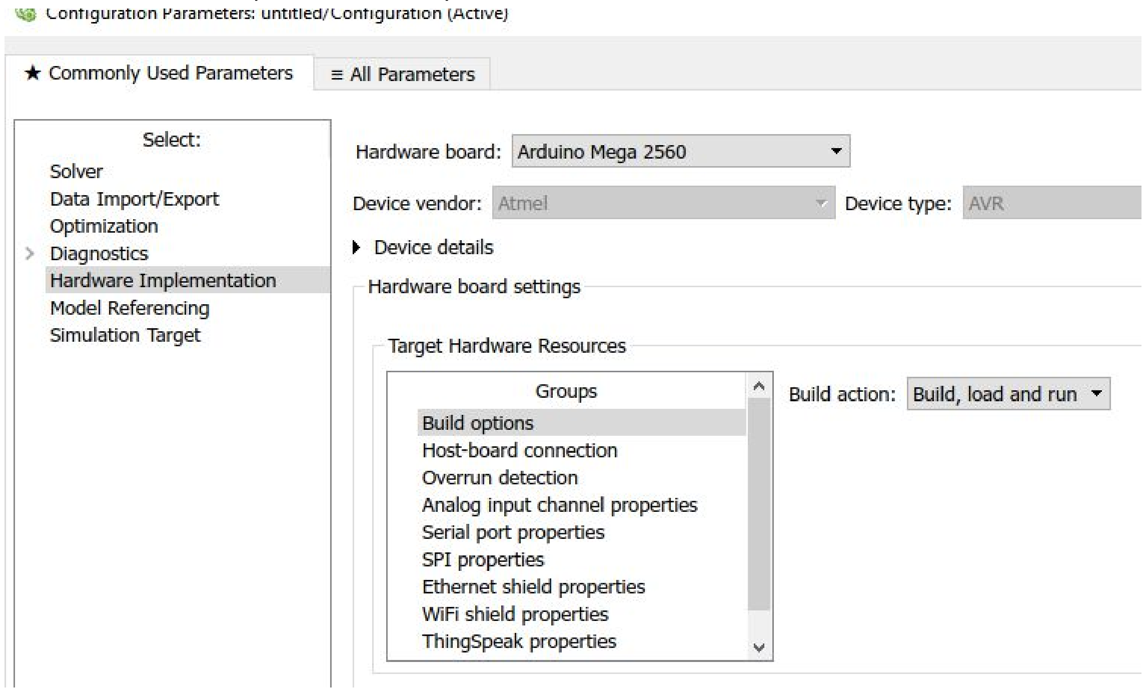
3) Change the output pin to 13, which is connected to the onboard LED. This is done by double clicking on the Digital Output block.

4) Double click on the Pulse Generator to change the amplitude to 1, the Period to 10, and the pulse width to 50%.

5) Right click in the Simulink model background to open Model Configuration Parameters:



- From the left-hand menu, pick “Hardware Implementation”. Select “Arduino Mega 2560” from the drop-down menu by “Hardware board”.


Configuration Parameters: untitled/Configuration (Active)

★ Commonly Used Parameters | ≡ All Parameters

Select:
Solver
Data Import/Export
Optimization
Diagnostics
Hardware Implementation
Model Referencing
Simulation Target

Hardware board: Arduino Mega 2560

Device vendor: Atmel | Device type: AVR

Device details

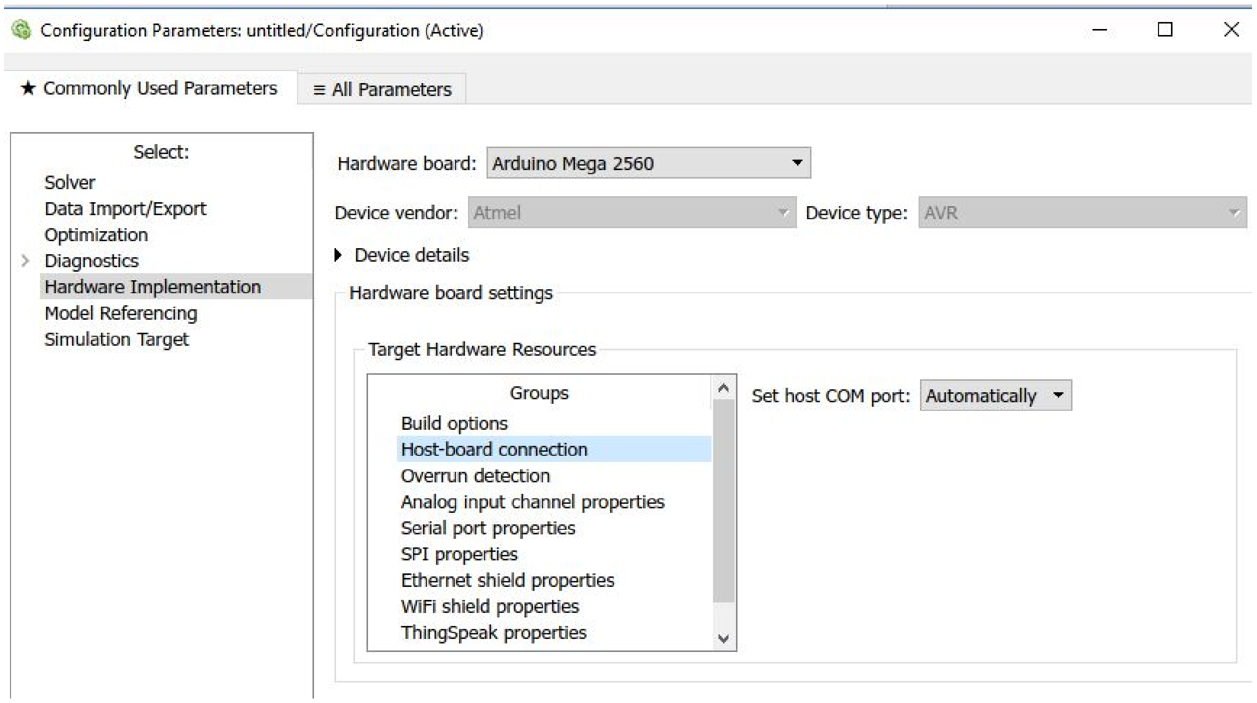
Hardware board settings

Target Hardware Resources

Groups
Build options
Host-board connection
Overrun detection
Analog input channel properties
Serial port properties
SPI properties
Ethernet shield properties
WiFi shield properties
ThingSpeak properties

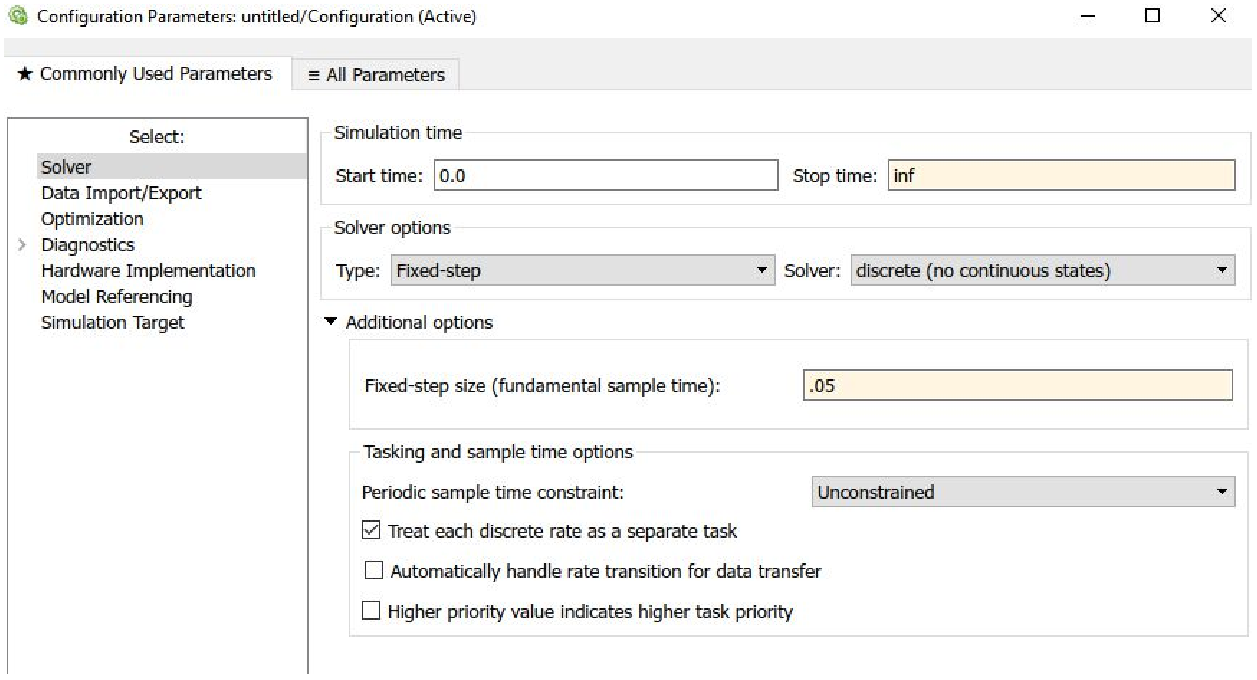
Build action: Build, load and run

- Under “Target Hardware Resources” and “Host-board connection“, set host COM port: should be set to “Automatically”, click OK. You should pecify the COM port “Manually” here if your board was "Prolific USB-to-Serial Comm..", or "USB-SERIAL CH340" instead of "Arduino Mega".



Hardware board: Arduino Mega 2560
Device vendor: Atmel
Device type: AVR
Set host COM port: Automatically

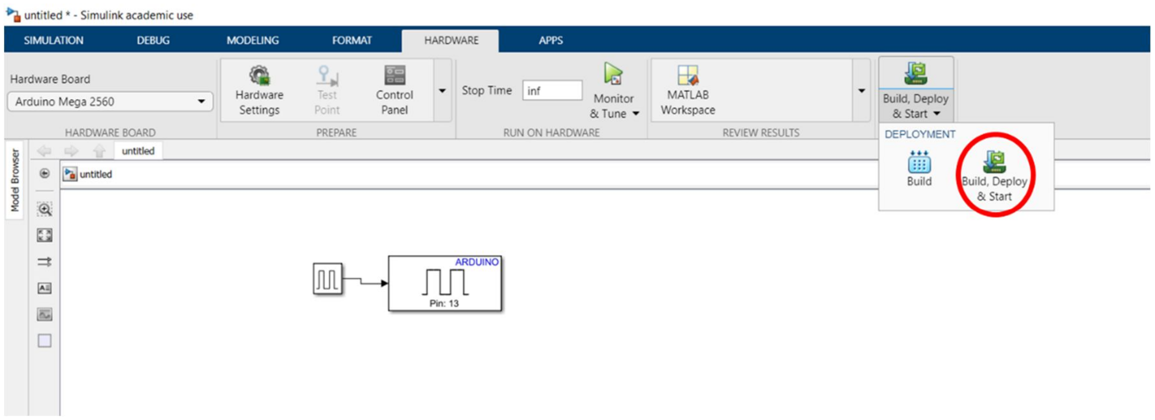
- Go to 'Solver' on the left hand menu and type 'inf' into the 'Stop time' spot so that the model will keep running. Make sure the solver options are set to "Fixed-step" and "discrete (no continuous states)". The "Fixed-step size" should be 0.05. This is how fast in seconds the control loop will execute. In this case, every 50 milliseconds it will execute the Simulink code.



Start time: 0.0
Stop time: inf
Type: Fixed-step
Solver: discrete (no continuous states)
Fixed-step size (fundamental sample time): .05
Periodic sample time constraint: Unconstrained
☑ Treat each discrete rate as a separate task
☐ Automatically handle rate transition for data transfer
☐ Higher priority value indicates higher task priority

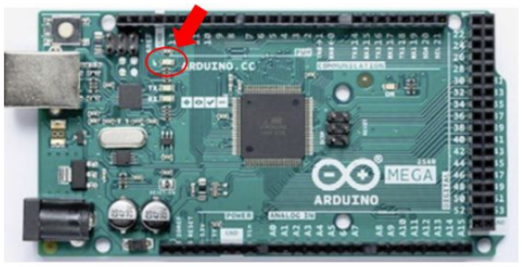
6) Compile and download the code to the board:

- Under the "HARDWARE" tab, choose "Build, Deploy, & Start", or the keystroke Ctrl+B:



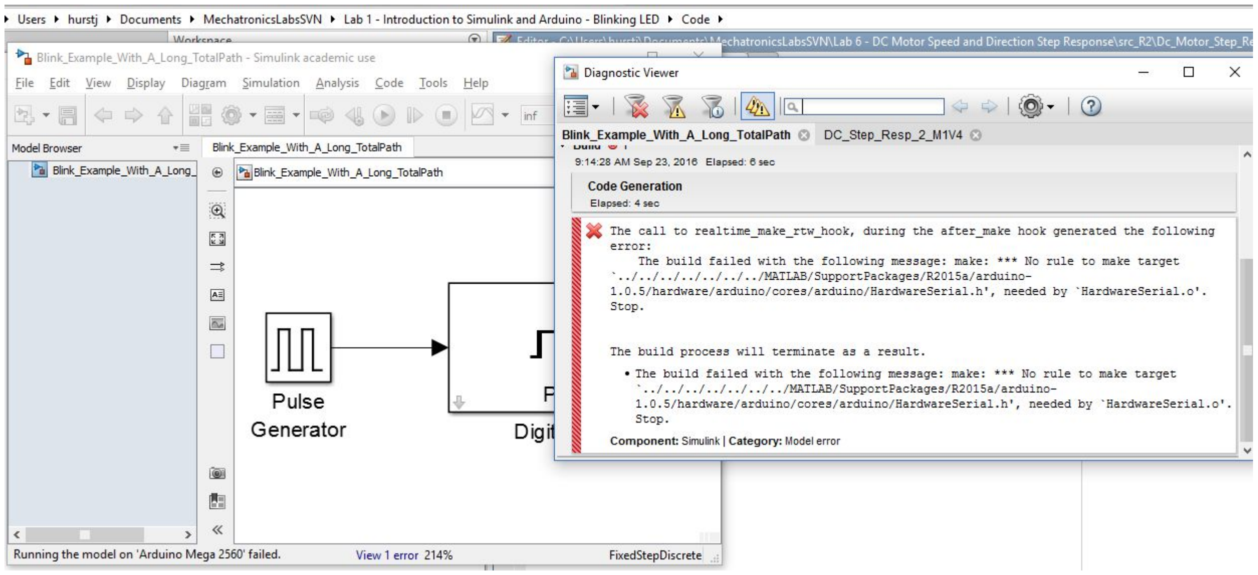
This will download the code to the board and **will not interact or connect with it.** In this “Normal” Mode the code is downloaded to the board, and no information is passed between it and your computer unless your program specifically instructs the board to do so.

At this point the onboard LED connected to pin 13 should start blinking on and off every 5 seconds as prescribed by the pulse generator block. This LED is on the lower board towards the USB connector If not, check the troubleshooting section below.



Troubleshooting

- If an error occurs indicating that there is no driver, follow the directions on the screen.
- If MATLAB cannot find the board find COM port number from the device manager and in ‘Configuration Parameters’ set the port manually to the correct number.
    - Also, try unplugging the Arduino USB cable and plugging it back in.
    - Can also try restarting MATLAB
- Some errors appear on the MATLAB workspace screen - if something is not working, check here – the error messages will indicate the problem
- If you get a code generation error like below:


Users ▸ hurstj ▸ Documents ▸ MechatronicsLabsSVN ▸ Lab 1 - Introduction to Simulink and Arduino - Blinking LED ▸ Code ▸
Blink_Example_With_A_Long_TotalPath - Simulink academic use
File Edit View Display Diagram Simulation Analysis Code Tools Help
Diagnostic Viewer
Blink_Example_With_A_Long_TotalPath
DC_Step_Resp_2_M1V4
9:14:28 AM Sep 23, 2016 Elapsed: 6 sec
Code Generation
Elapsed: 4 sec
Pulse Generator
Digit
View 1 error 214%
FixedStepDiscrete

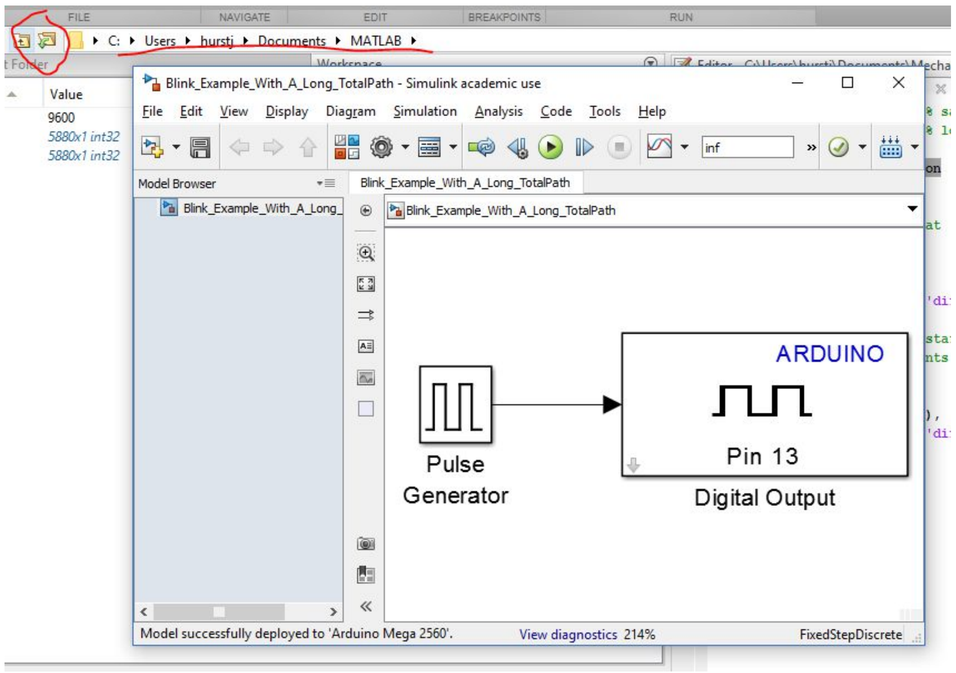
```
The call to realtime_make_rtw_hook, during the after_make hook generated the following
error:
    The build failed with the following message: make: *** No rule to make target
`../../../../../../MATLAB/SupportPackages/R2015a/arduino-
1.0.5/hardware/arduino/cores/arduino/HardwareSerial.h', needed by `HardwareSerial.o'.
Stop.


The build process will terminate as a result.
  • The build failed with the following message: make: *** No rule to make target
    `../../../../../../MATLAB/SupportPackages/R2015a/arduino-
    1.0.5/hardware/arduino/cores/arduino/HardwareSerial.h', needed by `HardwareSerial.o'.
    Stop.
Component: Simulink | Category: Model error
```

Running the model on 'Arduino Mega 2560' failed.

This is because MATLAB is compiling in a directory where the directory and filename it too long. In the example above the compilation path is:
C:\Users\hurstj\Documents\MechatronicsLabsSVN\Lab 1 - Introduction to Simulink and Arduino - Blinking LED\Code\Blink_Example_With_A_Long_TotalPath.slx

To remedy this you can try a shorter file name, or simply change (point) MATLAB to a shorter directory to compile in like:


FILE
NAVIGATE
EDIT
BREAKPOINTS
RUN
C: ▸ Users ▸ hurstj ▸ Documents ▸ MATLAB ▸
Value
9600
5880x1 int32
5880x1 int32
Blink_Example_With_A_Long_TotalPath - Simulink academic use
File Edit View Display Diagram Simulation Analysis Code Tools Help
inf
Model Browser
Blink_Example_With_A_Long_TotalPath
ARDUINO
Pin 13
Pulse Generator
Digital Output
View diagnostics 214%
FixedStepDiscrete

Model successfully deployed to 'Arduino Mega 2560'.

It is the same file, but MATLAB will now compile everything in C:\Users\hurstj\Documents\MATLAB so the compiler won't have a path problem.

- If your file and your compile directory are on a different drive for example D: instead of the installation drive for MATLAB, you also might get a compile error. In this case you will want to try and move the file and the compile folder to the MyDocuments/MATLAB folder that is created when MATLAB installs.

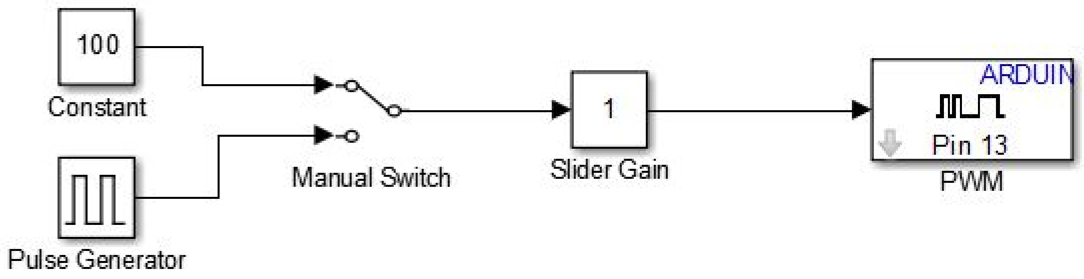
Part 2: Communicating with External Mode

Objective

- Communicate with the target board (Arduino) using external mode by changing the brightness of an LED with PWM

Simulink

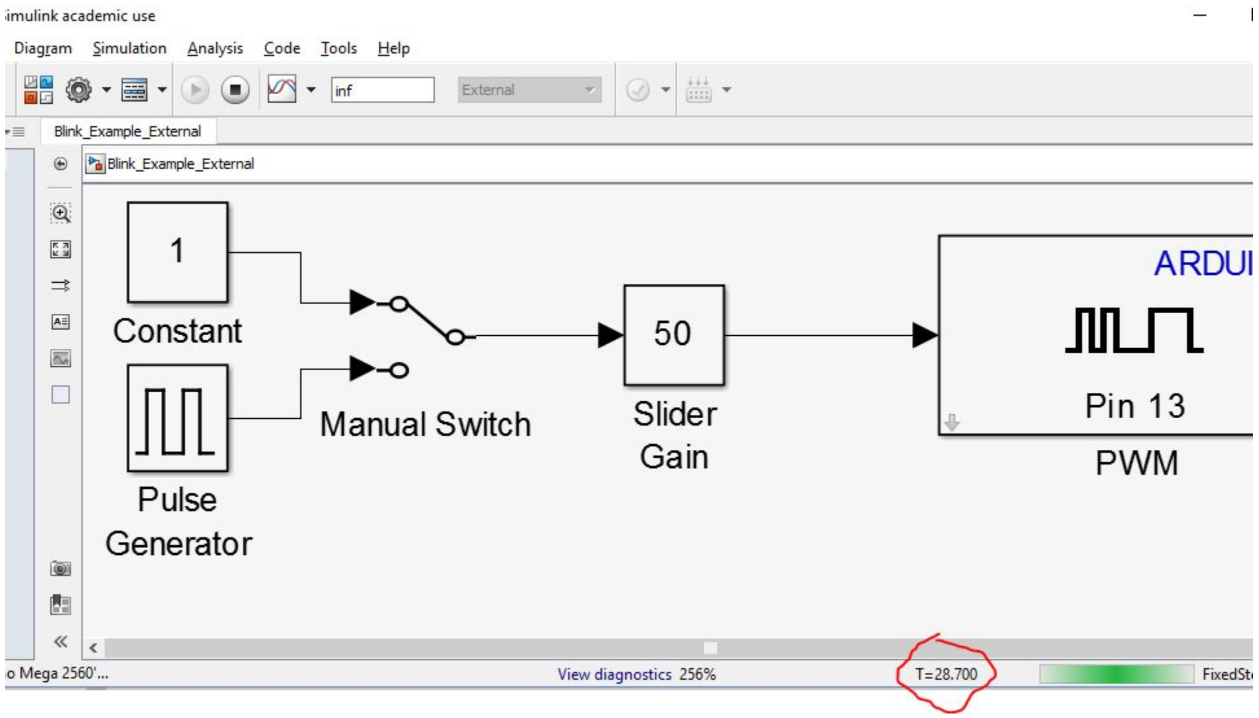
Modify your Simulink diagram to the following



- Pin 13 can also be set as a PWM instead of just an on/off digital output – replace the Digital Output block with the PWM block
- The “Slider Gain” can be found in View->Library Browser->Simulink-> Math Operations. Change the high value to 255 (the max value of the PWM)
- The “Manual Switch” can be found in View->Library Browser->Simulink ->Signal Routing
- The “Constant” can be found in View->Library Browser->Simulink -> “Commonly Used Blocks”

External Mode

- **External mode,** enabled by using the “Monitor & Tune” button, **allows bi-directional communication** to/from the application board to the PC, but can limit how fast your code can execute.
    - The “**Deploy, Build, & Start**” buttom downloads the code and **does not connect or interact with it,** it can allow your code to run faster
    - When external mode is selected it downloads additional code to the board to allow this communication. This does increase the program size (requires more memory)
- When you use external mode, you can change parameters while the system is running
    - However, this has an impact on the performance due to the communication bandwidth.
    - You can tell how fast you can run your code by examining the running clock in the Simulink diagram after you press the play button. When the code is running the green bar will be flashing and the time in seconds should be advancing:

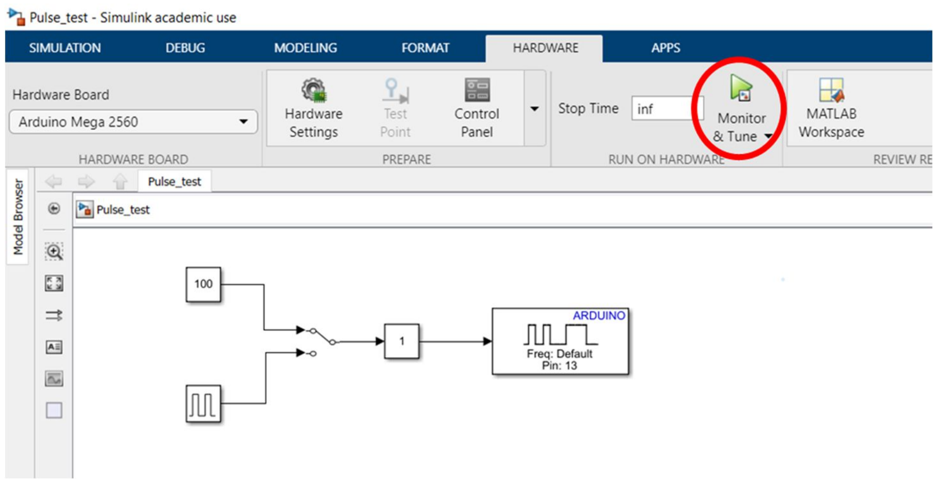


If this clock advances as fast as the normal real word clock your code is "keeping up". If, for example, it can take 1 minute for the this timer to reach 10 seconds your code is not keeping up with real time and your requested sample time is to fast – it can not compute everything it needs to do fast enough.

- The values change on the Slider Gain and the Manual Switch can be changed while the program is running.
- If necessary the code can be run faster if external mode is not used, by using "Deploy, Build, & Start" button instead of the "Monitor & Tune" button.

Enabling External Mode:

- Under the "HARDWARE" tab, select "Monitor & Tune":



Run the code and experiment with the setup. Observe the effects on the LED when changing the value of the Slider Gain, Manual Switch, and the Pulse Generator.

- While running the simulation, you can change the position of the switch by double clicking on the switch
- You can change the gain of the Slider Gain while running the simulation by double clicking it.
- Adjusting the Pulse Generator, experiment with all three parameters. Which increases brightness? What is the approximate range over which you can observe a difference? [Note: These parameters will be addressed in a later lab on PWM]

Stopping the Simulation:

When you are running the simulation in external mode you are connected to the device and are sending data and receiving data.

  - Use the green "Monitor & Tune" button to download and run the code in external mode
  - Use the square "Stop" button to stop. Since the board is no longer connected it will continue doing the last action before it was disconnected.

Debugging – cannot connect to device or cannot download code

- If you get an error about "Error occurred while executing External Mode MEX-file 'ext_serial_win32_comm':Timed-out waiting for first connect response packet.", run the following command at the MATLAB prompt:

  ```
  >> codertarget.arduinobase.registry.setBaudRate(gcs,115200)
  ```

- If you cannot connect to the device or there is an error when trying to download the code usually this means external mode was enabled and the simulation was stopped with the stop button or the USB cable was unplugged while being connected to the device. In this case you have a couple things to try that may successfully close the serial port
  - Disconnect and reconnect the USB Cable

- Try the following commands at the Matlab command line (find connected instruments and close them):
  - newobjs=instrfindall
  - fclose(newobjs)
  - delete(newobjs)
- Reset the device with the reset button or disconnect/connect the serial cable (this sometimes works)
- Close and restart Matlab (this usually works)
- Log off your machine, log back in, then restart Matlab (this almost always works)

- Make sure the COM port is correct in the "Model Configuration Parameters"

Checkpoint:

To complete Lab 1 you will need to show the ability to control the blink of the LED in one program with both the 'manual switch' and the 'slider gain'.

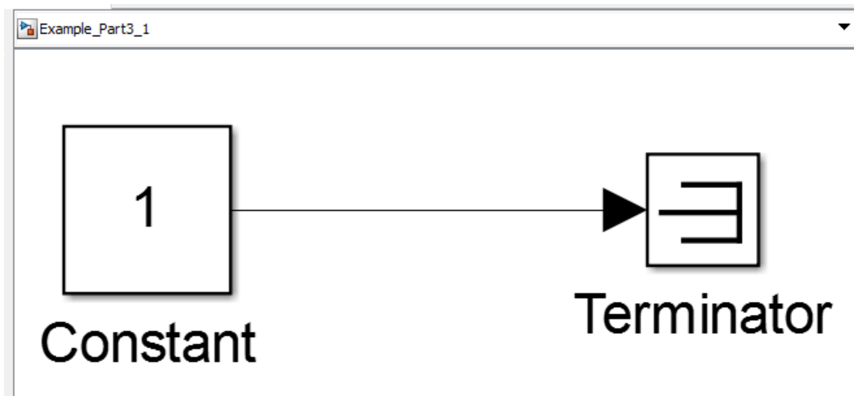
Part 3 (optional): Evaluating code execution speed – how fast can I run my code

Objective:

- Determine how fast your code can be run on the target hardware using "Enable overrun detection" in Simulink.
- This can be used in both External and Normal mode to ensure your code is executing in time.

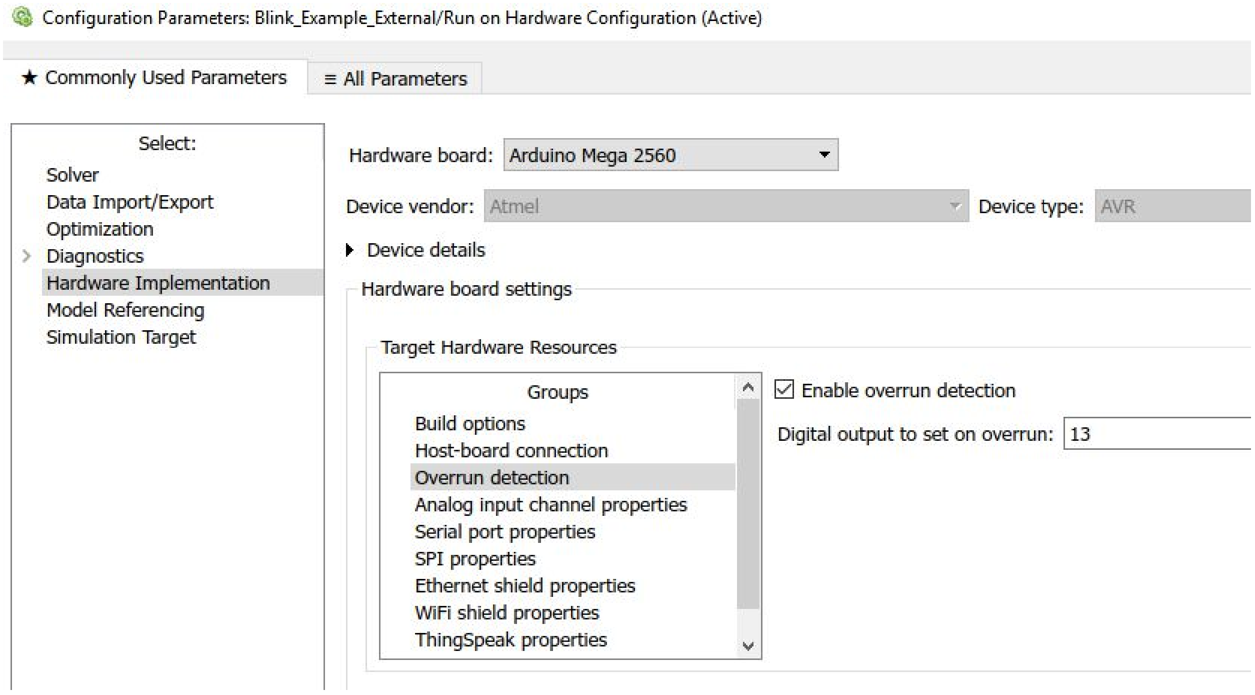
Simulink:

Build the following model:



Note that this code essentially does nothing. The terminator is also in the 'commonly used blocks'. To determine how fast the hardware can execute a simple loop, use "Enable overrun detection" on pin 13:

Right click in the Simulink model background to open Model Configuration Parameters. In the window that pops up, select "Hardware Implementation" from the left-hand menu, and expand "Hardware board settings”. You will need to select Arduino Mega 2560 in the drop down menu at the top. Under Target Hardware Resources and Overrun detection, check “Enable overrun detection” and set the pin to 13.


Configuration Parameters: Blink_Example_External/Run on Hardware Configuration (Active)

Hardware board: Arduino Mega 2560

Device vendor: Atmel

Device type: AVR

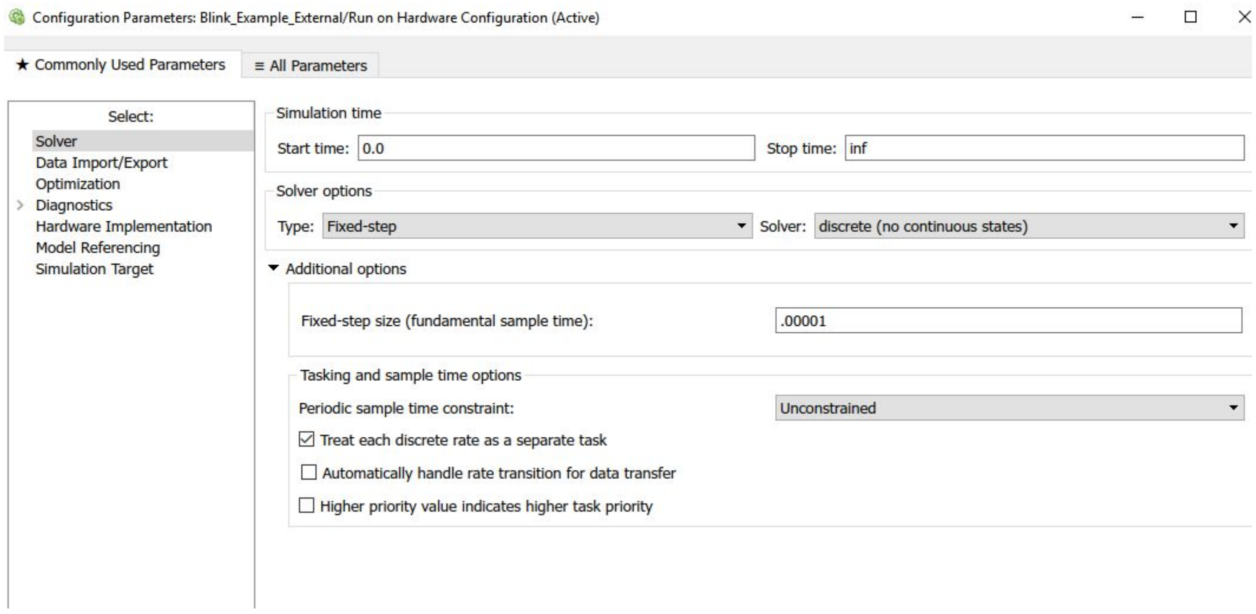
Hardware board settings

Target Hardware Resources

☑ Enable overrun detection

Digital output to set on overrun: 13

If the hardware cannot execute your code in the same time you specify the digital output, in this case the LED on pin 13 will light up indicating that you code cannot be executed in the time you have specified. You can use any digital output, but pin 13 is conveniently connected to an LED on our board. For the example code above, start with a sample time of 0.01 milliseconds (.00001 seconds). Download the code with the deploy to hardware button (Normal mode, not external mode) and observe if LED 13 on the board lights up after downloading:


Configuration Parameters: Blink_Example_External/Run on Hardware Configuration (Active)

★ Commonly Used Parameters | ≡ All Parameters

Select:
- Solver
- Data Import/Export
- Optimization
- Diagnostics
- Hardware Implementation
- Model Referencing
- Simulation Target

Simulation time

Start time: 0.0 Stop time: inf

Solver options

Type: Fixed-step Solver: discrete (no continuous states)

Additional options

Fixed-step size (fundamental sample time): .00001

Tasking and sample time options

Periodic sample time constraint: Unconstrained

☑ Treat each discrete rate as a separate task

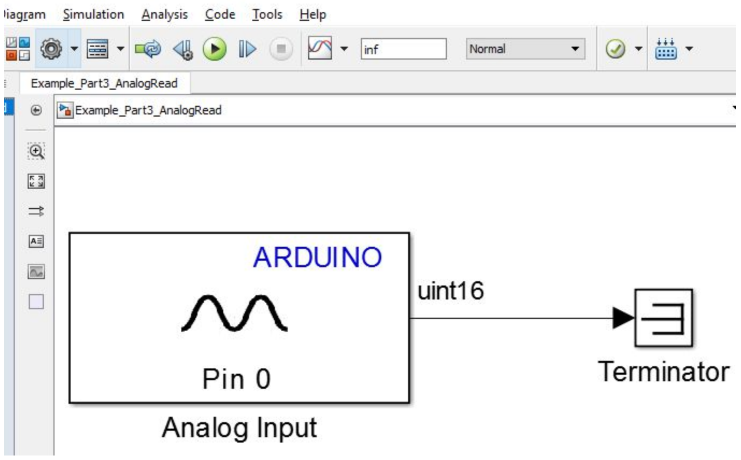
☐ Automatically handle rate transition for data transfer

☐ Higher priority value indicates higher task priority

You should notice LED 13 will light up. This indicates the microprocessor can not run the code every .00001 second, so you have to specify a larger sample time

Keep increasing the sample time (try .02 milliseconds, etc.) until LED 13 does not light up after downloading. This is the fastest you the board can execute your code.

Using this procedure try and see how fast you can run the following, start at a sample time of 0.01 milliseconds.



In this fashion, you can determine how fast your code, or any part of your code takes to run on your hardware.

Questions:

How can you determine how fast you can run code in External mode?

How can you determine how fast your code can run in Normal mode?

What is the difference between Normal mode and External mode?