Lab 1: Managing Sensors and Actuators with Raspberry Pi & Arduino using Wyliodrin Studio (1)

物聯網技術與應用(英) IoT/M2M Technologies and Applications

國立交通大學資訊工程系
Department of Computer Science
National Chiao Tung University

September 21, 2018

Outline

- Wyliodrin Studio Overview
  - Setup and Starting Wyliodrin Studio
  - Wyliodrin Studio GUI *(Checkpoint 1)*
- Raspberry and Arduino Basics
- Hello World Application *(Checkpoint 2)*
- Traffic Light Simulator Application
  - Assembling Schematics *(Checkpoint 3)*
  - Coding with Blocks *(Checkpoint 4)*

WYLIODRIN STUDIO OVERVIEW

Wyliodrin Studio Overview (1)

- Wyliodrin Studio is an Open Source, local version of Wyliodrin (https://www.wyliodrin.com/).
- Wyliodrin STUDIO is an standalone IDE for software and hardware development for IoT and Embedded Linux systems.
- You can program and monitor the boards which are in the same network or are directly connected to your computer.

Wyliodrin Studio Overview (2)

- Connect to devices using TCP/IP or serial port
- Develop software and firmware for IoT in several programming languages
- Shell access to the device
- Import and export Wyliodrin STUDIO projects
- Visual dashboard for displaying sensor data
- Display the hardware schematics
- Manage packages for Python and Javascript
- Task manager for managing the device
- Network connection manager for the device (Ethernet and WiFi)
- Interactive electronics documentation (resistor color code)
- Example projects and firmware
- Wyliodrin API documentation in C/C++, Python and Javascript

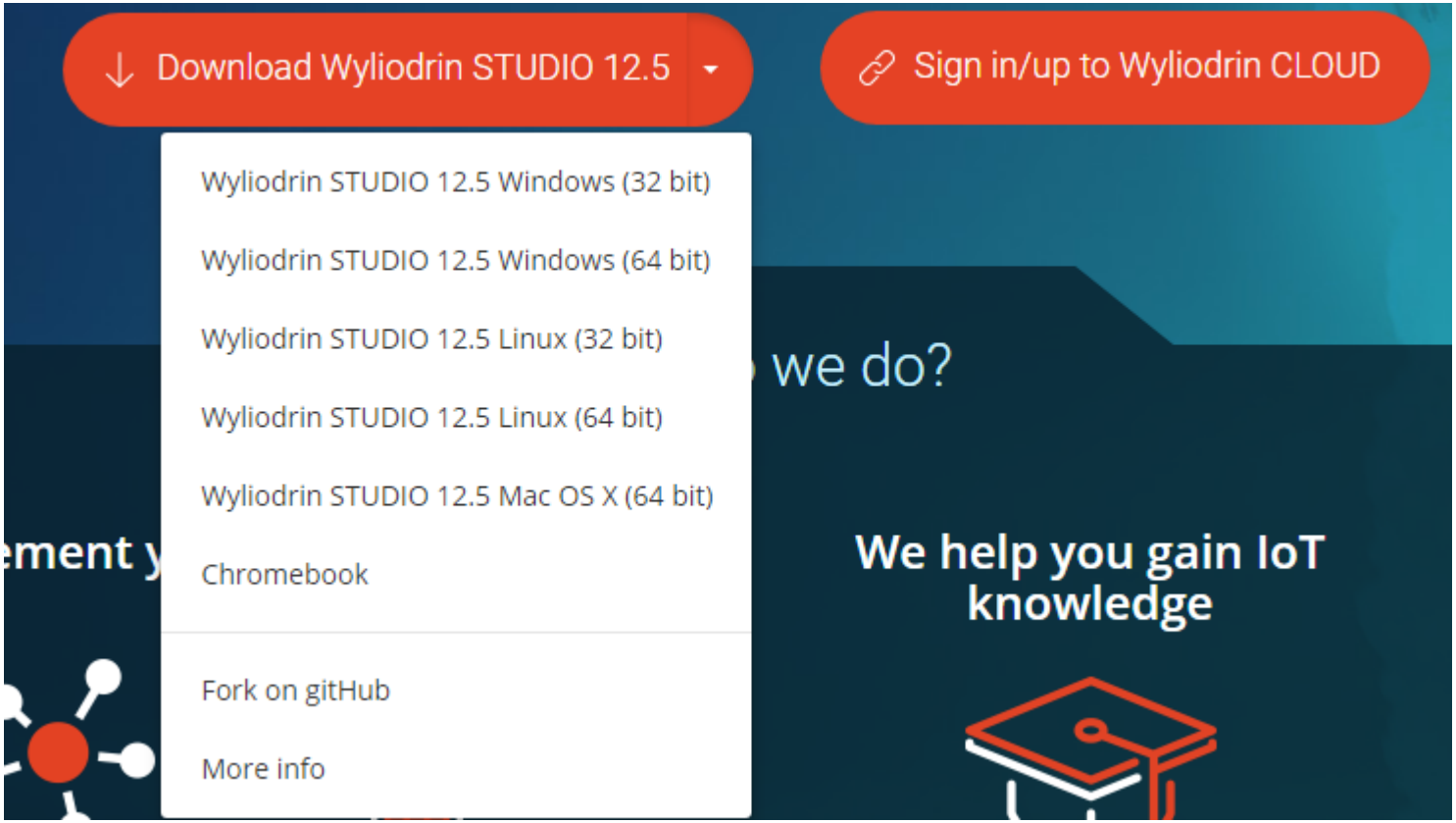
Wyliodrin Studio Overview (3)

- **Supported devices**
  - UDOO Neo
  - **Raspberry Pi and Arduino**
  - BeagleBone Black
  - **Arduino**

- **Supported languages**
  - **Visual Programming (translates to Python)**
  - Javascript
  - Python
  - Shell Script (bash)

Wyliodrin Studio Setup

https://www.wyliodrin.com/



Attention!! Wyliodrin Studio is already installed in the “IoTClass” VM. You don’t need to execute these steps.

Use your VM from now on…

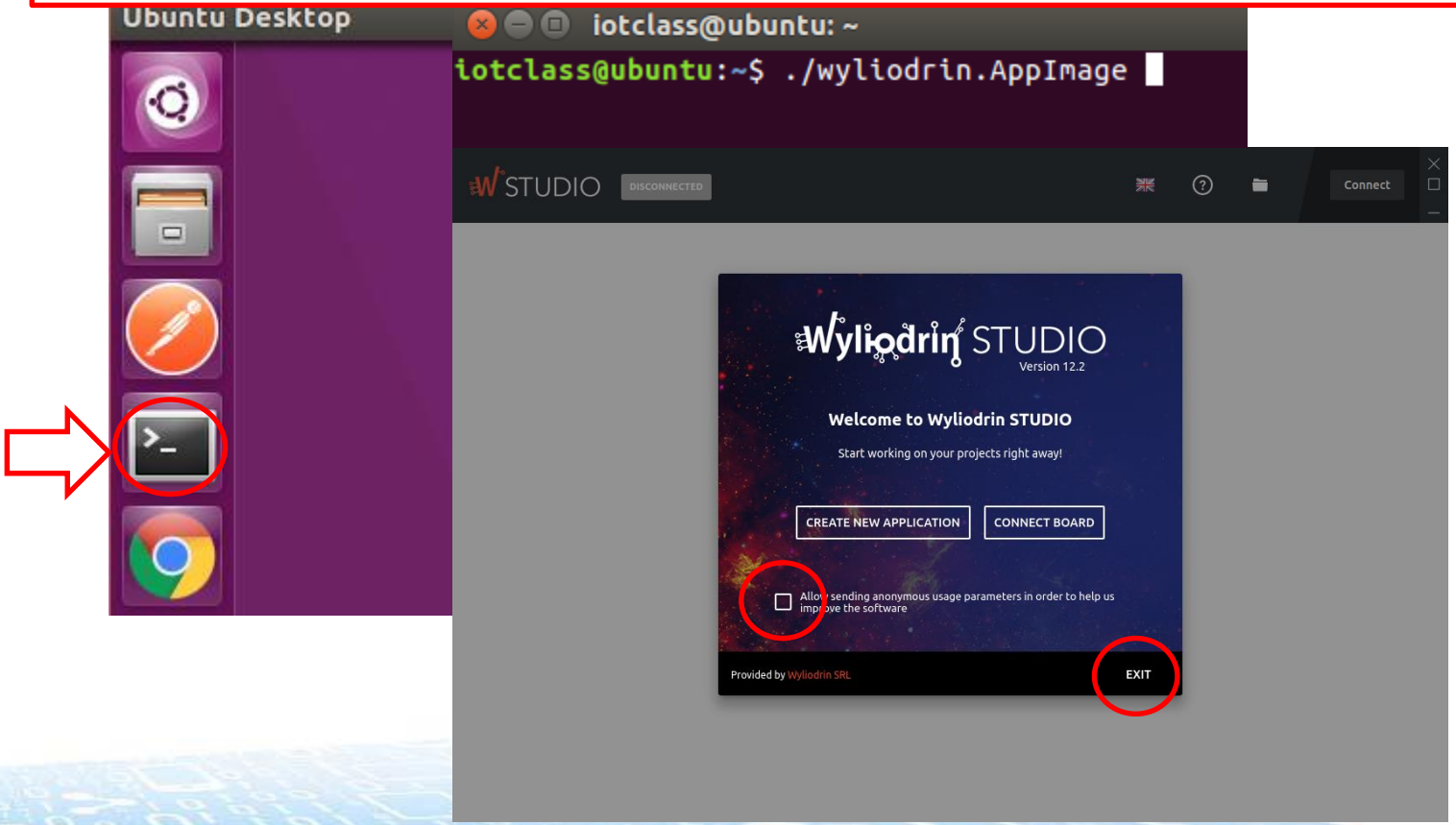
The steps showed in the coming slides have to be performed in your VM.

Username = iotclass

Password = iotclass

Starting Wyliodrin Studio

1. Open a Terminal window.
2. Type ./wyliodrin.AppImage and press ENTER.
3. Attention!! Uncheck “Allow sending anonymous” and Click Exit (If needed). Otherwise you will see just an empty window, which is correct!



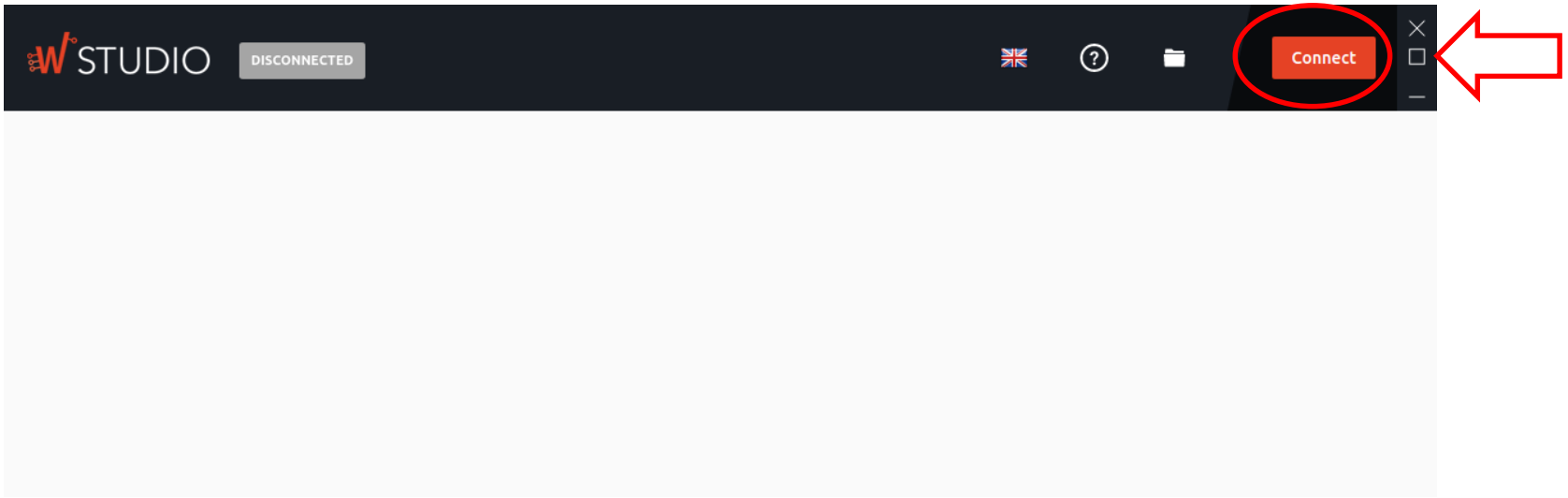
Connect the Raspberry Pi to the Power Source

Connect your Raspberry Pi to the Power Source Now!

Connecting a Raspberry Pi to Wyliodrin Studio (1)

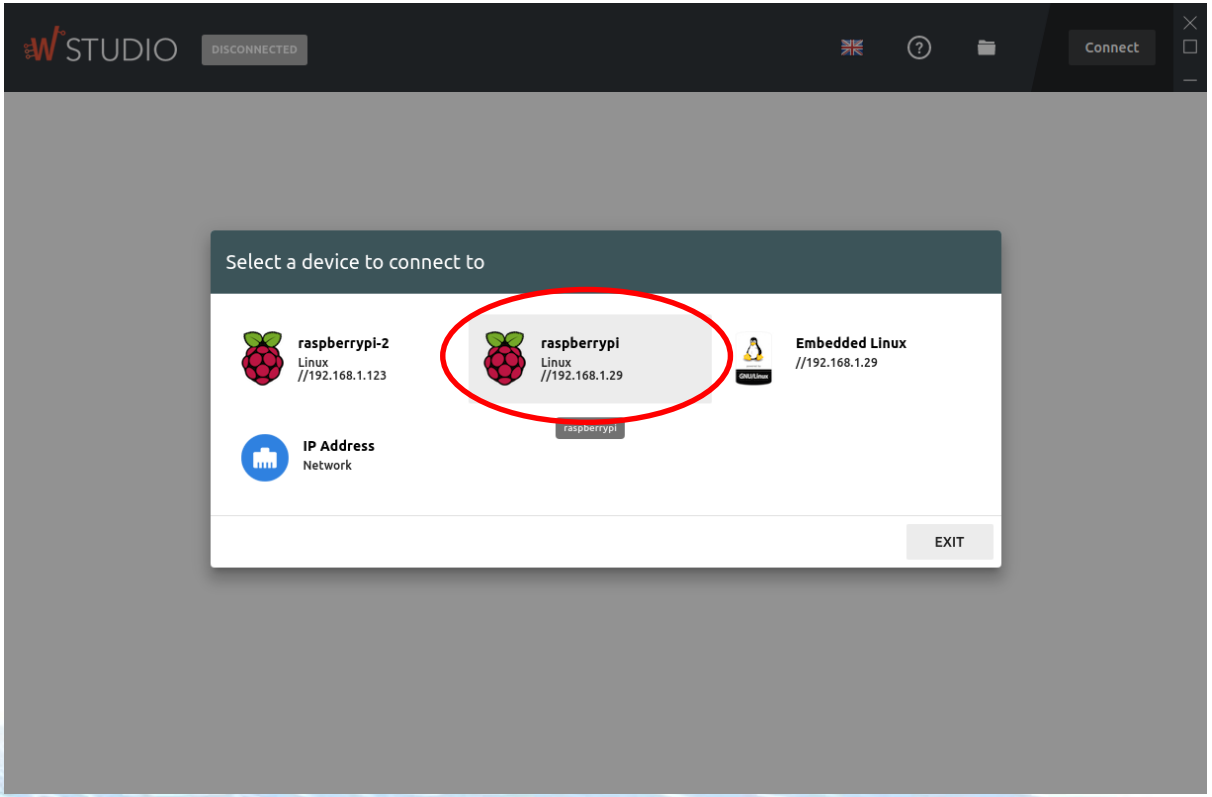
In order to see all the options in the GUI, we need to connect Wyliodrin with a board. The following steps are used to connect Wyliodrin with a Raspberry Pi.

1. Click on “Connect”.



Connecting a Raspberry Pi to Wyliodrin Studio (2)

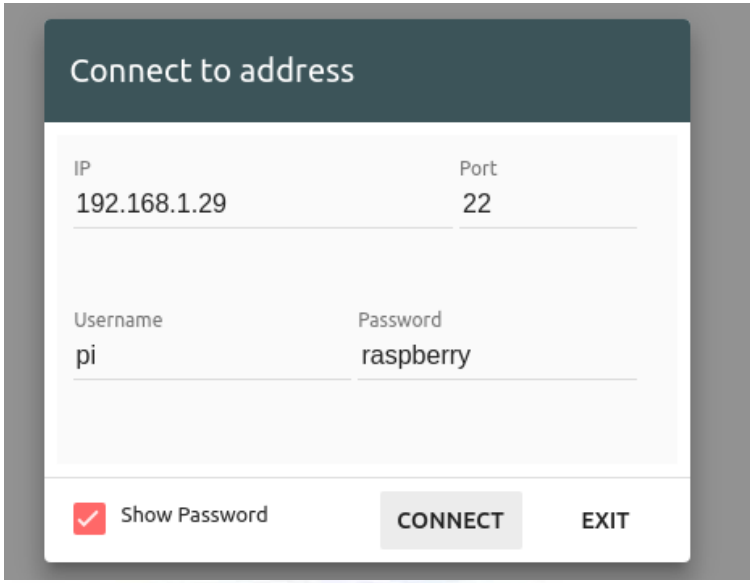
2. Click on the Raspberry Pi **ASSIGNED TO YOUR TEAM**.
   Use the IP address instructed by TA as guide.
   (**Ask TA if you are not sure which Pi belongs to your team**).



Connecting a Raspberry Pi to Wyliodrin Studio (3)

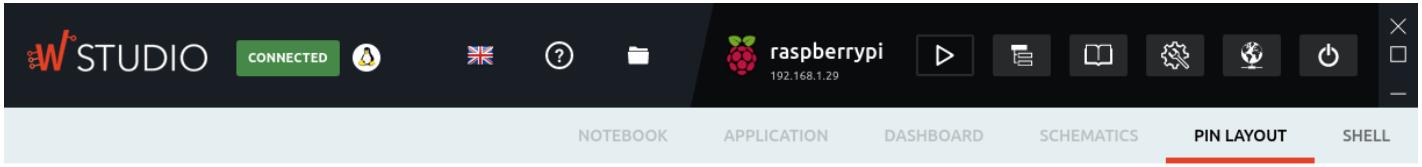
6. Input the following credentials:
   Username = pi
   Password = raspberry

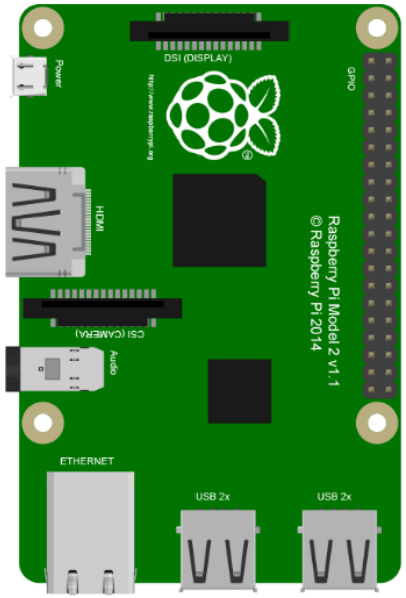
7. Click on “Connect”.



Connecting a Raspberry Pi to Wyliodrin Studio (4)

You should see now all the features that Wyliodrin Studio GUI offers.

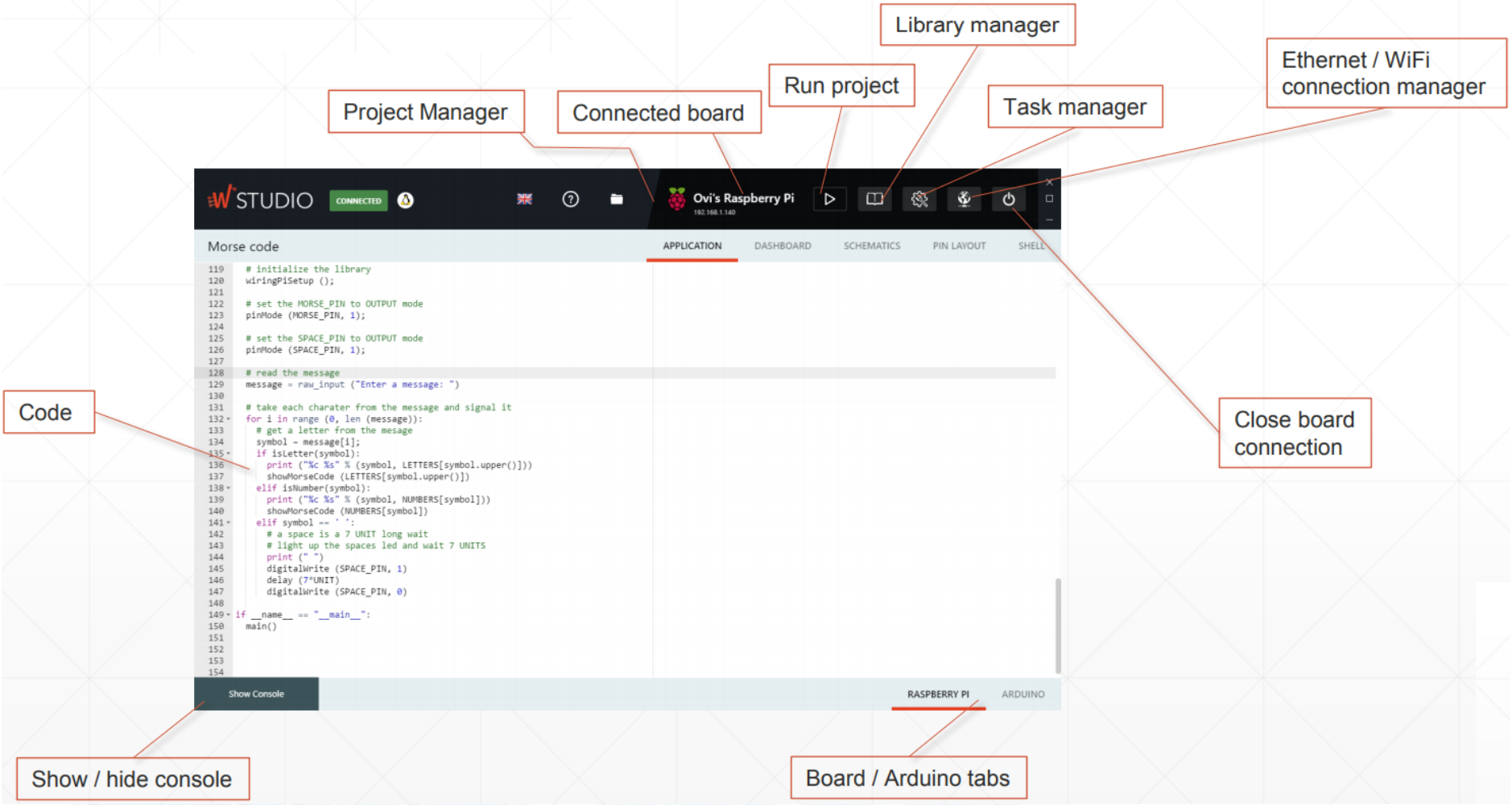




Raspberry Pi 2 Model B (J8 Header) pinout

WiringPi	BCM(Name)	Physical		Physical	BCM(Name)	WiringPi
	3v3 Power	1	2	5v Power		
8	BCM 2 (SDA)	3	4	5v Power		
9	BCM 3 (SCL)	5	6	Ground		
7	BCM 4 (GPCLK0)	7	8	BCM 14 (TXD)	15	
	Ground	9	10	BCM 15 (RXD)	16	
0	BCM 17	11	12	BCM 18 (PCM_C)	1	
2	BCM 27 (PCM_D)	13	14	Ground		
3	BCM 22	15	16	BCM 23	4	
	3v3 Power	17	18	BCM 24	5	
12	BCM 10 (MOSI)	19	20	Ground		
13	BCM 9 (MISO)	21	22	BCM 25	6	
14	BCM 11 (SCLK)	23	24	BCM 8 (CE0)	10	
	Ground	25	26	BCM 7 (CE1)	11	
	BCM 0 (ID_SD)	27	28	BCM 1 (ID_SC)		
21	BCM 5	29	30	Ground		
22	BCM 6	31	32	BCM 12	26	
23	BCM 13	33	34	Ground		
24	BCM 19 (MISO)	35	36	BCM 16	27	
25	BCM 26	37	38	BCM 20 (MOSI)	28	
	Ground	39	40	BCM 21 (SCLK)	29	

Wyliodrin Studio GUI (1)



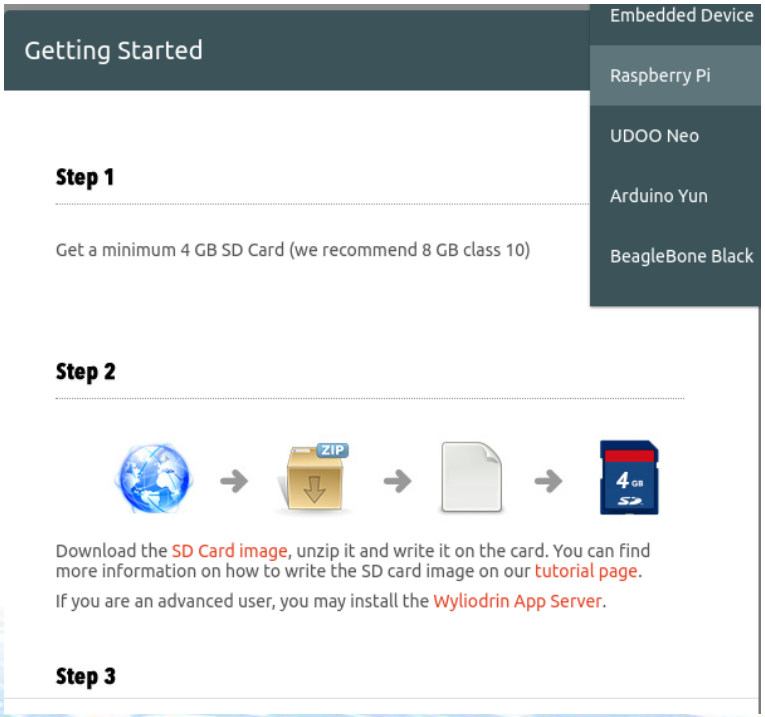
Wyliodrin Studio GUI (2)

Board Connection

- Direct connection (Serial communication).
- Remote connection (Uses mDNS to discover devices in the same network).

Additional Software

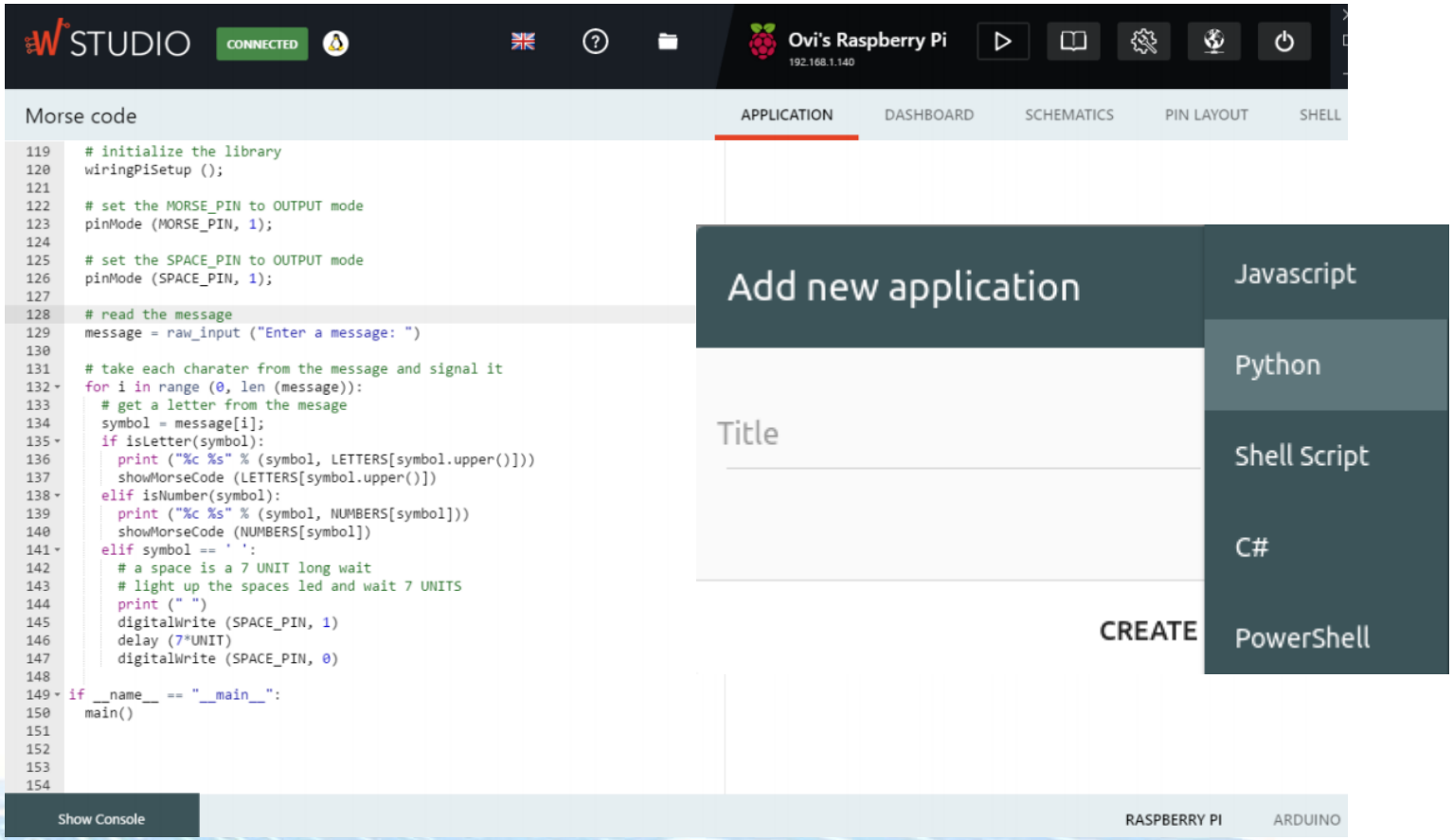
- Any board needs to run an instance of wyliodrin-server.
- Download the SD card image for your board before using it. **(TA has done this for you!)**



Wyliodrin Studio GUI (3)

Programming

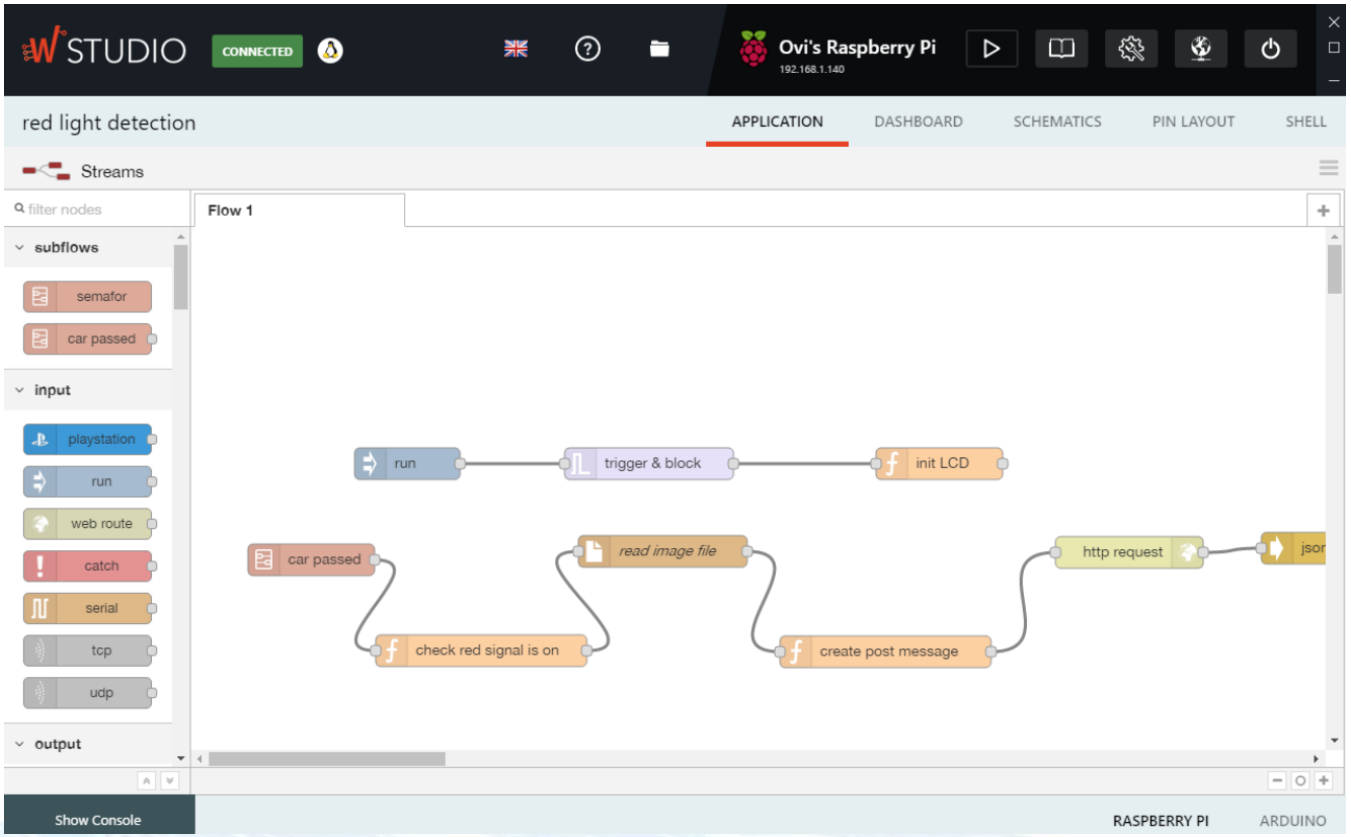
- Professional code editor.
- Advanced features such as autocomplete.



Wyliodrin Studio GUI (4)

Streams

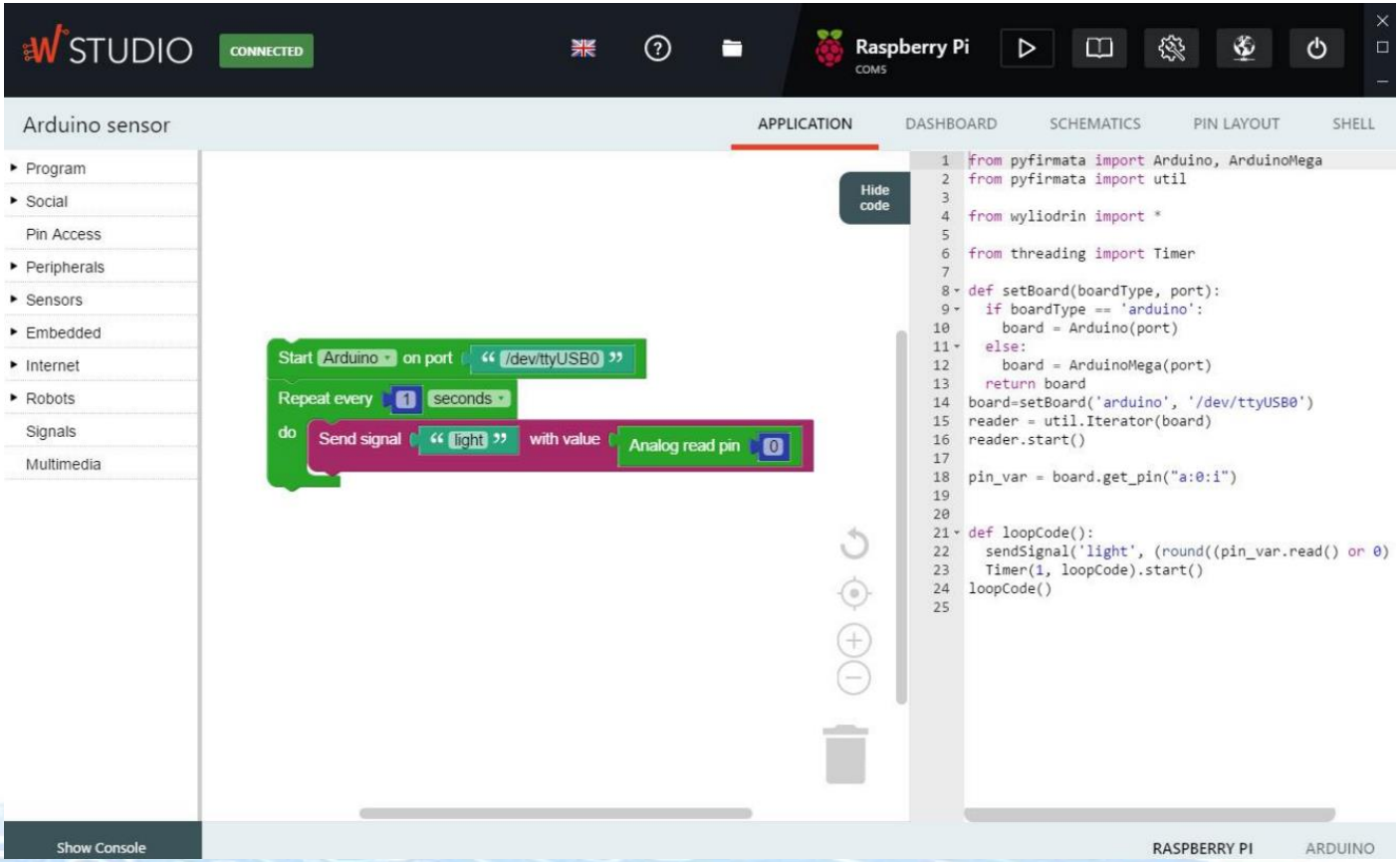
- Data flow programming.
- Implementation of node-red.



Wyliodrin Studio GUI (5)

Visual Programming

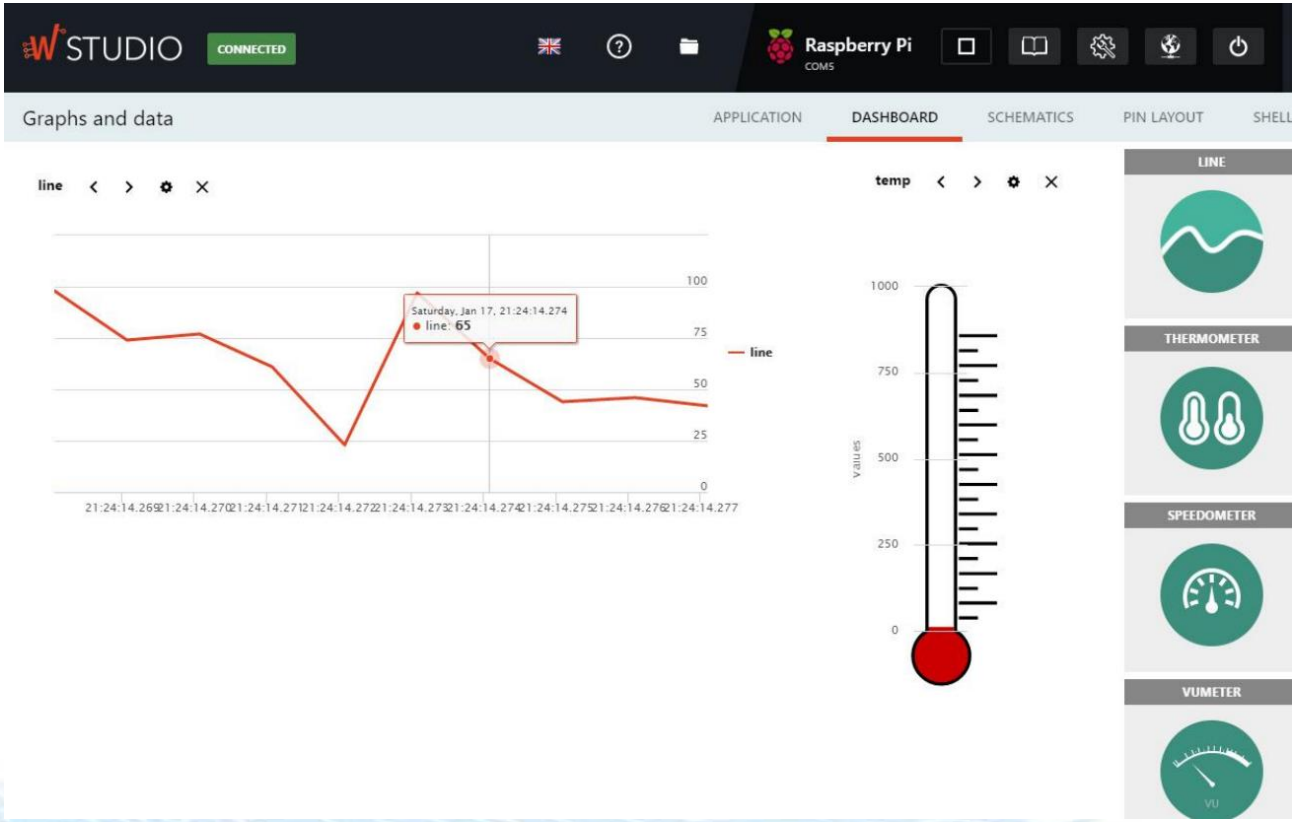
- Drag and drop blocks of code.
- Implements Google Blockly.
- View as Python code gets generated.



Wyliodrin Studio GUI (6)

Debug

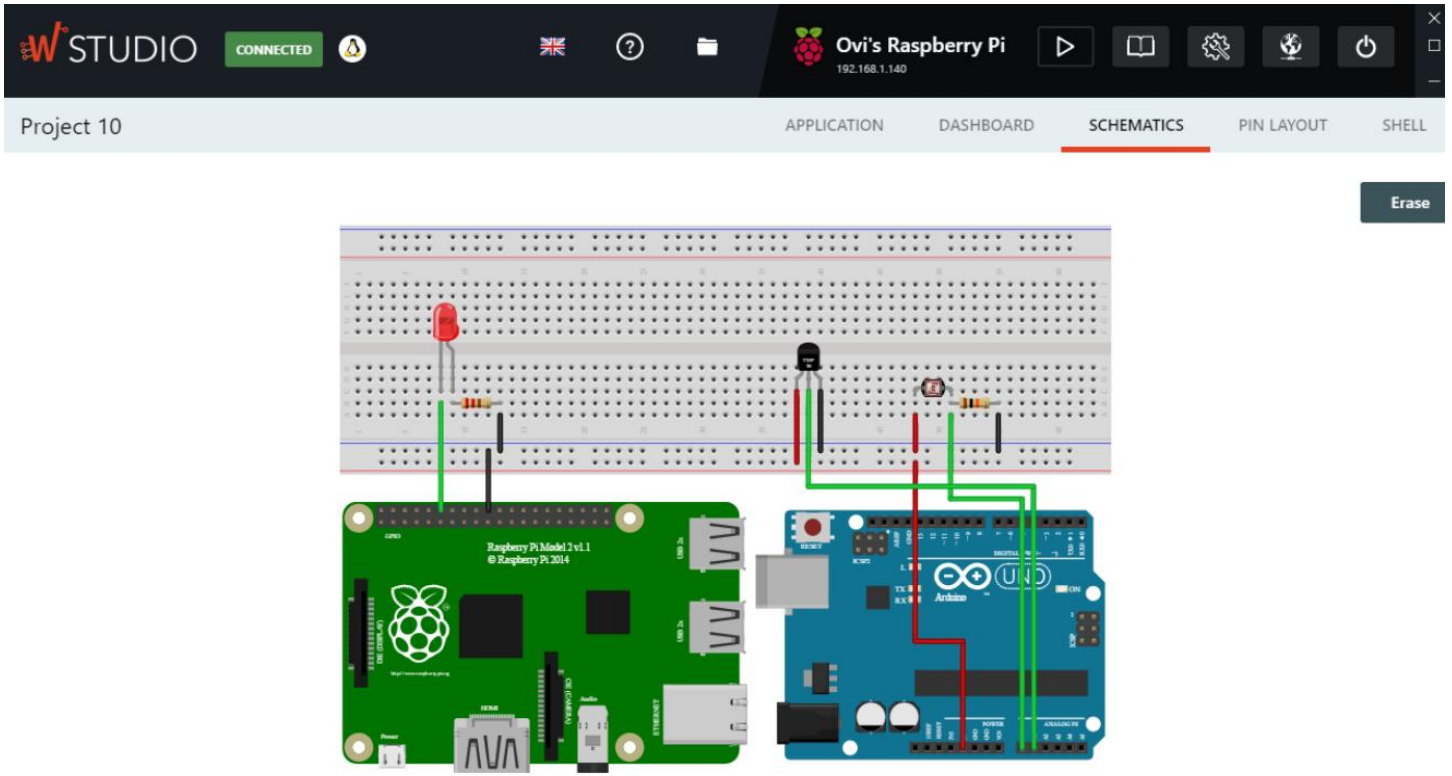
- Send signals to dashboard.
- Real time graphs allowing marks.



Wyliodrin Studio GUI (7)

Fritzing Schemas

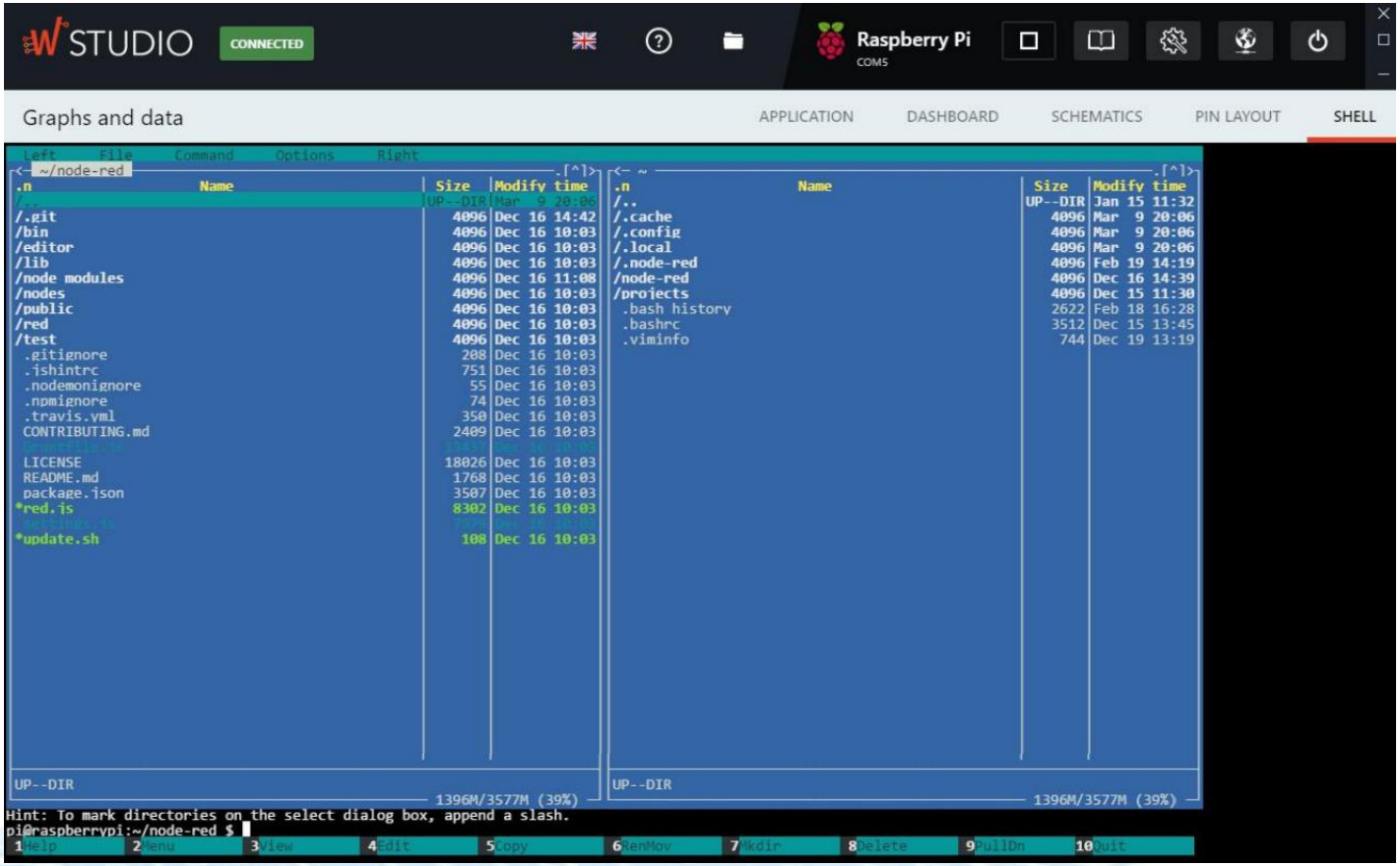
- Import SVG from Fritzing (http://fritzing.org).
- Attach schema to application (helps on documentation).



Wyliodrin Studio GUI (7)

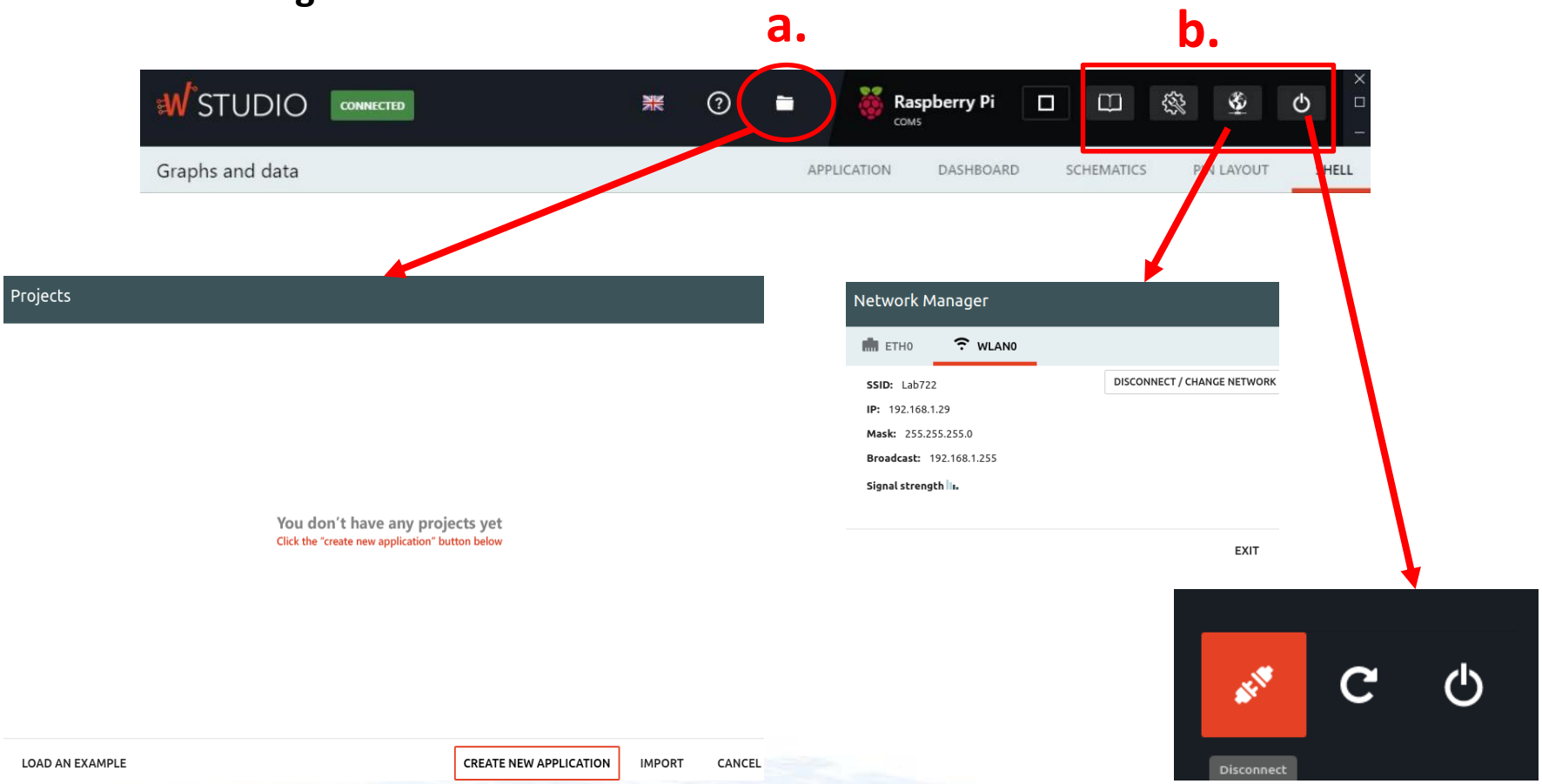
Shell

- Direct shell for advanced users (ssh terminal).



Wyliodrin Studio GUI (8)

a. Project Manager
b. Board Manager

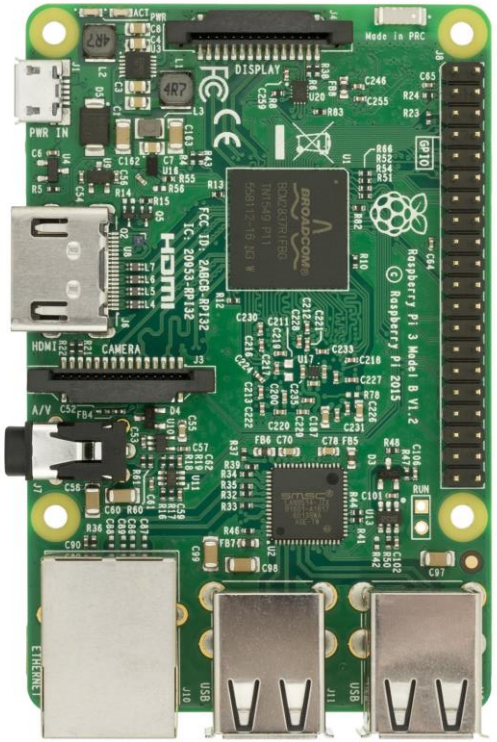


CHECKPOINT 1!

RASPBERRY PI AND ARDUINO BASICS

Raspberry Pi (1)

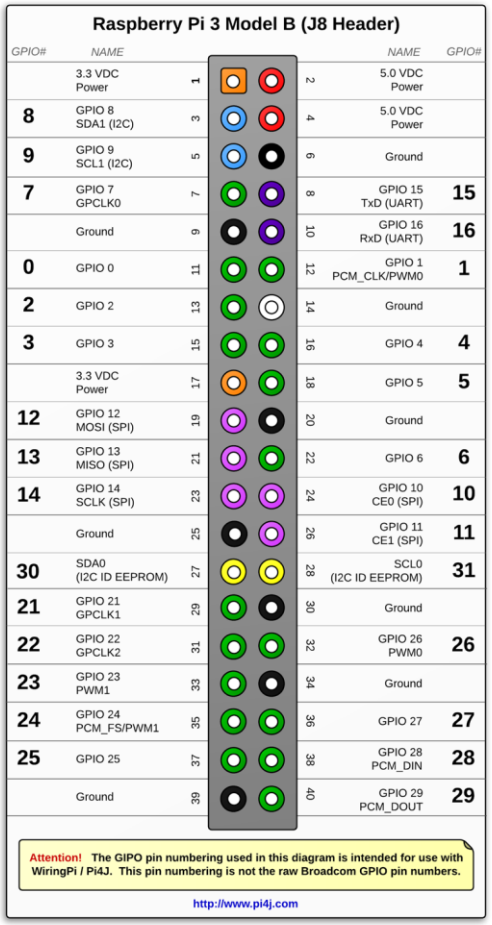
- The Raspberry Pi is a credit-card-sized computer with integrated Ethernet, Wi-Fi, Bluetooth 4, HDMI, USB, and audio.
- Initially made in Taiwan and China, but now all are manufactured in the UK.
- Capable of hosting an OS (Raspbian).
- Multipurpose: Office, Internet, games, etc.
- GPIO (control sensors).



Raspberry Pi

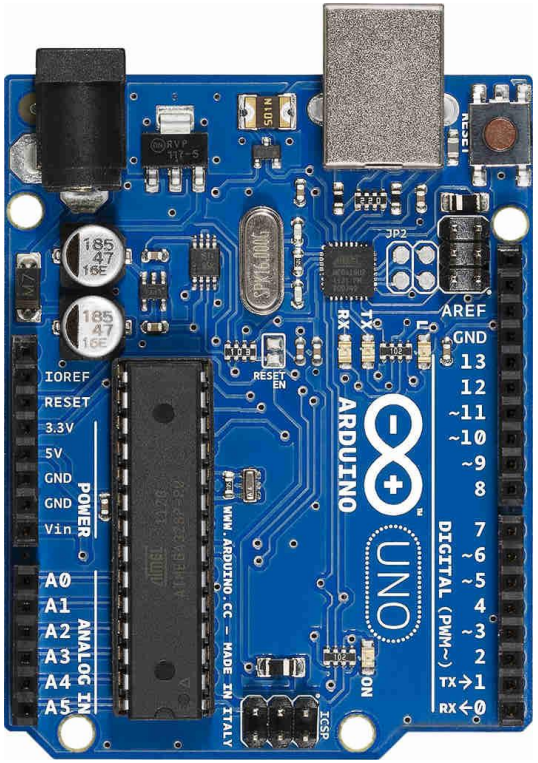
Raspberry Pi (2)

- Raspberry Pi can be directly used in electronics projects because it has a set of **general-purpose input/output (GPIO)** pins.
- These GPIO pins can be accessed for controlling sensors (temperature, light, motion, proximity, etc.) and actuators (LEDs, motors, relays, etc.).
- All of these GPIO pins are digital IO. In order to read/write analog inputs, an external ADC (analog-to-digital converter) can be used.



Arduino (1)

- Arduino is an open-source electronics platform.
- Consists of both a physical programmable circuit board (microcontroller) and an IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.
- It has analog and digital GPIO.
- It functionality can be extended via ArduinoShields.



ARDUINO

Arduino (2)

Digital Pins

Digital pins are GPIO.

Each pin has an internal pull-up resistor which can be turned on and off using digitalWrite() (with a value of HIGH or LOW, respectively).

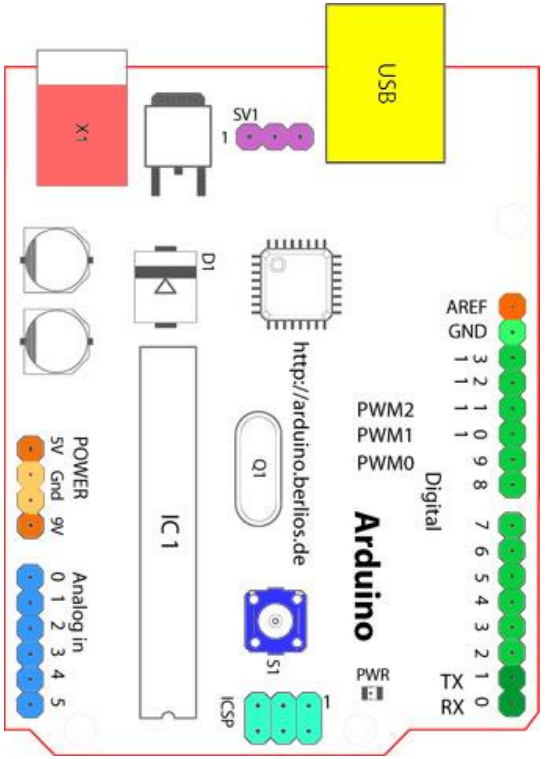
The maximum current per pin is 40 mA.

Analog Pins

Analog pins are input.

Support 10-bit analog-to-digital conversion (ADC) using the analogRead() function.

Most of the analog inputs can also be used as digital pins (depends on the model).



Arduino (3)

Why use Arduino when you have a Pi?

- The two platforms are complementary.
- There are quite a few situations where you might want to put the Arduino and Pi together:
  - To use the large number of libraries and sharable examples for the Arduino.
  - To supplement an Arduino project with more processing power.
  - When you're dealing with 5V logic levels. The Pi operates at 3.3V, and its pins are not tolerant of 5V. The Arduino can act as a "translator" between the two.

How Arduino and Raspberry Communicate?

- Installing "Firmata" is one of the solutions.
- Firmata is a generic protocol for communicating with microcontrollers from software on a host computer.
- It is intended to work with any host computer software package.

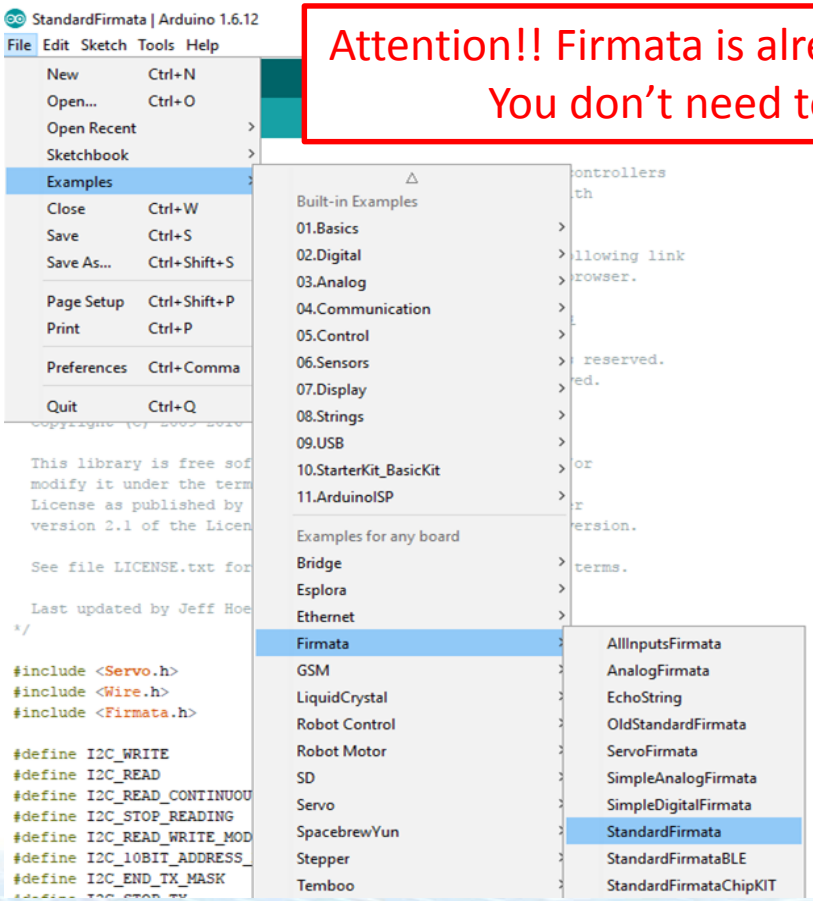
Arduino : Installing Firmata (1)

1. Download and Install the Arduino IDE Application in your computer from https://www.arduino.cc/en/Main/Software.

Attention!! Firmata is already installed in your Arduino.
You don't need to execute these steps.

Arduino : Installing Firmata (2)

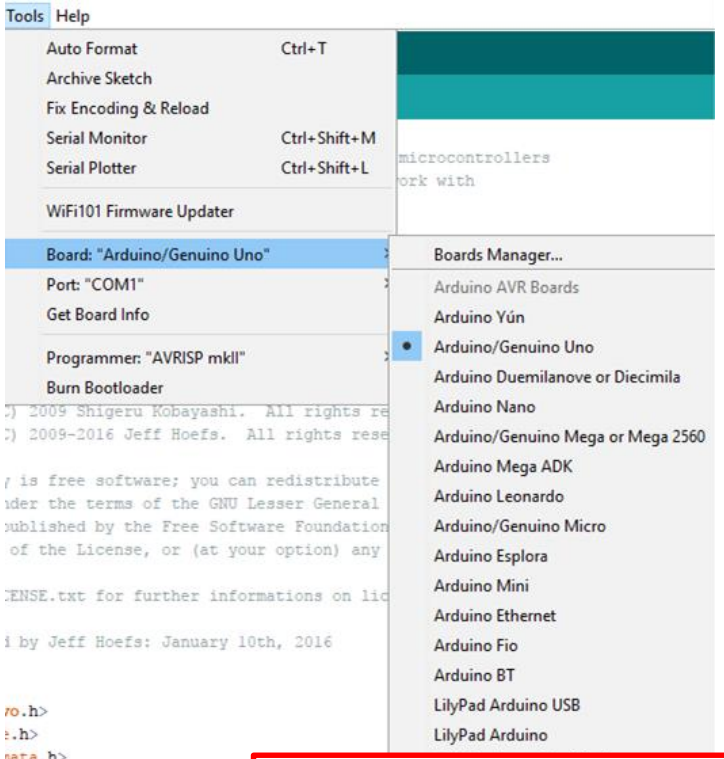
2. Open Arduino IDE application and plug the Arduino board into a your computer.
3. Open File > Examples > Firmata > StandardFirmata.

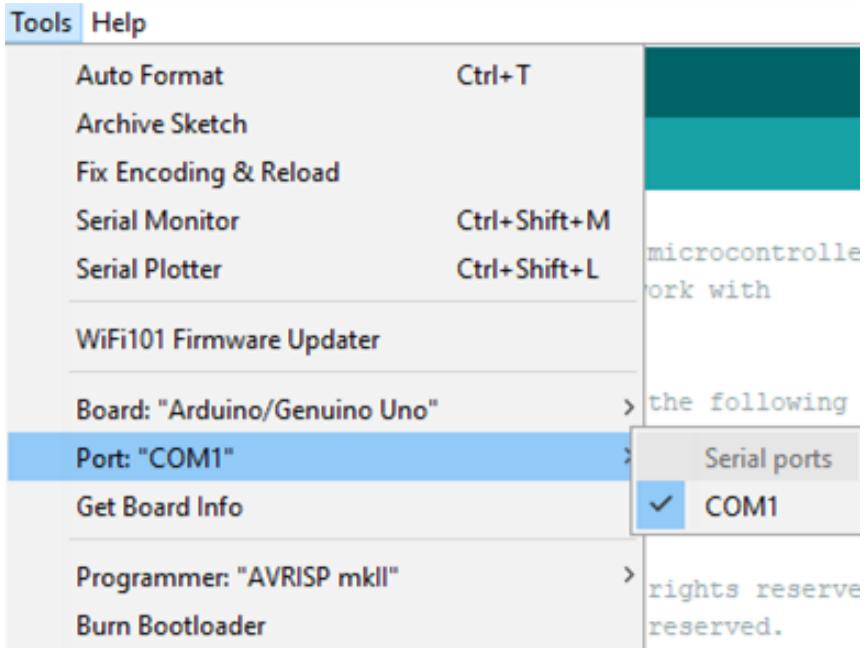


Attention!! Firmata is already installed in your Arduino.
You don't need to execute these steps.

Arduino : Installing Firmata (3)

4. Select your Arduino board model in Tools > Board: ”Your Arduino Model”.
5. Select the port which your board is connected in Tools >Port: ”Your Arduino Port”.

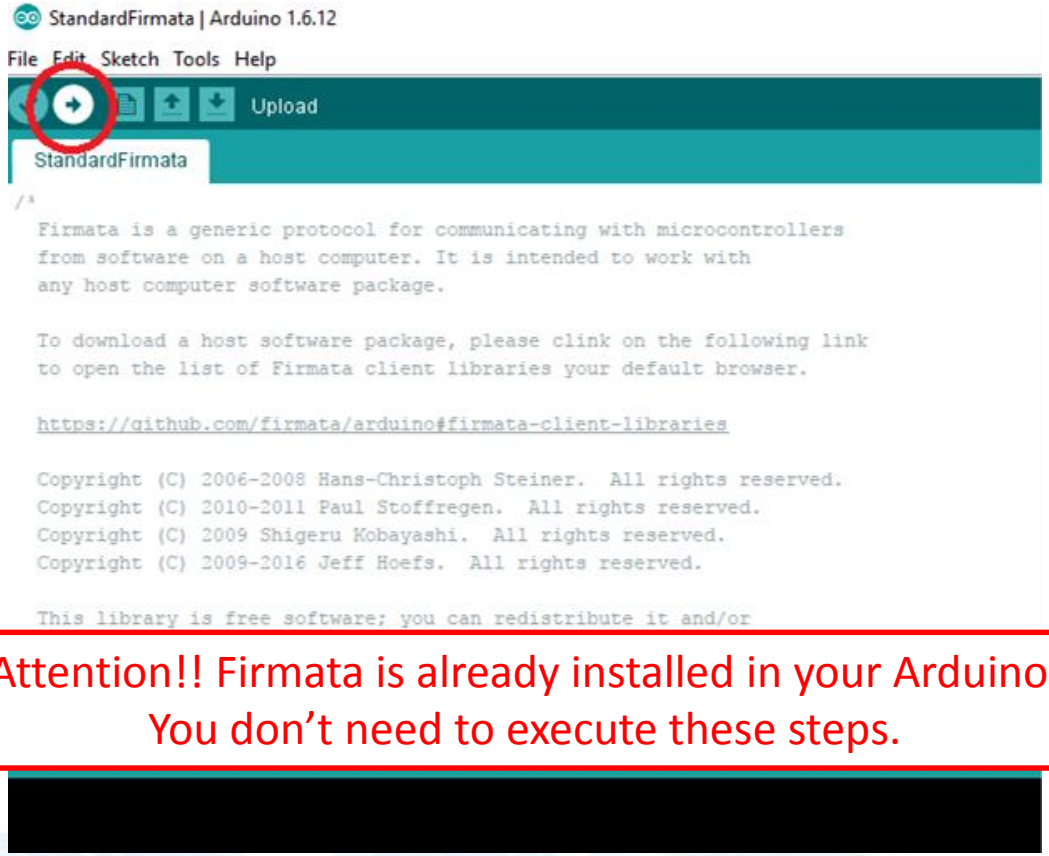




Attention!! Firmata is already installed in your Arduino.
You don’t need to execute these steps.

Arduino : Installing Firmata (4)

5. Upload “StandartFirmata” into your board by pressing highlighted button or simply choose Sketch > Upload.

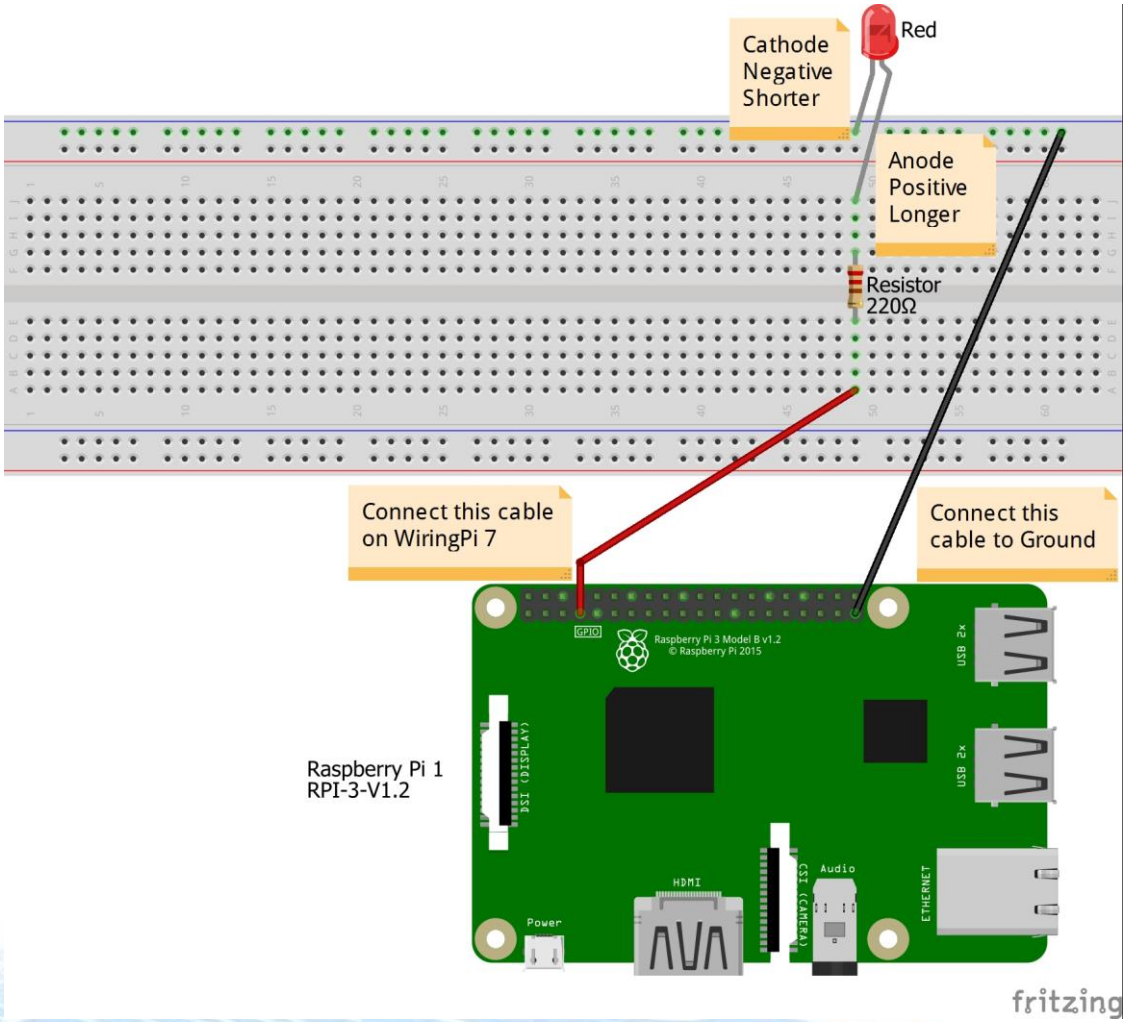


Attention!! Firmata is already installed in your Arduino.
You don’t need to execute these steps.

HELLO WORLD APPLICATION

Hello World Application (1)

1. Prepare your Raspberry Pi and your board according to the schematics below.



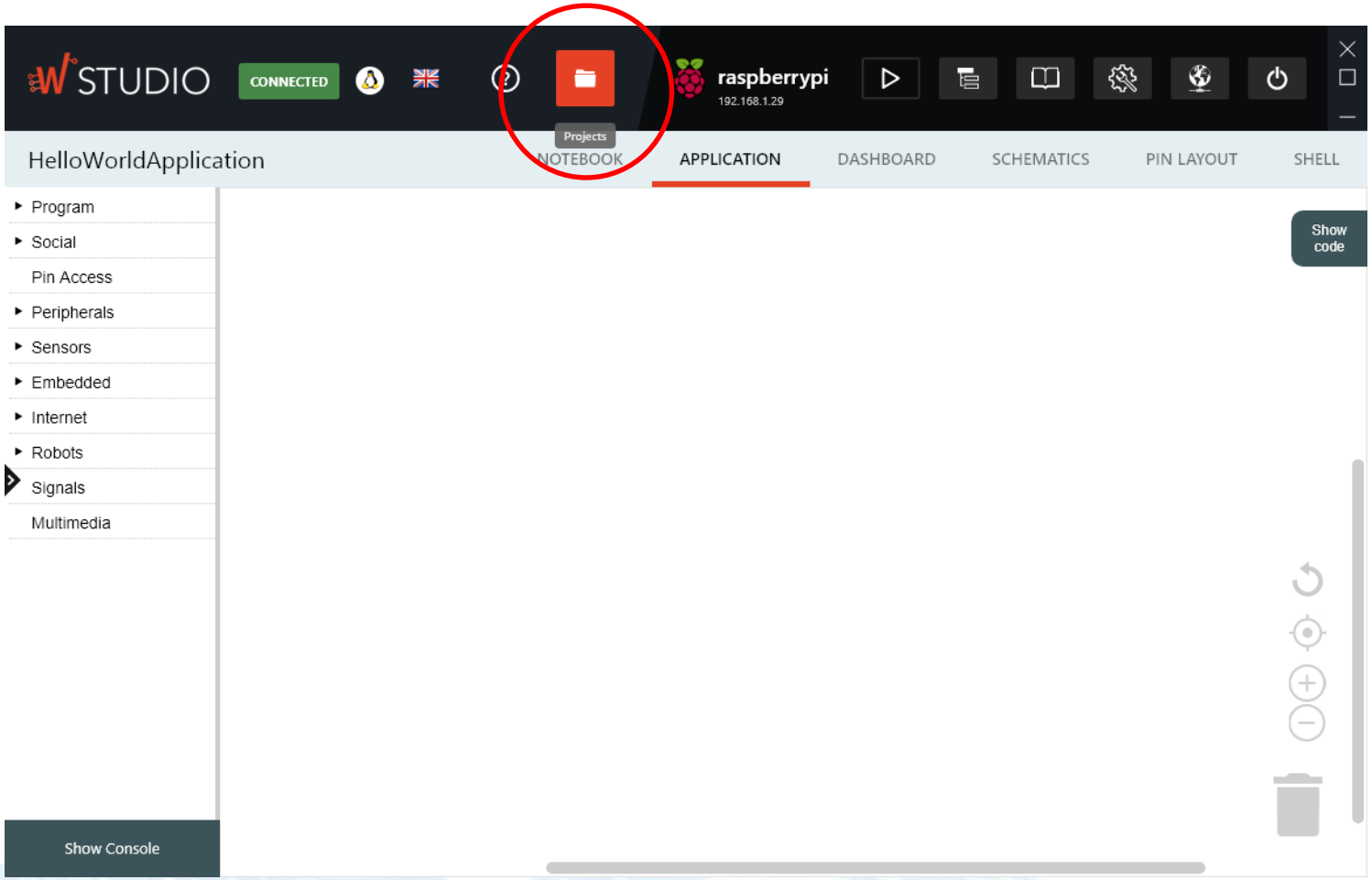
Goal: Turn ON/OFF LED

Components:

- 1 Raspberry Pi.
- 1 LED.
- 1 Resistor 220.
- Cables.

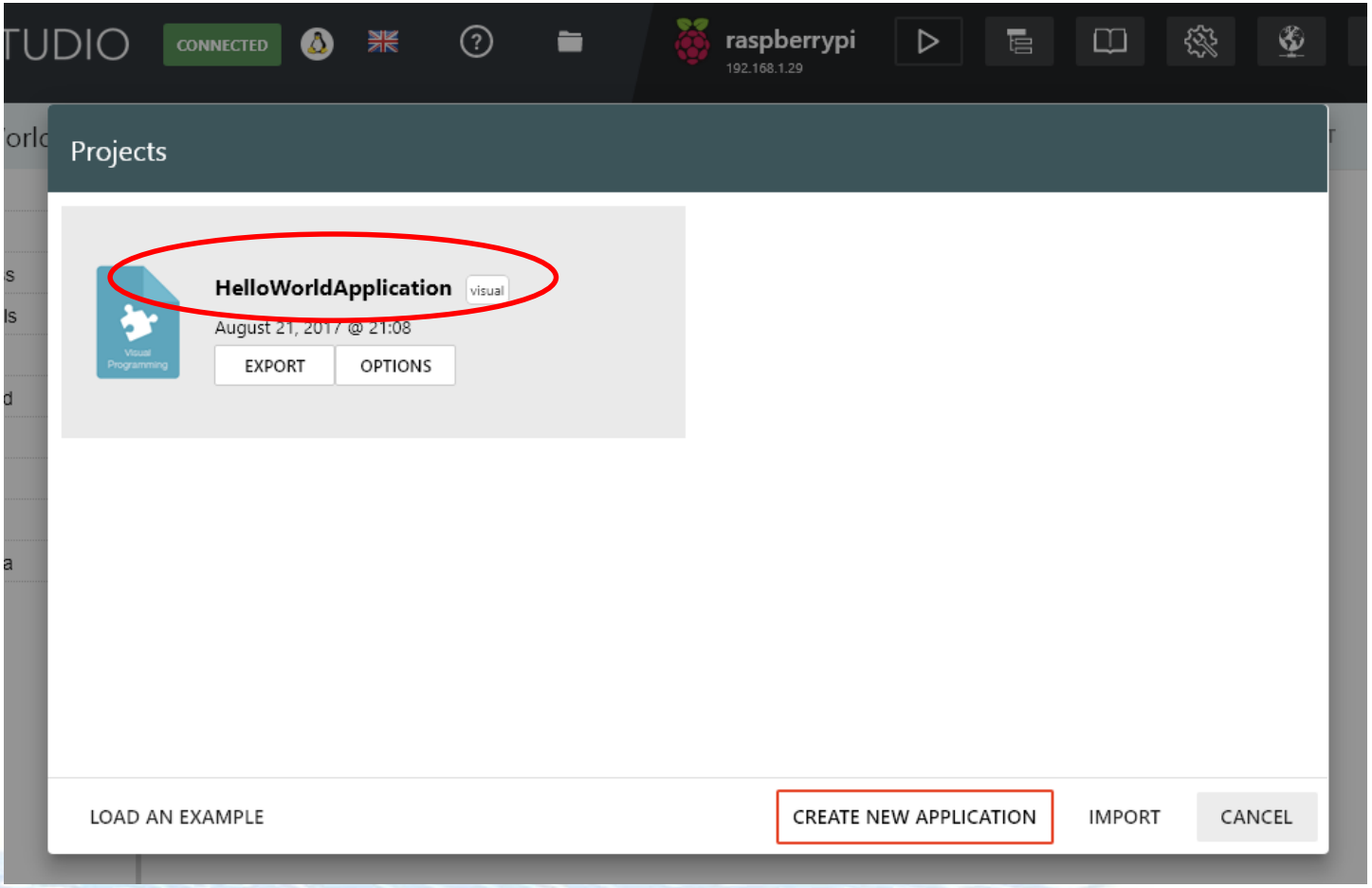
Hello World Application (2)

2. Click on “Projects”.



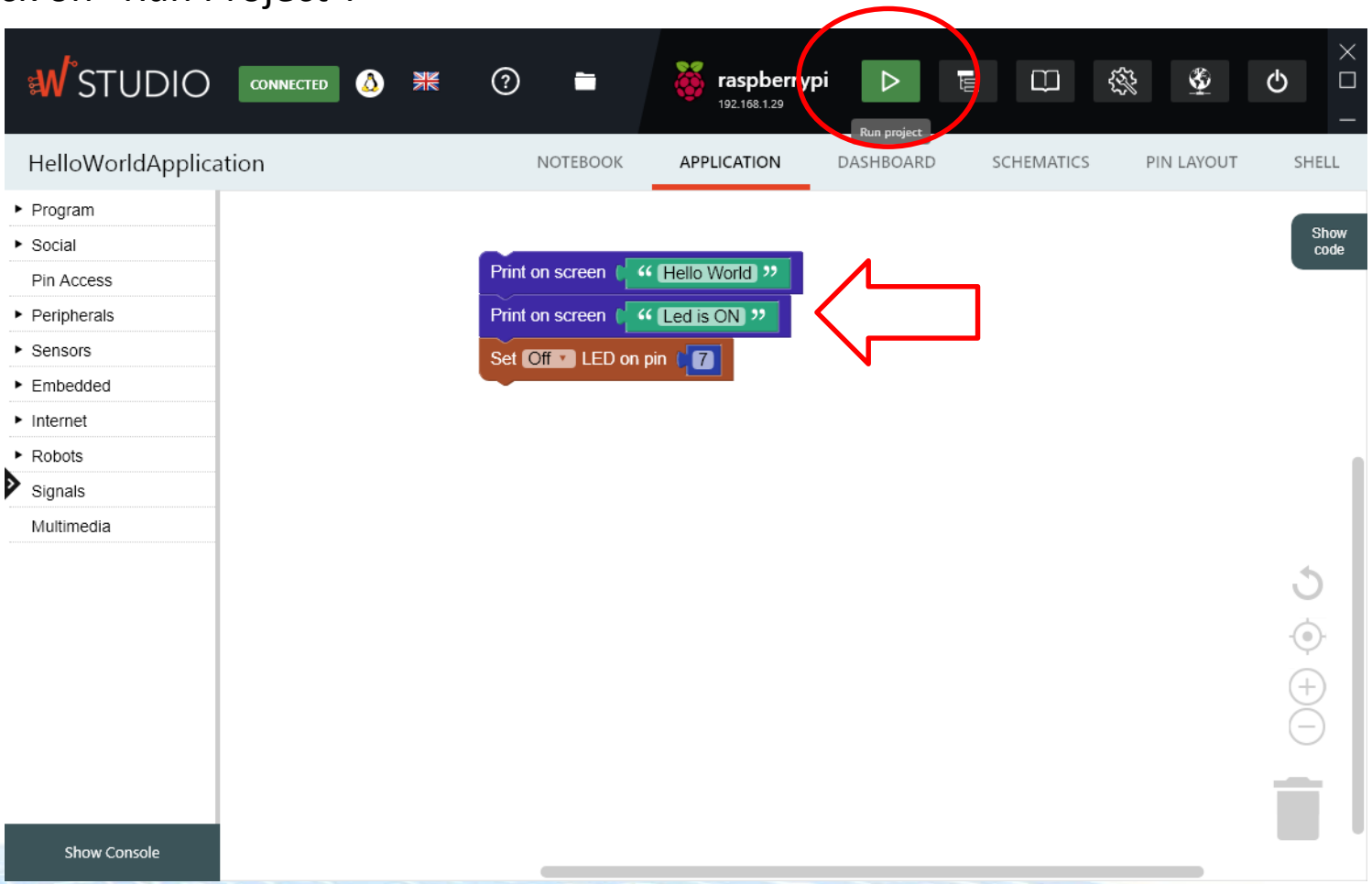
Hello World Application (3)

3. Click on “HelloWorldApplication”.



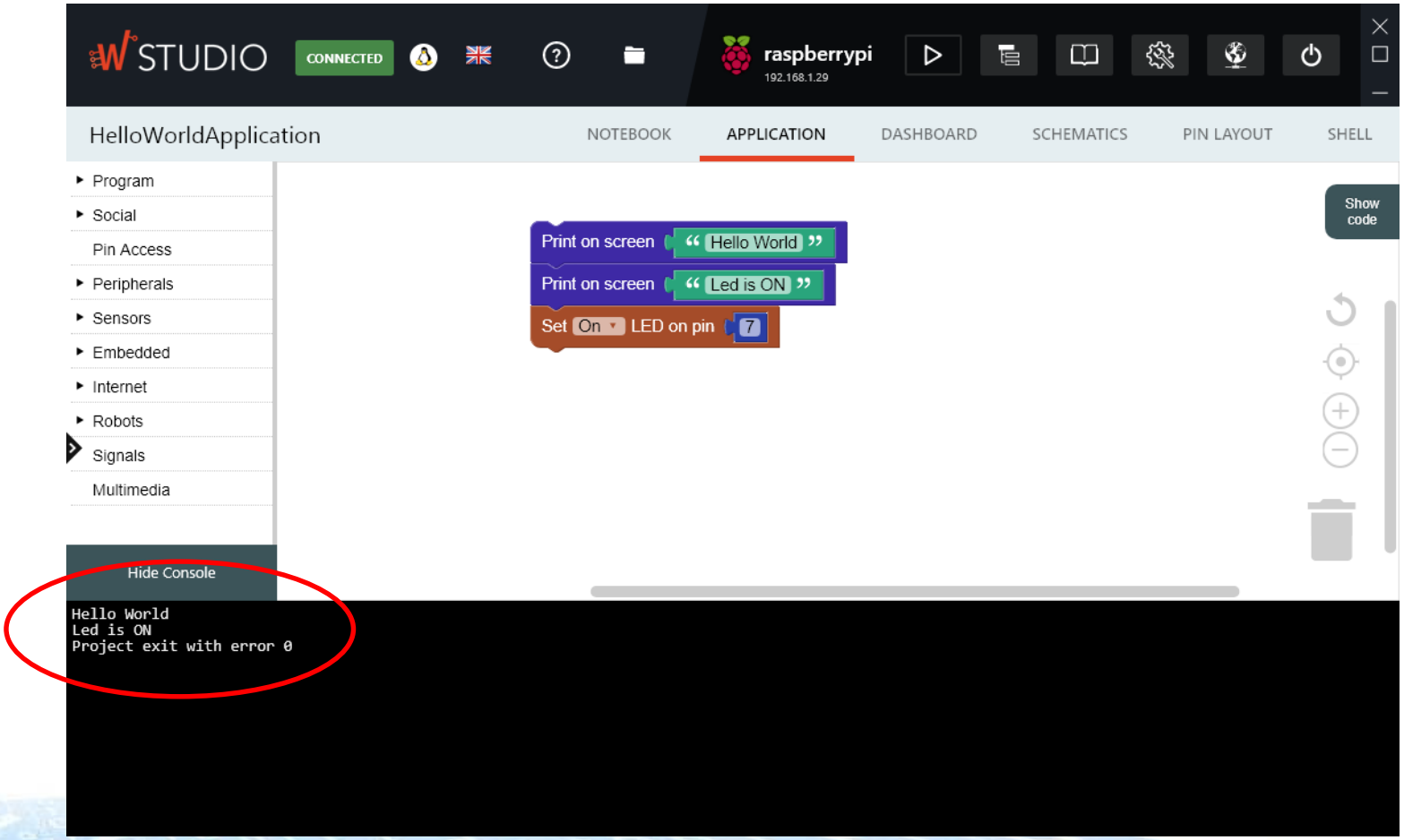
Hello World Application (4)

4. In the “Application” tab you will use a few blocks.
5. Click on “Run Project”.



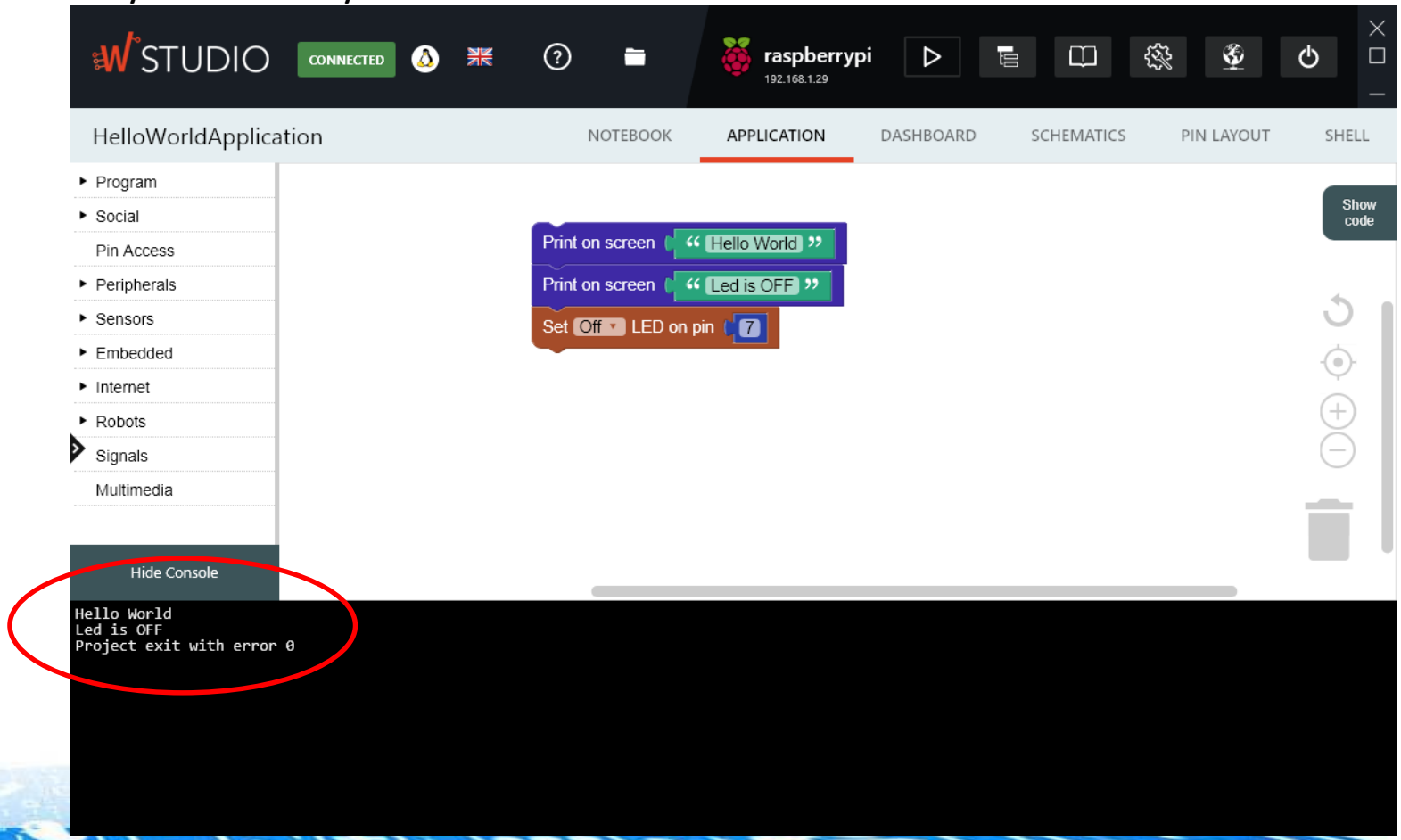
Hello World Application (5)

6. Notice the output in the console.
7. Verify the LED in your board is turned ON.



Hello World Application (6)

8. Modify the program so that the LED is OFF.
9. Modify also the message on screen to “LED is OFF”.
10. Verify the LED in your board is turned OFF.



CHECKPOINT 2!

TRAFFIC LIGHT SIMULATOR APPLICATION

Goal

- Simulate a vehicles traffic light and a pedestrian traffic light:

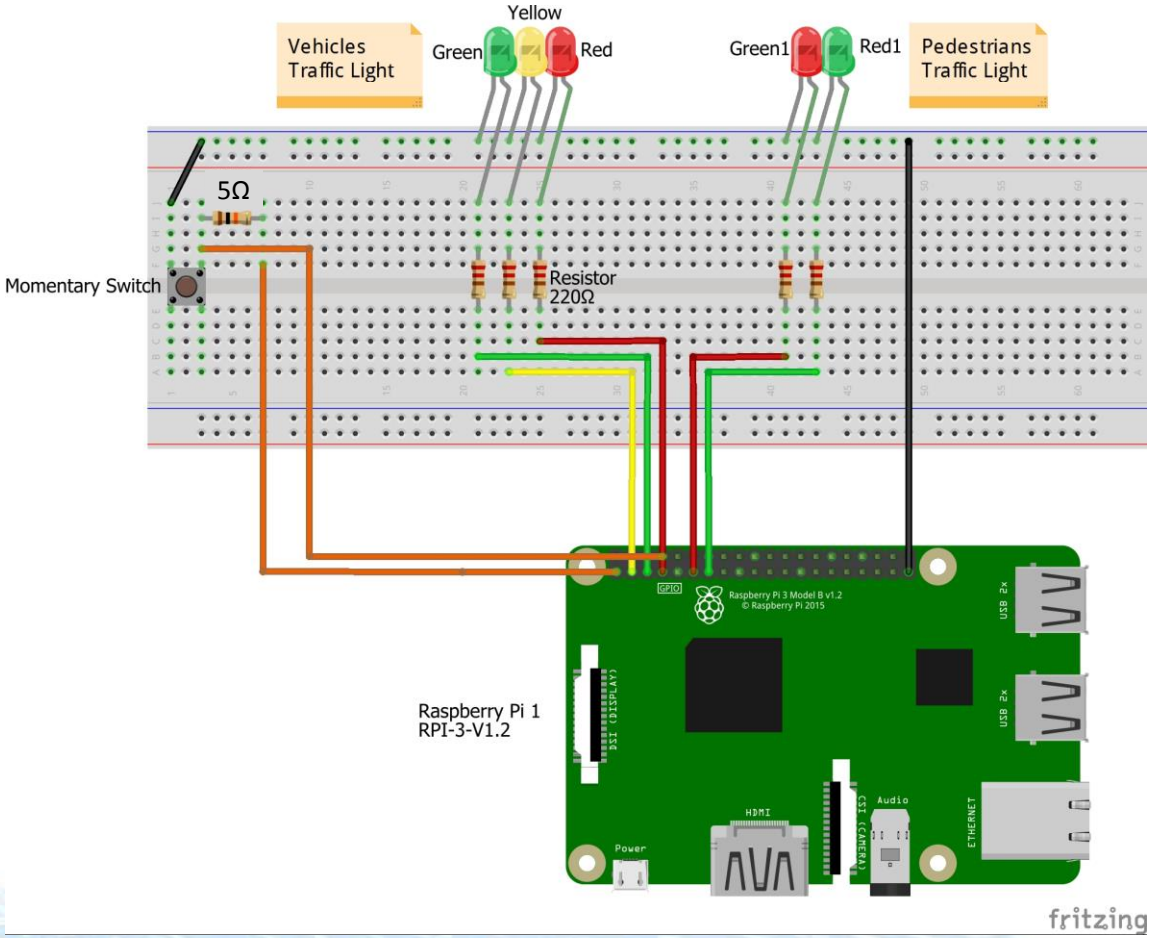
1. When the pedestrian pushes a button, the vehicles traffic light should change from green, to yellow, and finally red.
2. The pedestrian traffic light should change from red to green at the same time as (1) finishes.
3. The pedestrian traffic light should remain 30 seconds in green color before returning to red color, and at the same time the vehicles traffic light should return to green color.

List of Components

- 1 breadboard.
- 1 Raspberry Pi 3.
- 5 resistors 220 Ω, 1 resistor 5Ω.
- 2 LED green, 2 LED red, 1 LED yellow.
- 1 momentary switch.
- 9 cables.

Schematics

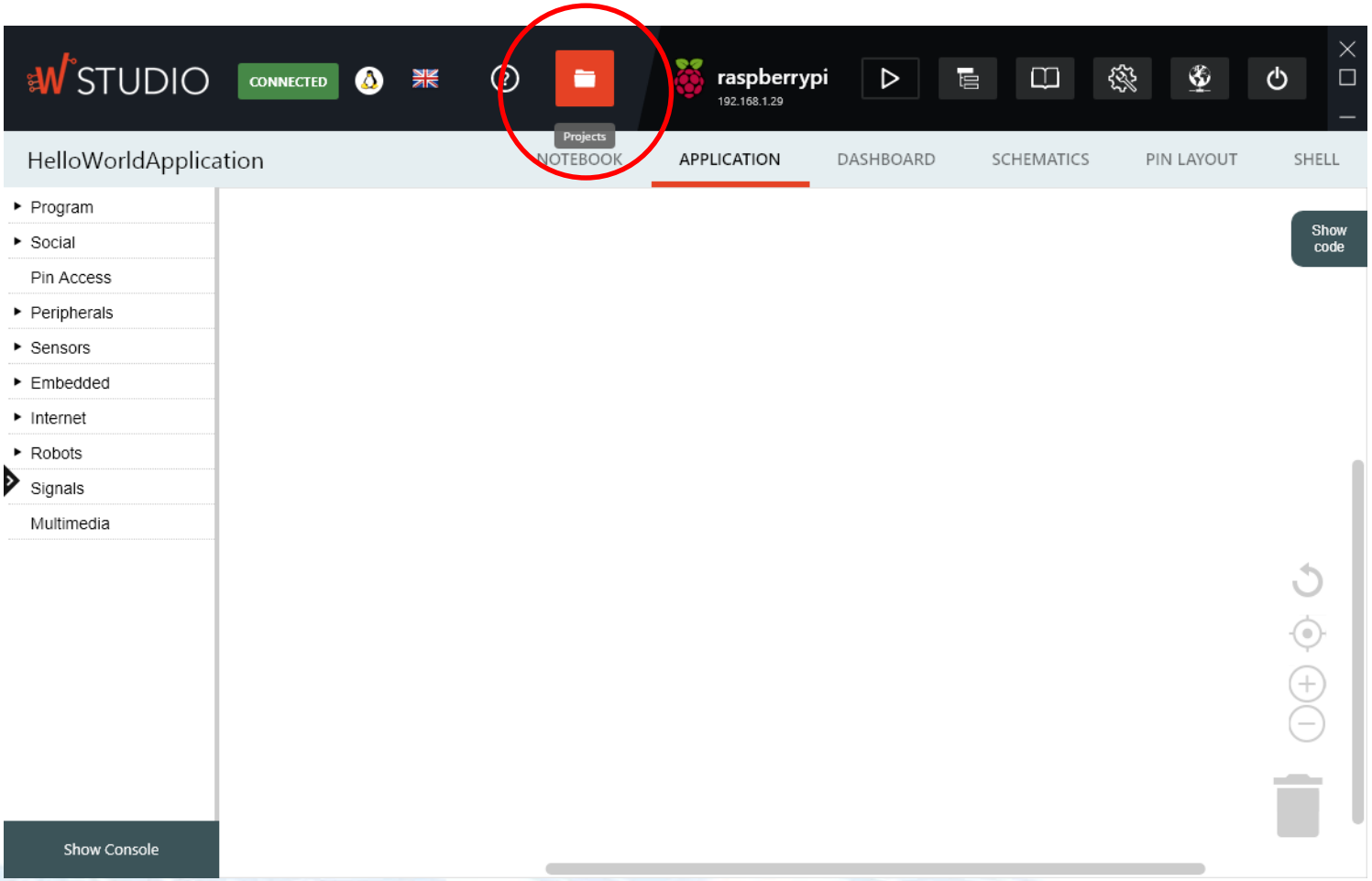
Connect all the components according to the schema.
Ask TA if you are not sure how to connect the components.



CHECKPOINT 3!

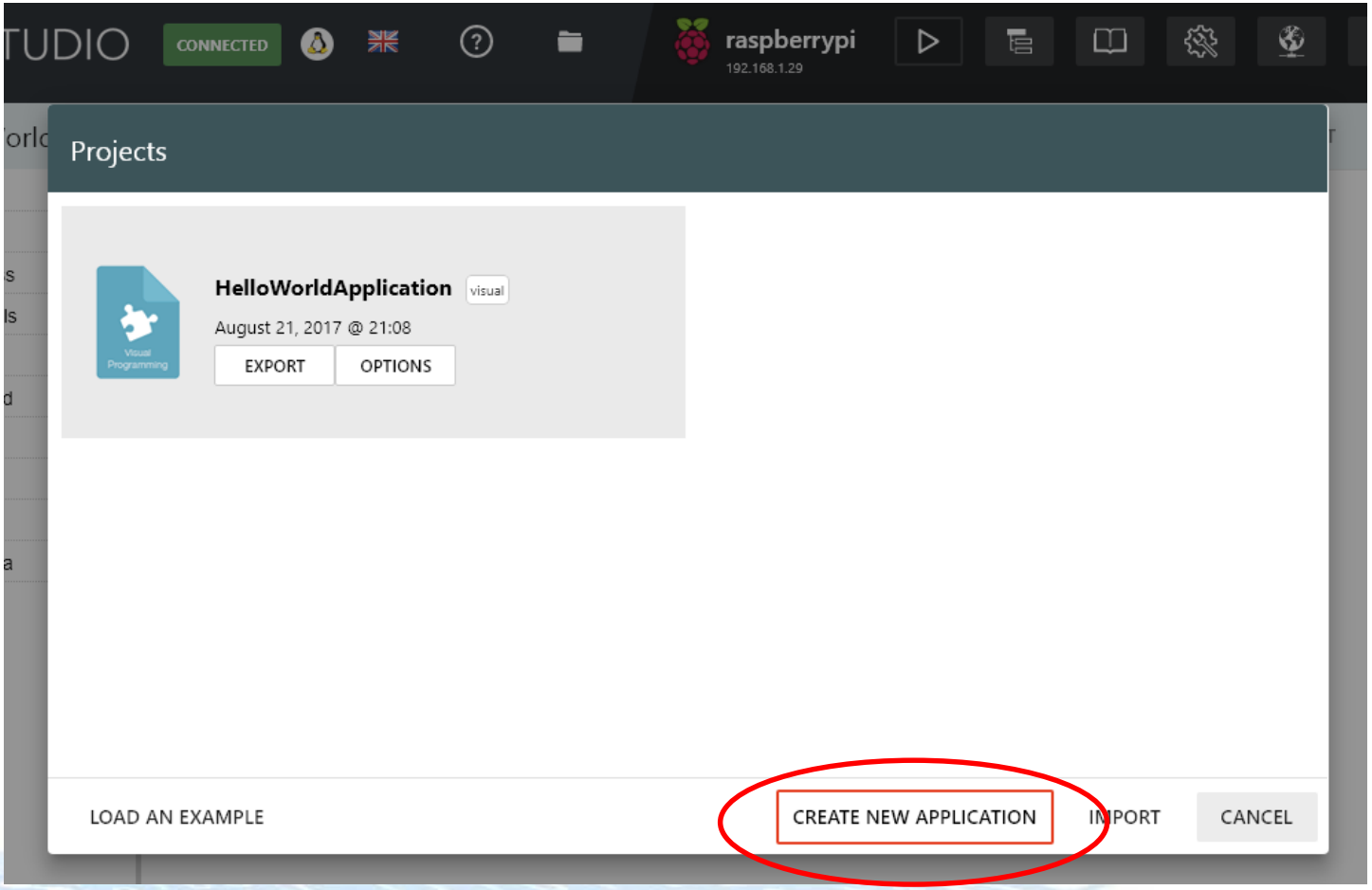
Traffic Light Application (1)

1. Click on “Projects”.



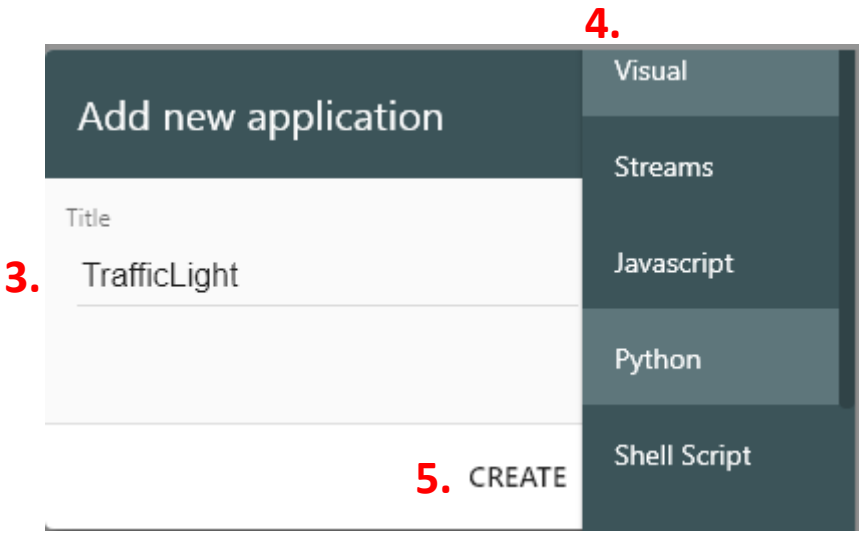
Traffic Light Application (2)

2. Click on “CREATE NEW APPLICATION”.



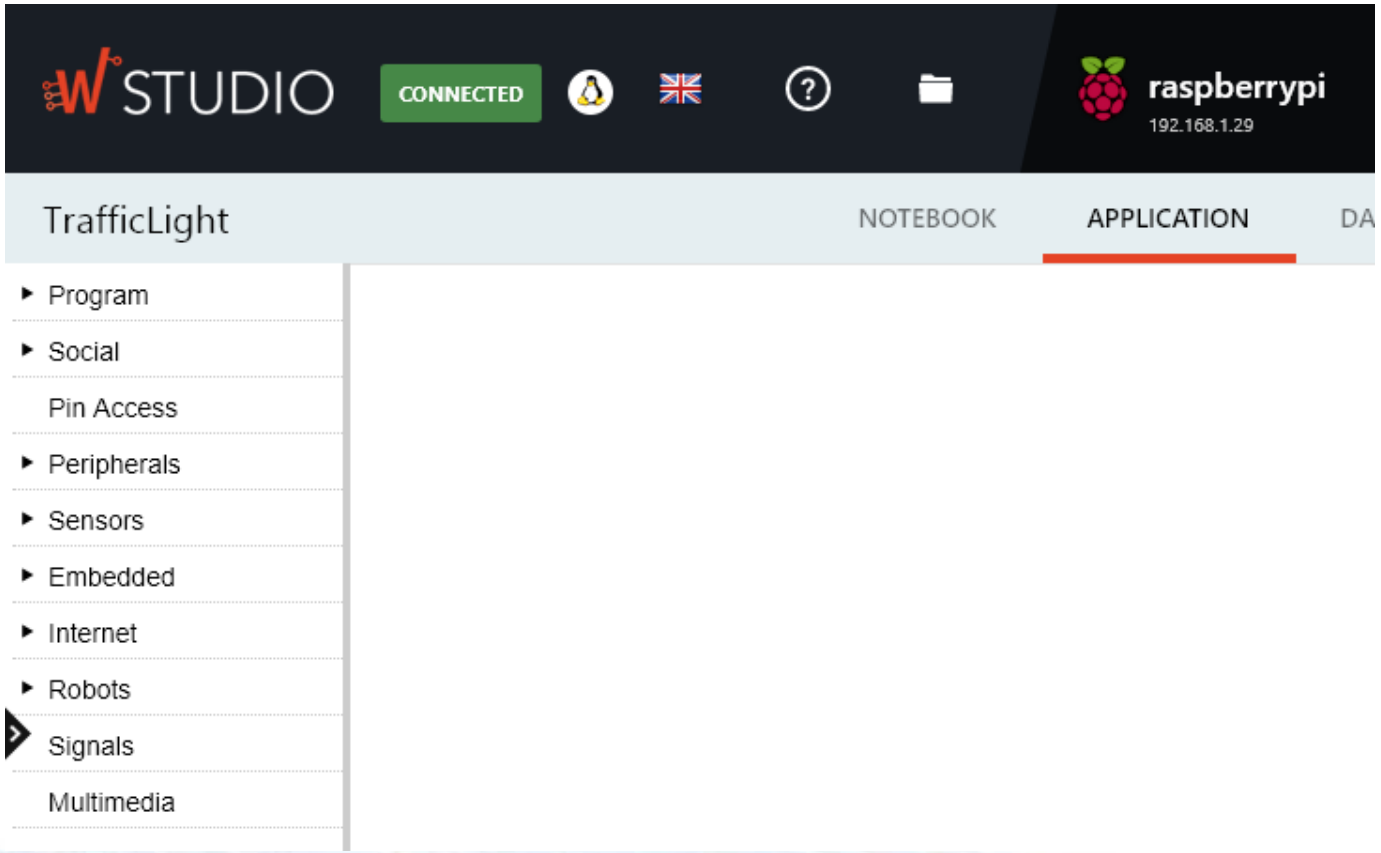
Traffic Light Application (3)

3. Input “TrafficLight” as the Title.
4. Select “Visual” as the Language.
5. Click on “Create”.



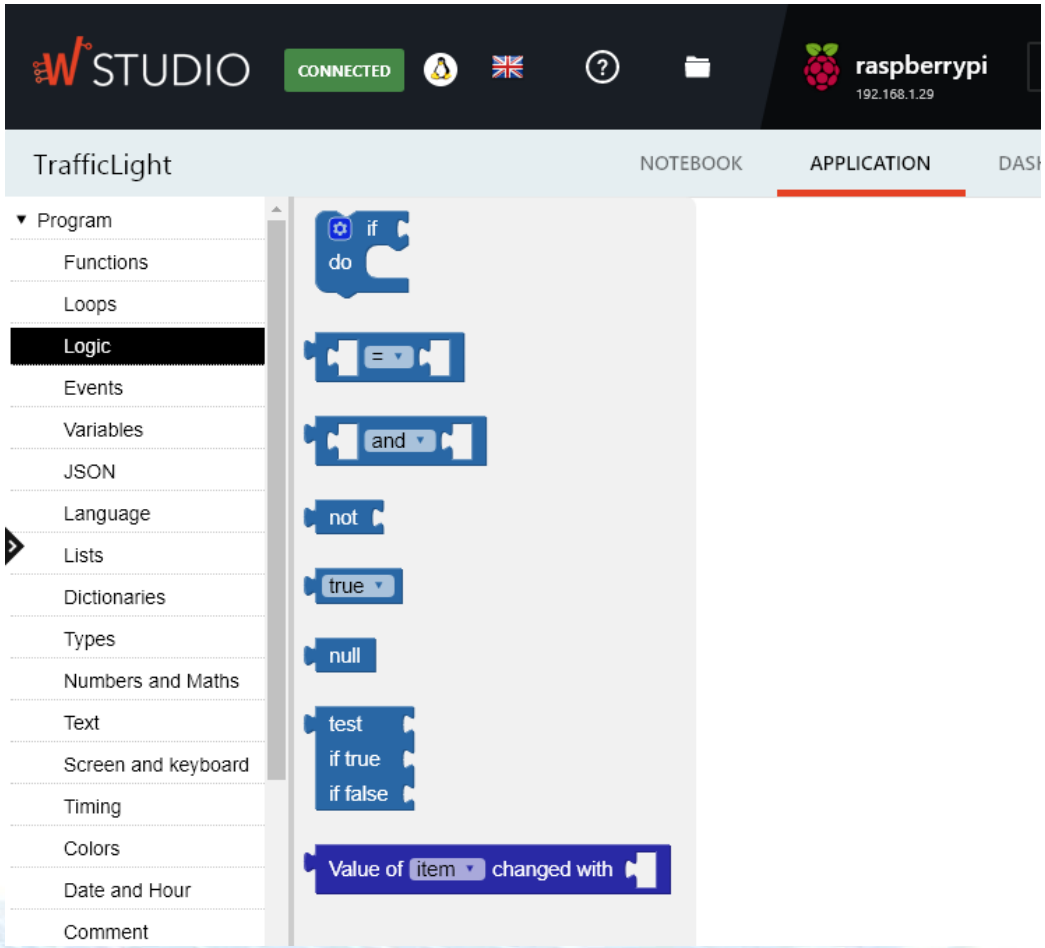
Coding with Blocks (1)

Visual Programming is a language based on Google Blockly and its purpose is learning. While writing your program in Visual Programming, you can see the Python output.



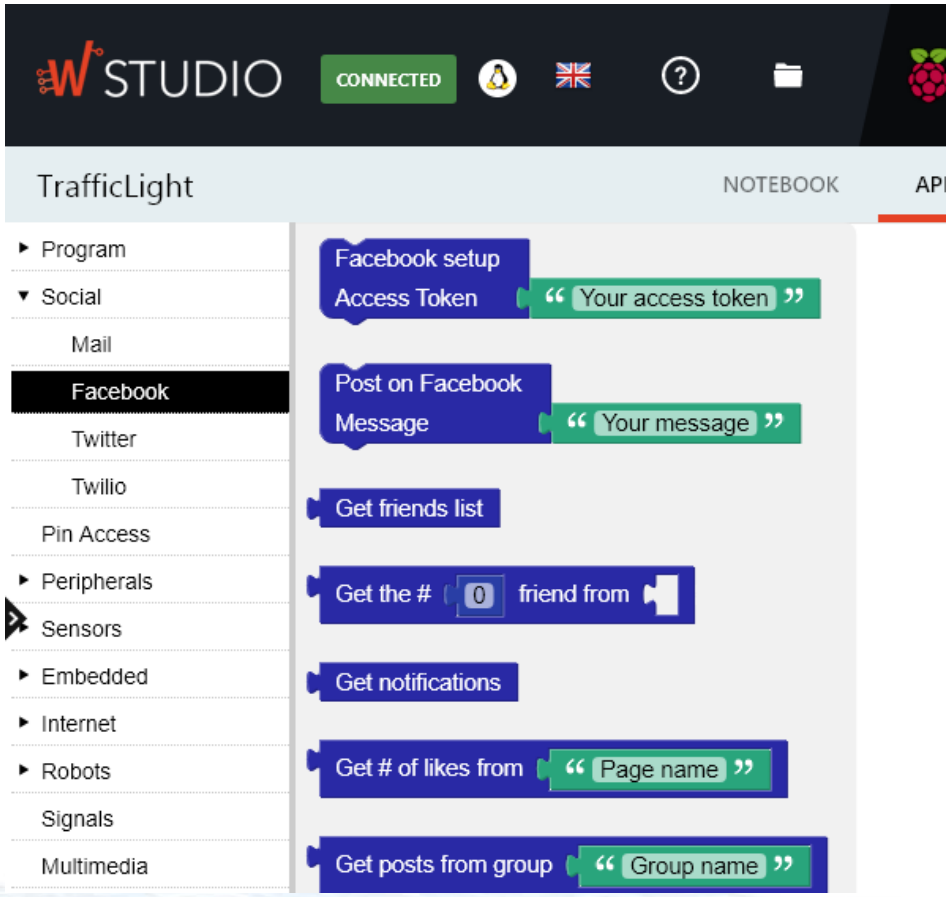
Coding with Blocks (2)

Program includes functions, loops, logic, variables, printing, timing, lists, math, etc.



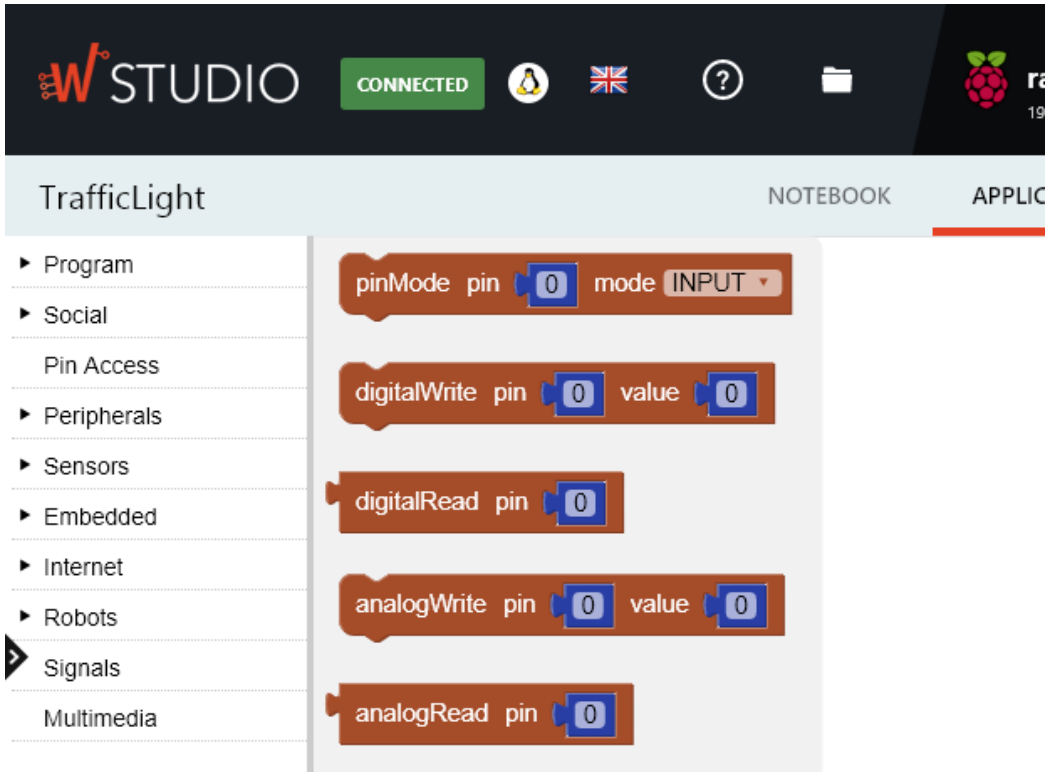
Coding with Blocks (3)

Social includes blocks for Email, Facebook, Twitter, and Twilio.



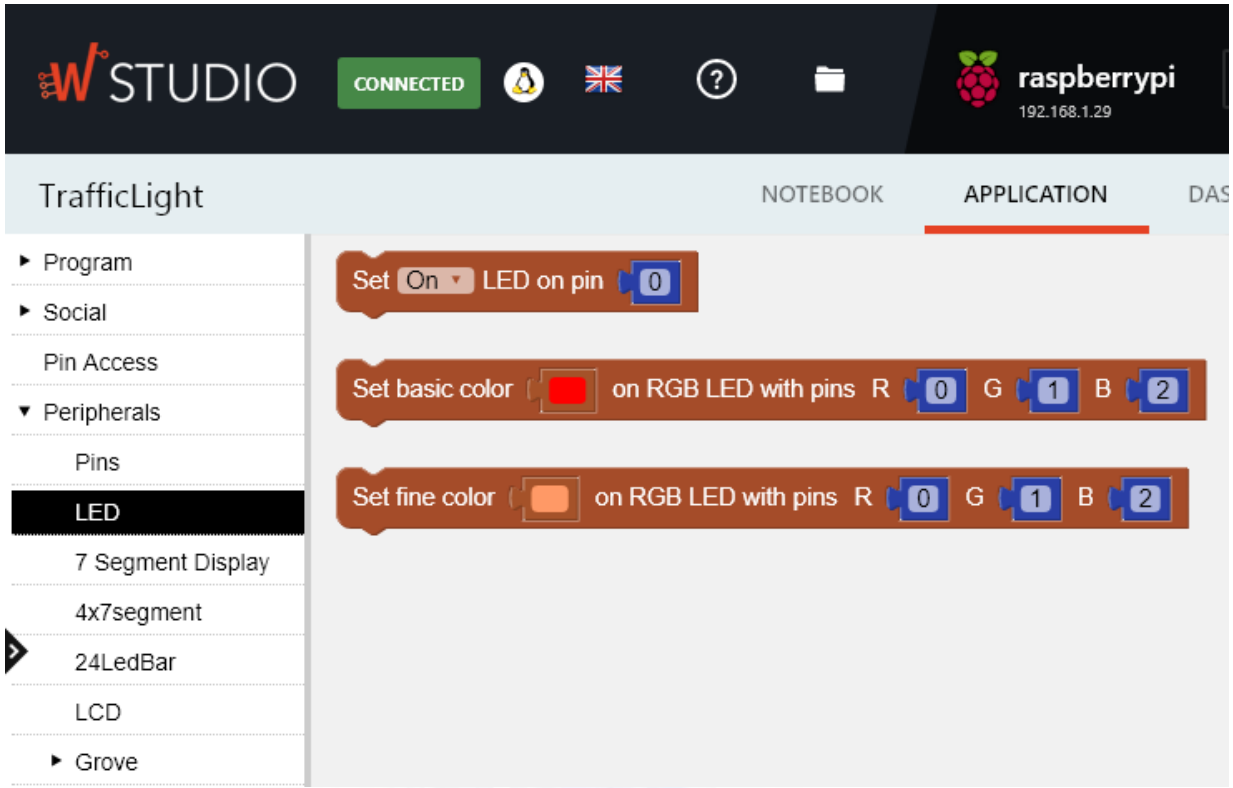
Coding with Blocks (4)

Pin Access includes blocks for setting the mode of boards' pin to Input or Output, and reading/writing Digital and Analog pins in the boards.



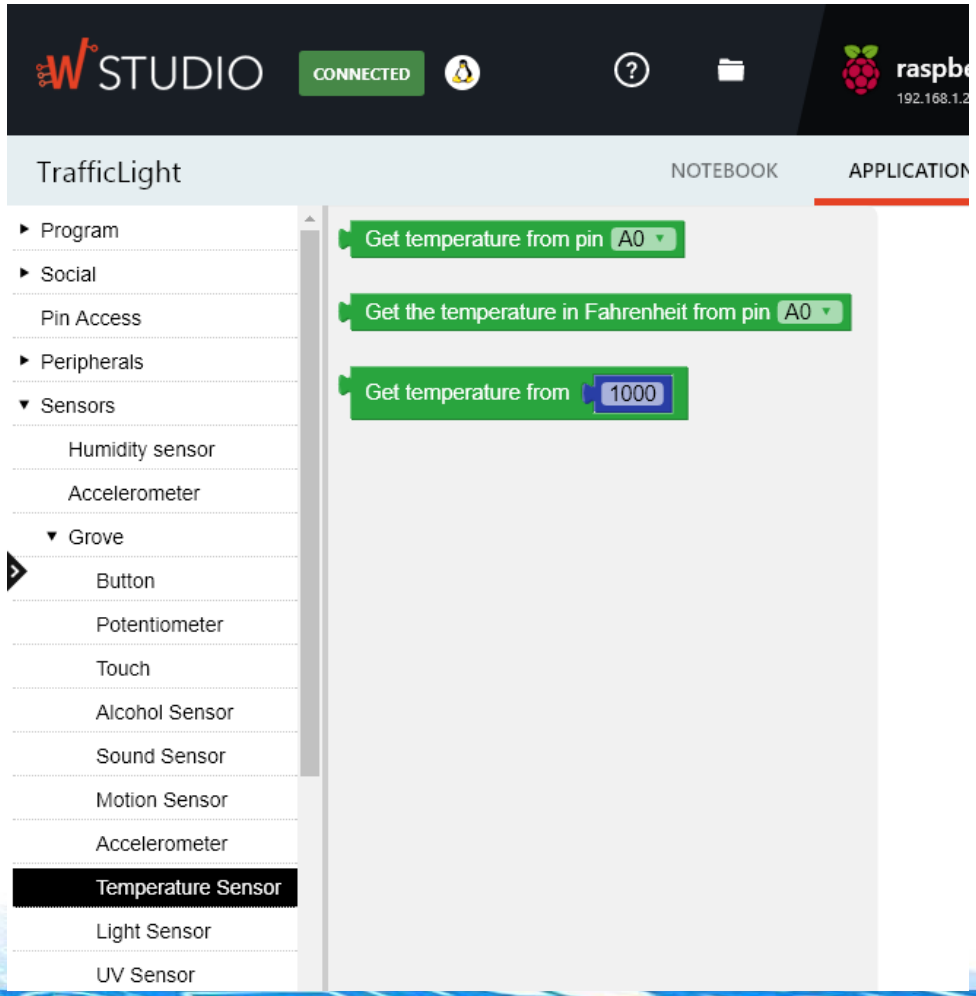
Coding with Blocks (5)

Peripherals provides blocks for accessing LEDs (turn ON/OFF) and Digital Displays.



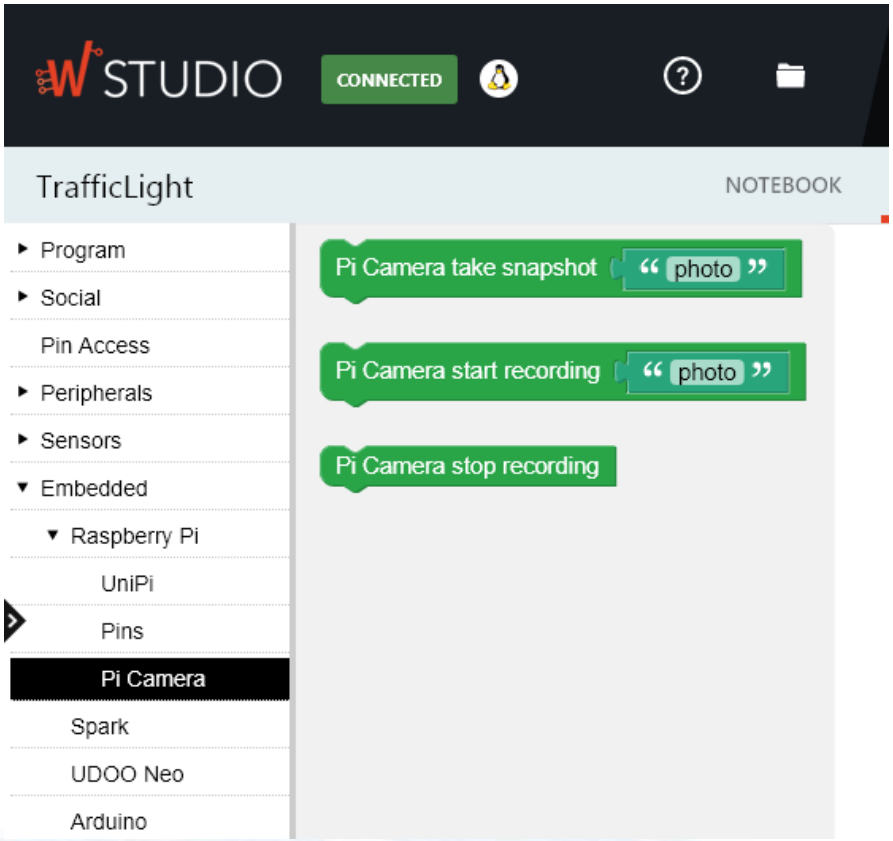
Coding with Blocks (6)

Sensors provides blocks for reading values from Arduino, Grove, and Raspberry Pi common sensors.



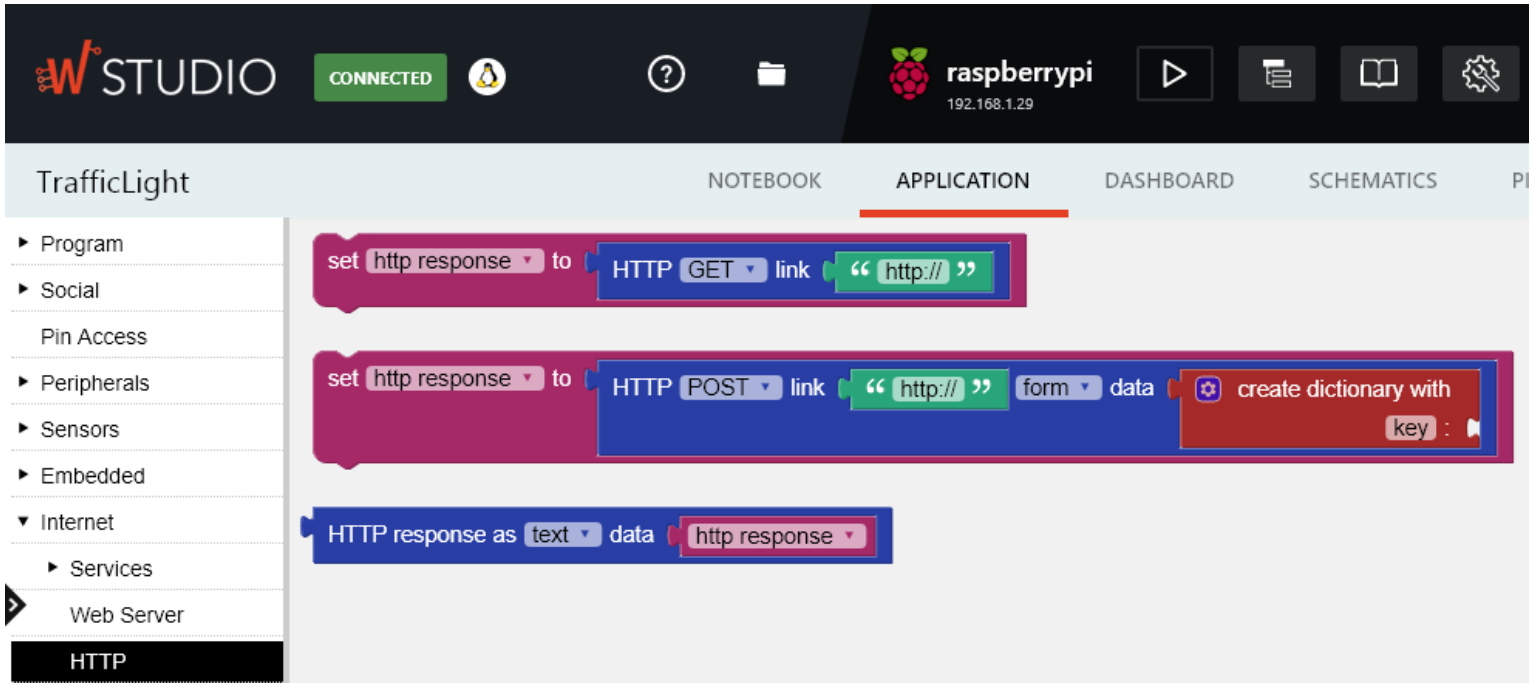
Coding with Blocks (7)

Embedded allows manipulating some actuators in UniPi, Pi Camera, Spark, UDOO Neo, and Arduino.



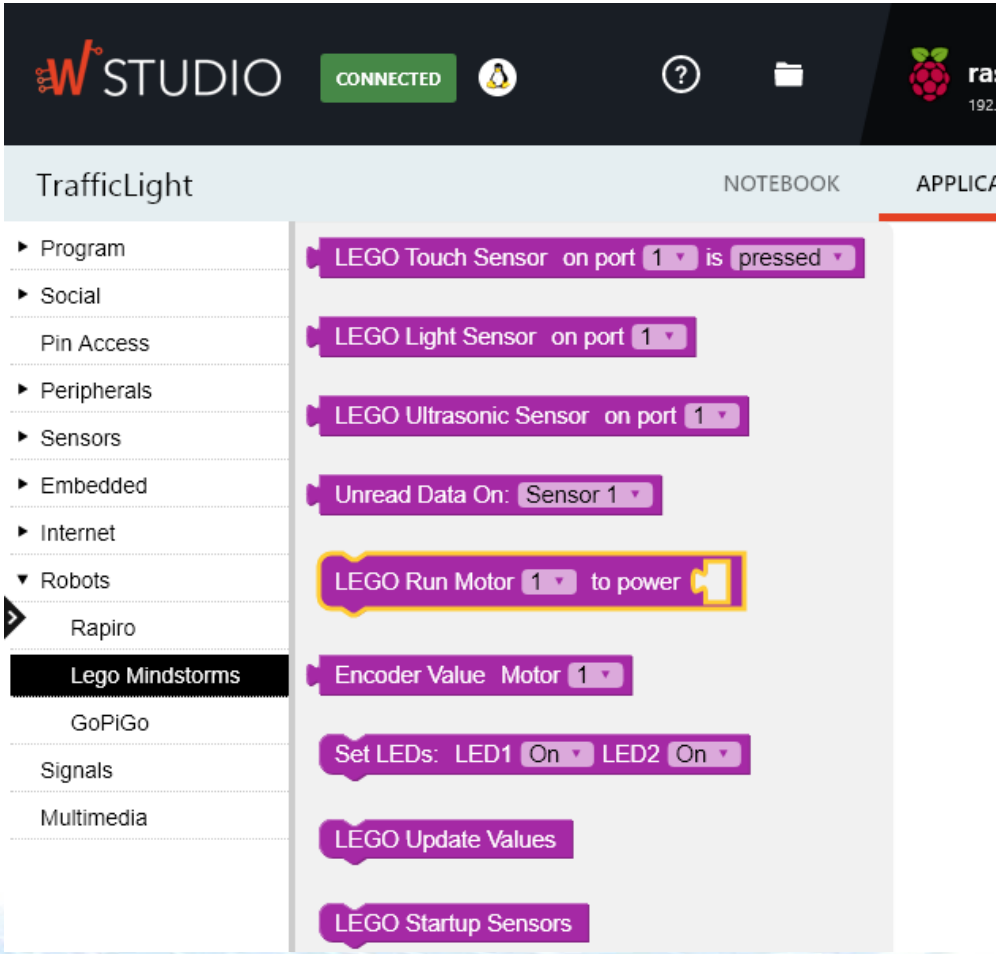
Coding with Blocks (8)

Internet contain blocks for HTTP RESTful communication and web services.



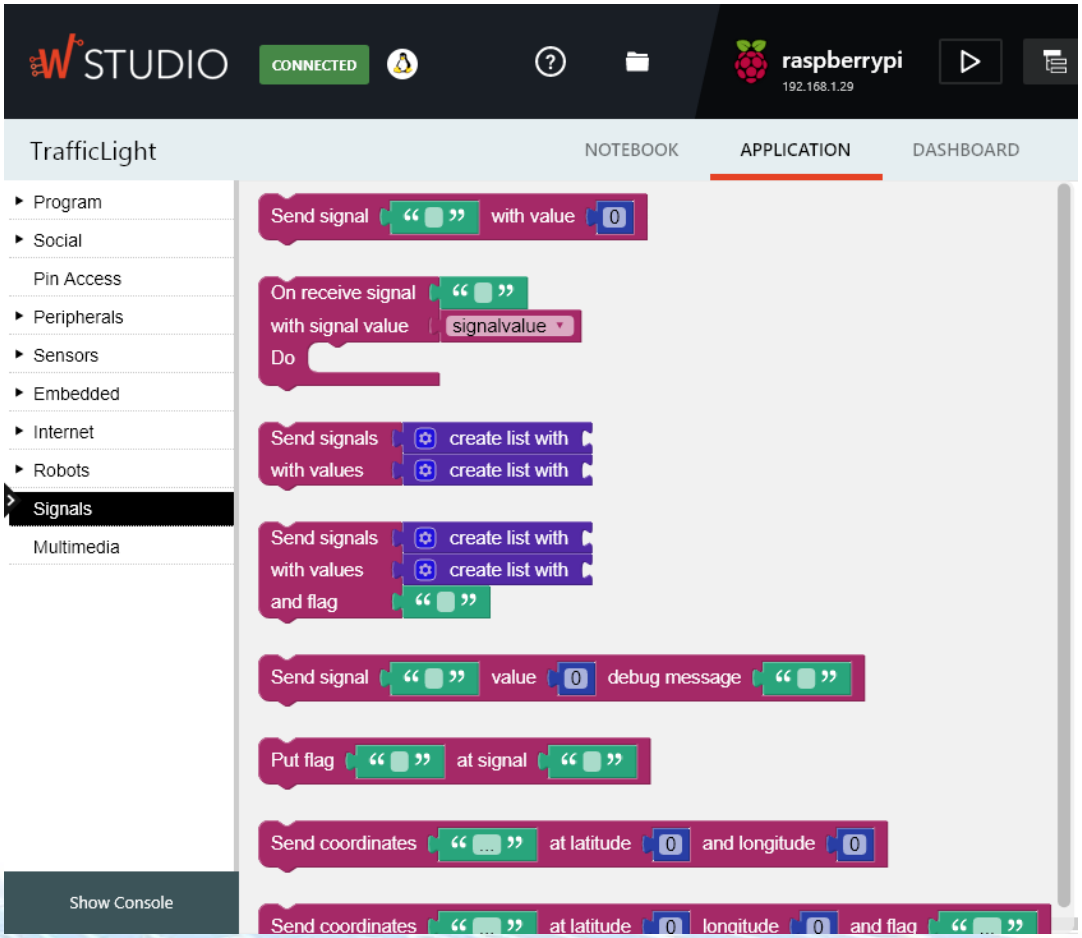
Coding with Blocks (9)

Robots contain blocks for interacting with Rapiro, Lego Mindstorms, and GoPiGo robots.



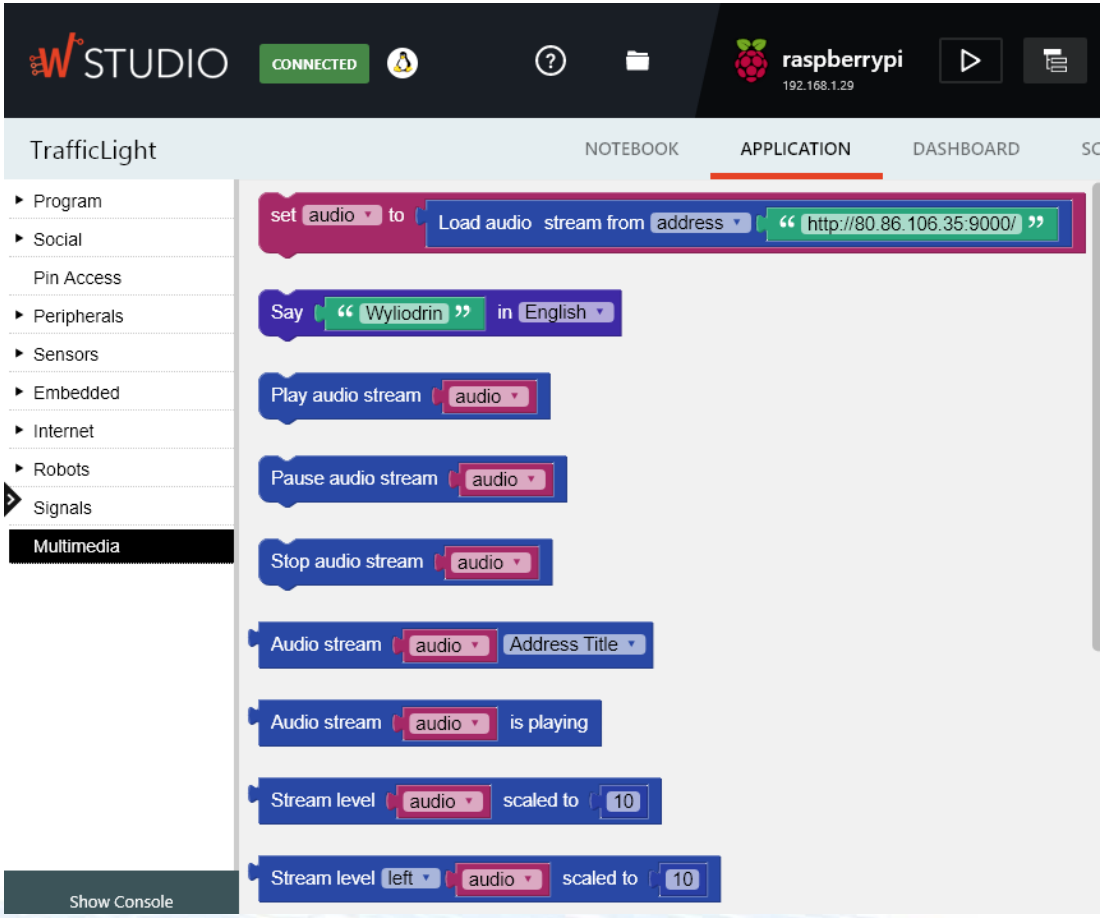
Coding with Blocks (10)

Signals can be utilized to create time-based or real-time output the sensor values. These output can be visualized later in the “Dashboard” tab.



Coding with Blocks (11)

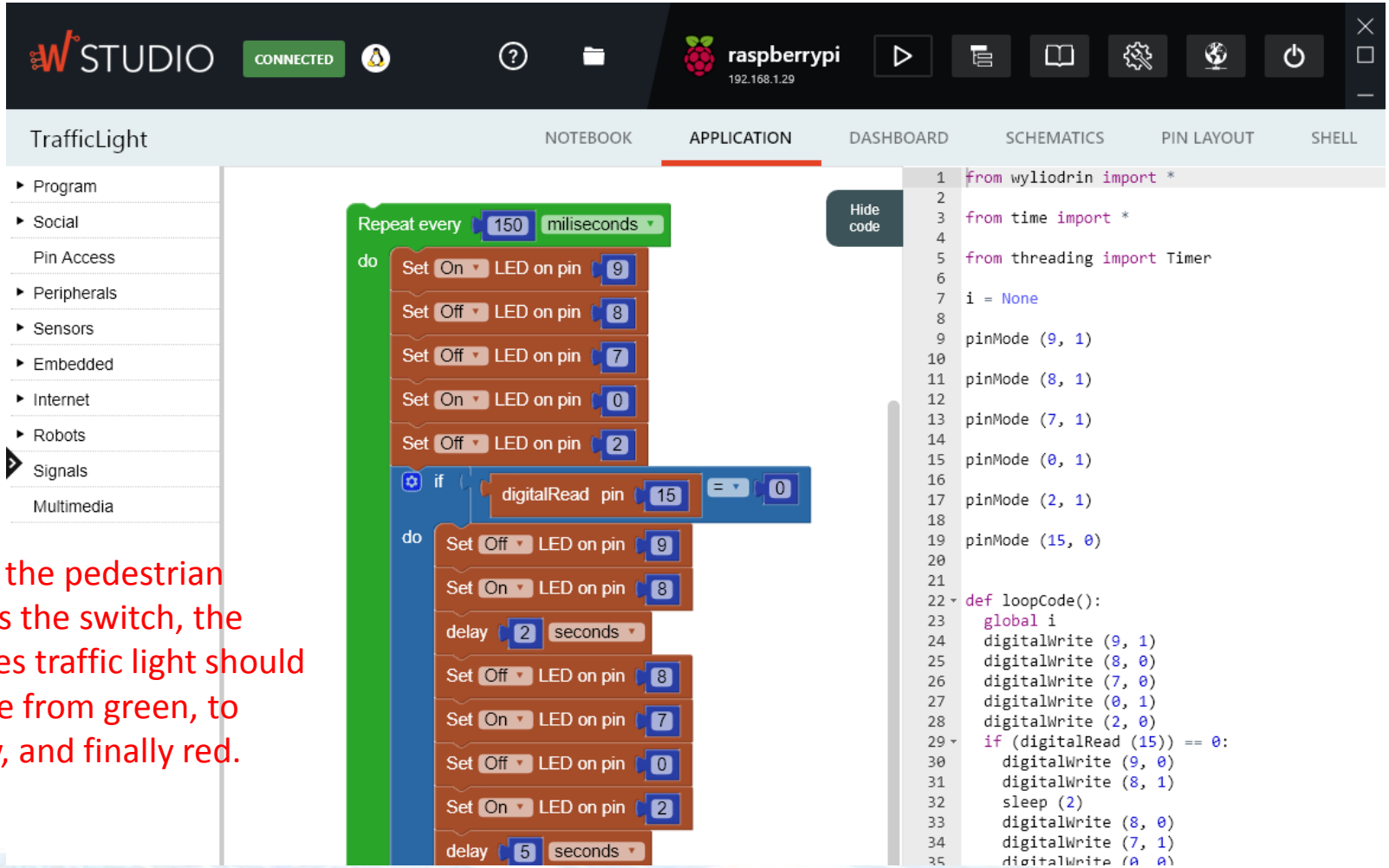
Multimedia allows audio management, streaming, and computer-based speech in your board.



Implementation of Traffic Light Simulator

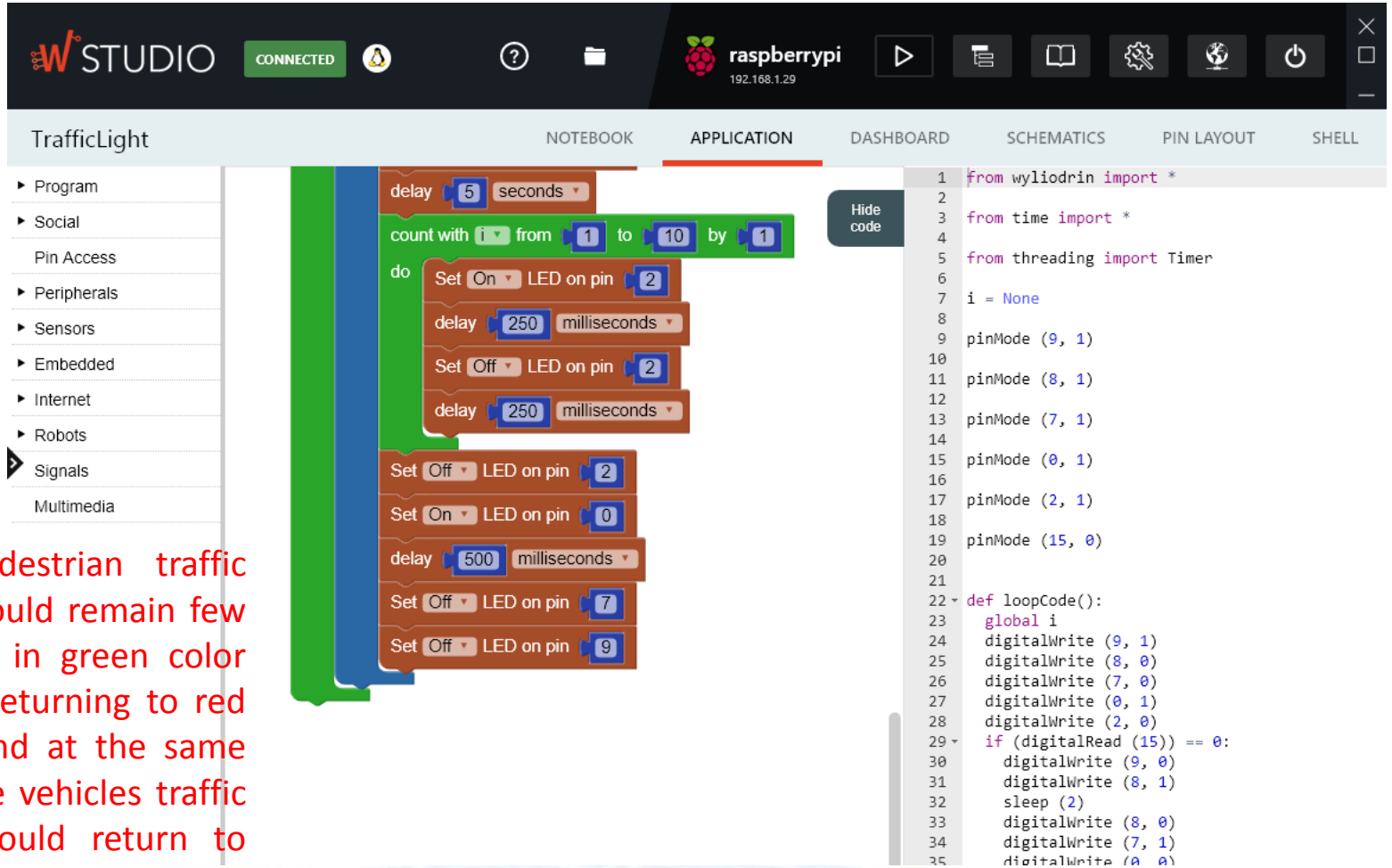
After this review of coding with blocks, let's continue with the implementation of our traffic light simulator…

Implementation (1)



When the pedestrian pushes the switch, the vehicles traffic light should change from green, to yellow, and finally red.

Implementation (2)



The pedestrian traffic light should remain few seconds in green color before returning to red color, and at the same time the vehicles traffic light should return to green color.

Implementation (3)

- You can play with timing/delay in order to make your traffic light more realistic.
- And print some messages about the status of the traffic lights in the console.
- After you finish your implementation, please demo to TA.

CHECKPOINT 4!

Apendix: VM & Software Tools

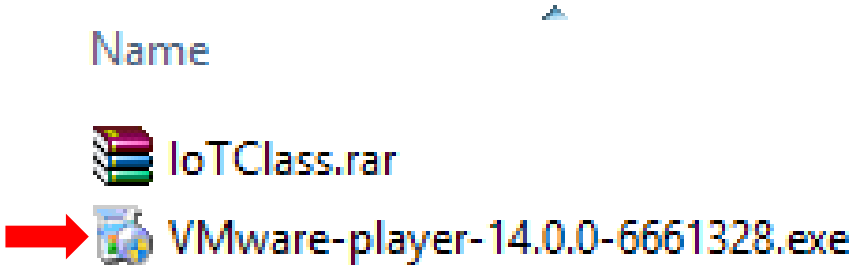
物聯網技術與應用 IoT/M2M Technologies and Applications

國立交通大學資訊工程系
Department of Computer Science
National Chiao Tung University

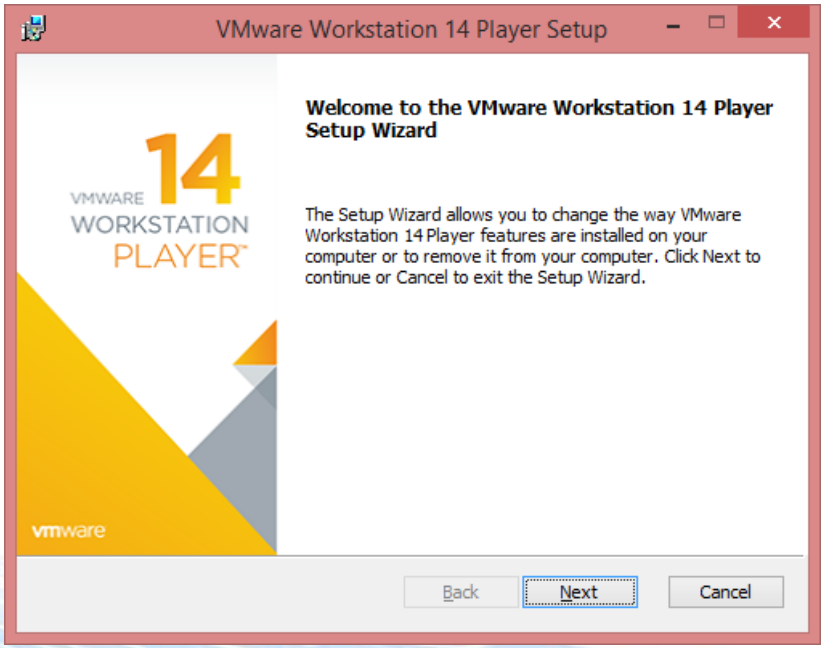
September 21, 2018

Install VMware

- Copy TA's files to your laptop

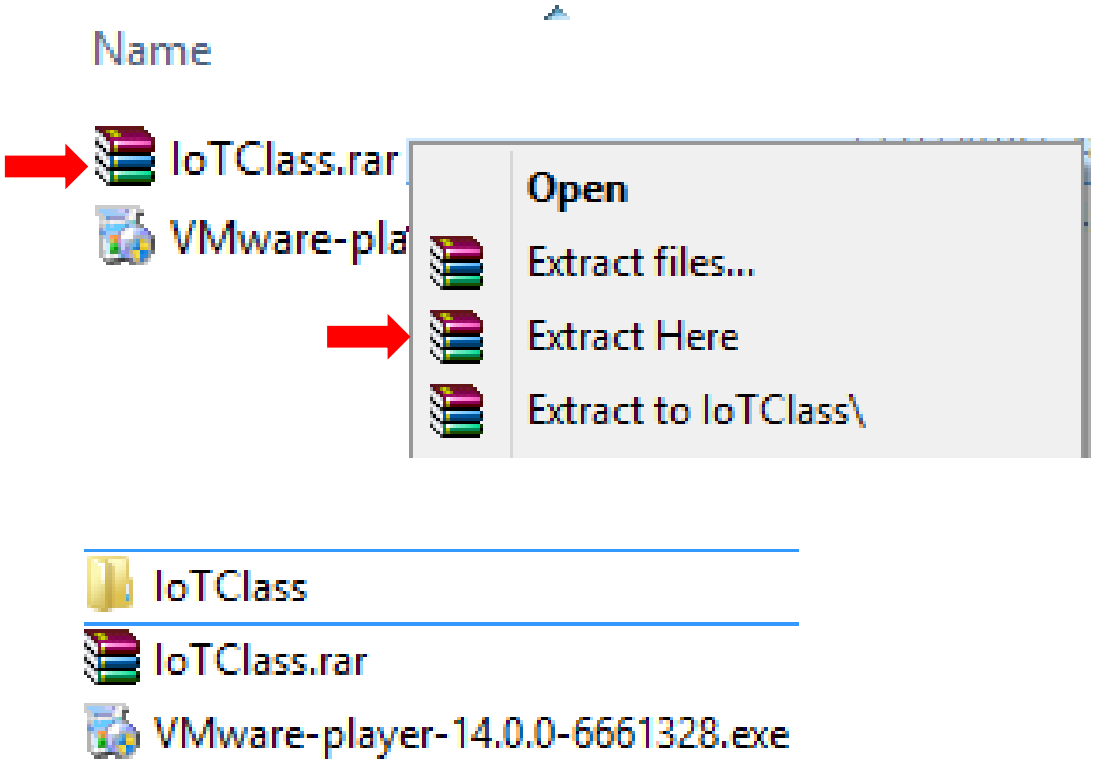


- Install VMware-Player



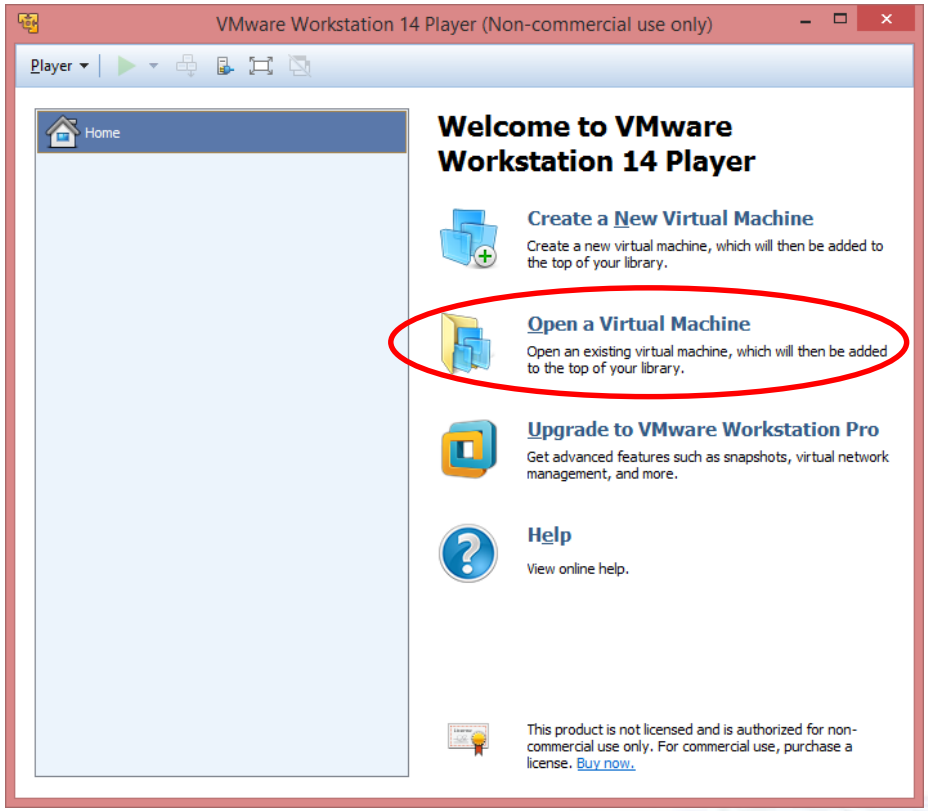
Extract the rar file

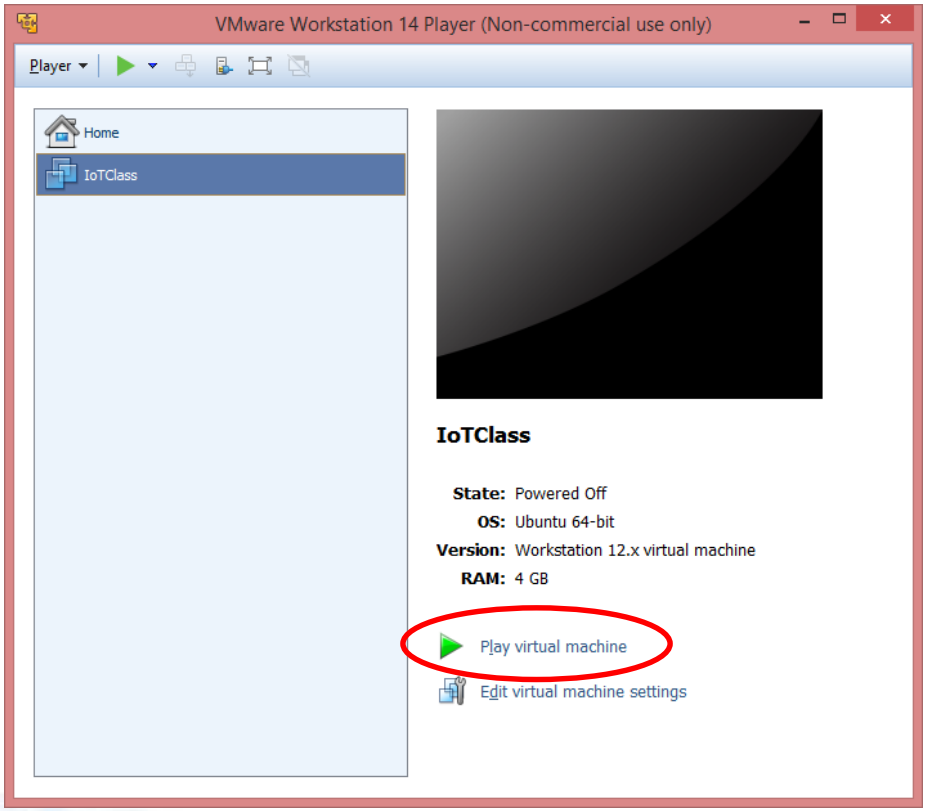
- Extract the IoTClass VM.



Run the VM

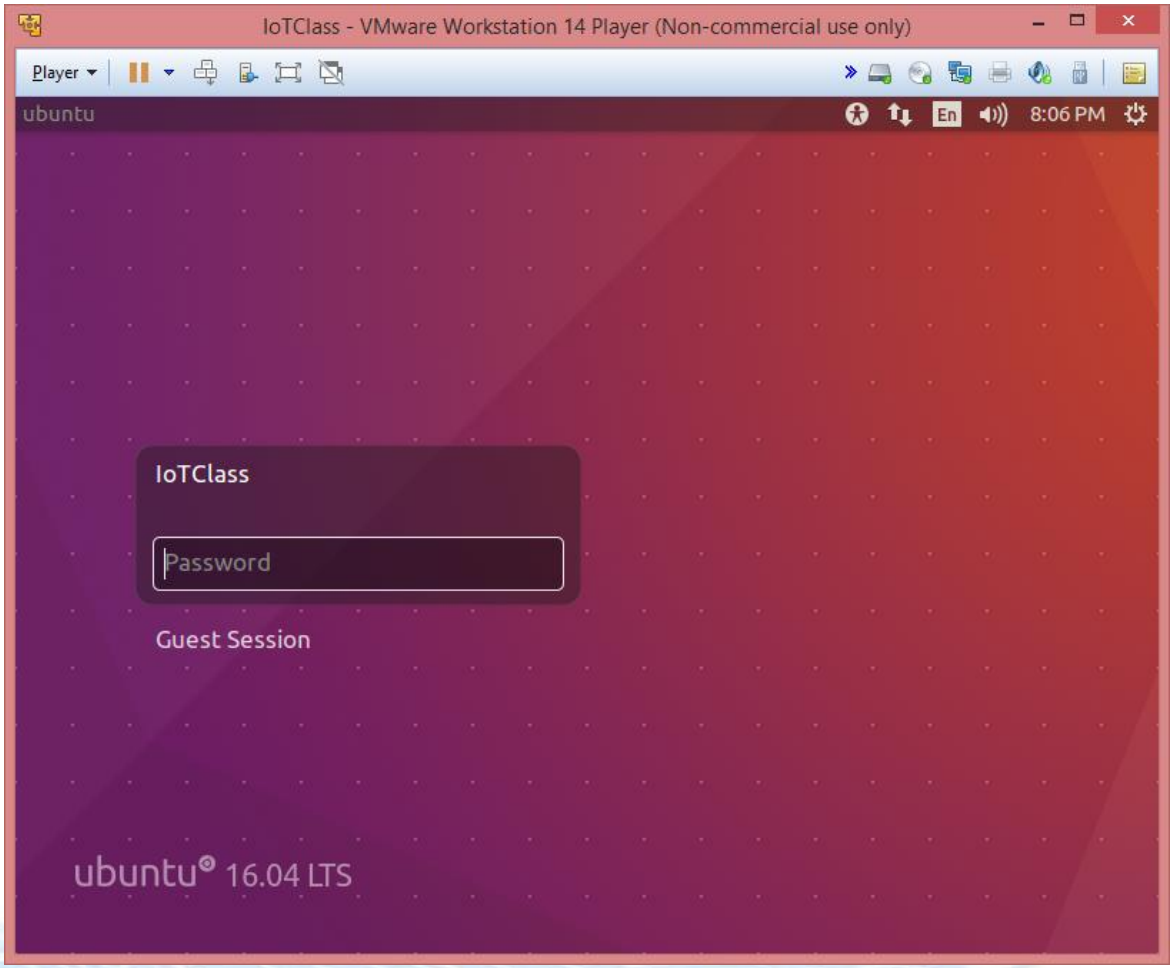
- Run VMware-Player, open the IoTClass VM, and start it.





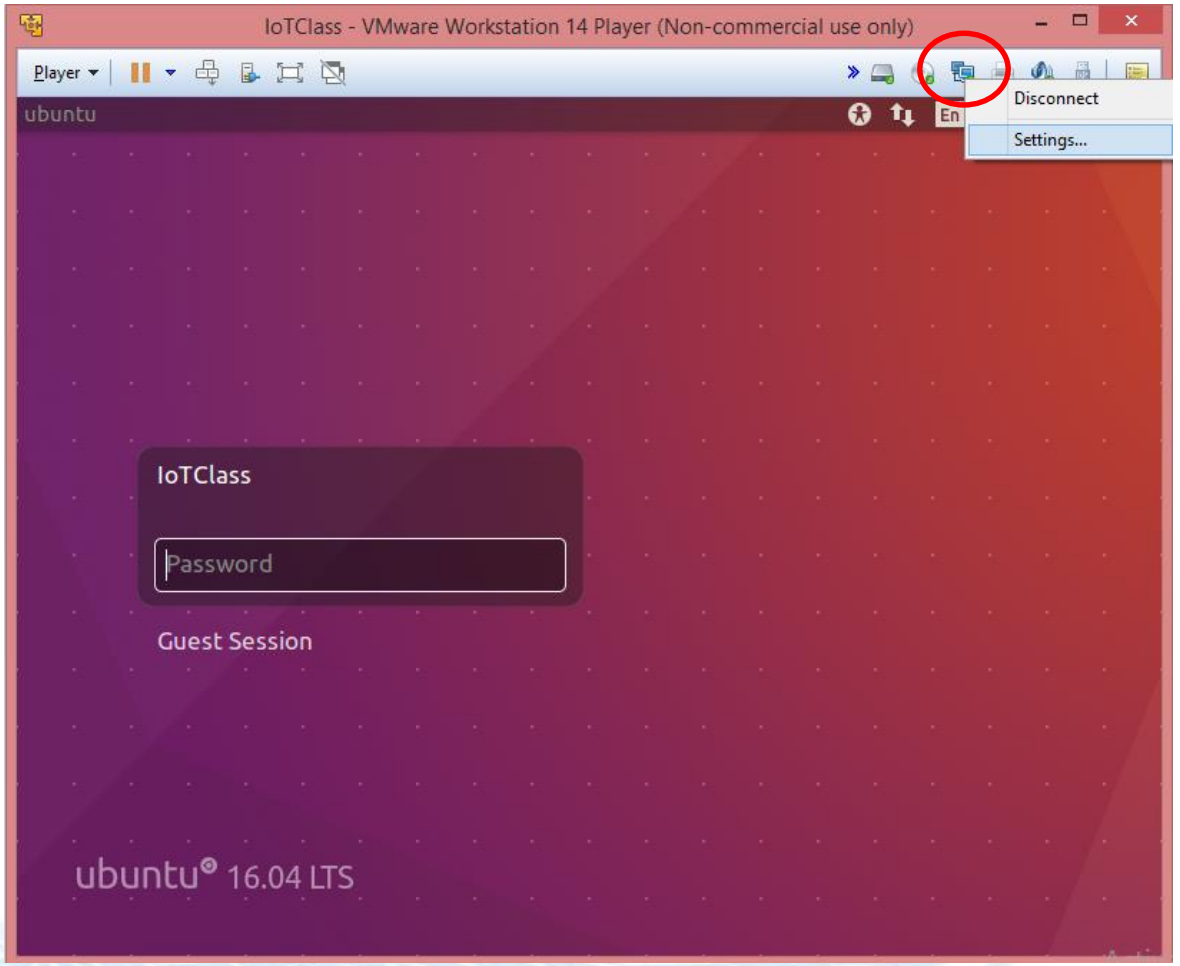
Login into the VM

- Password: iotclass



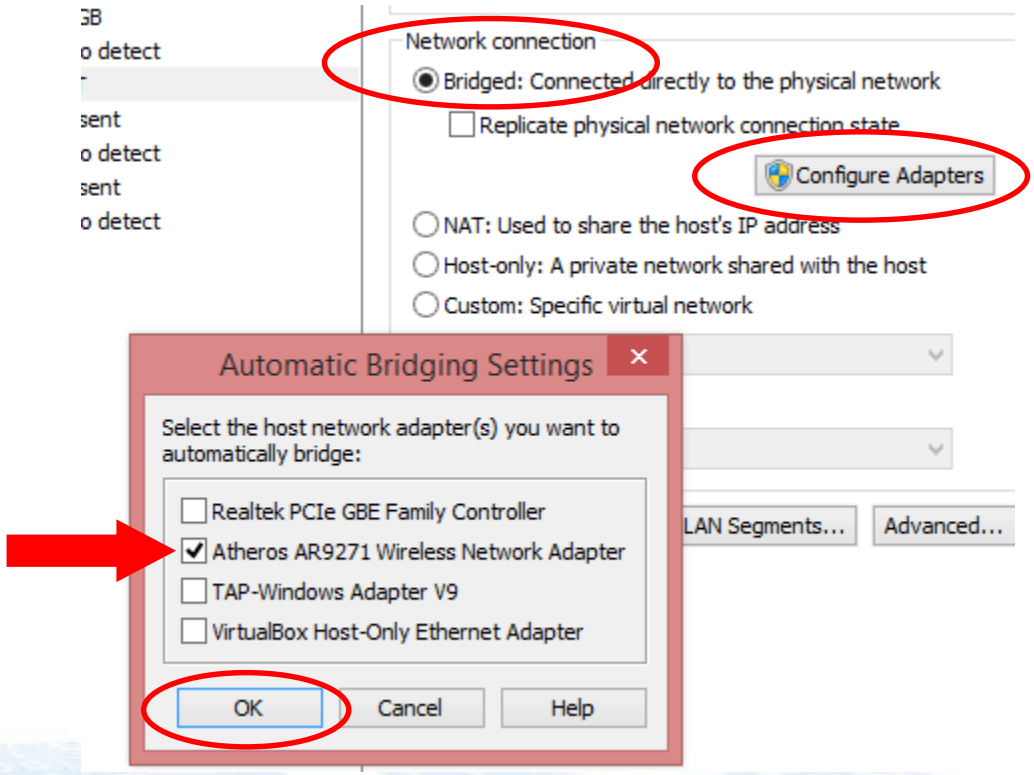
Setup Bridged Network (1)

- Right click on the network icon and choose Settings…



Setup Bridged Network (2)

- Choose **Bridged** radio button.
- Push **Configure Adapters** button.
- Mark your **wireless interface** name checkbox.



Content of the VM

- TA has installed the following software in the VM:
  - Java-JRE
  - Wyliodrin STUDIO
  - OM2M
  - Nodejs
  - node-red
  - contrib-oM2M.gz – a node-red library written by NCTU-LAAS
  - Postman