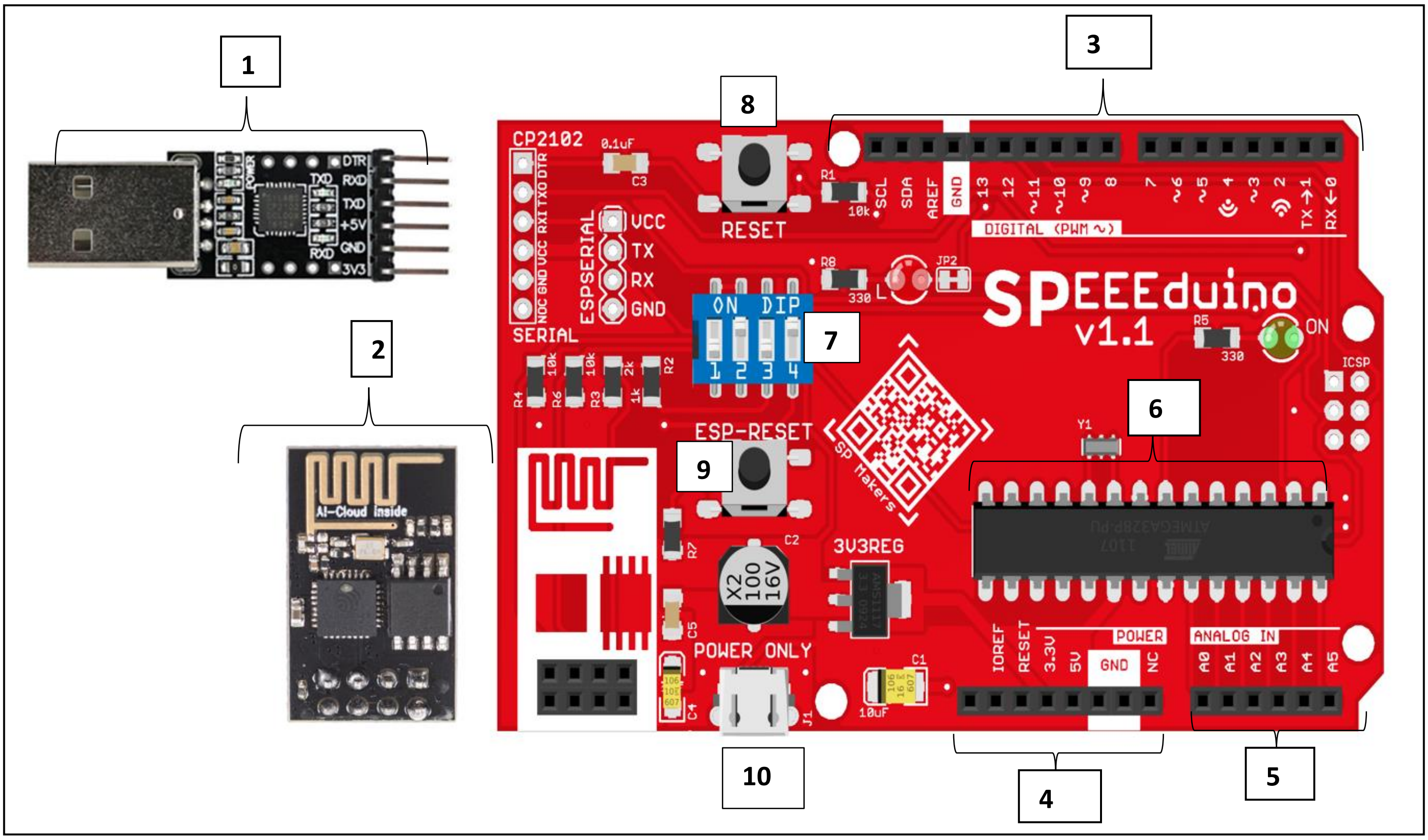
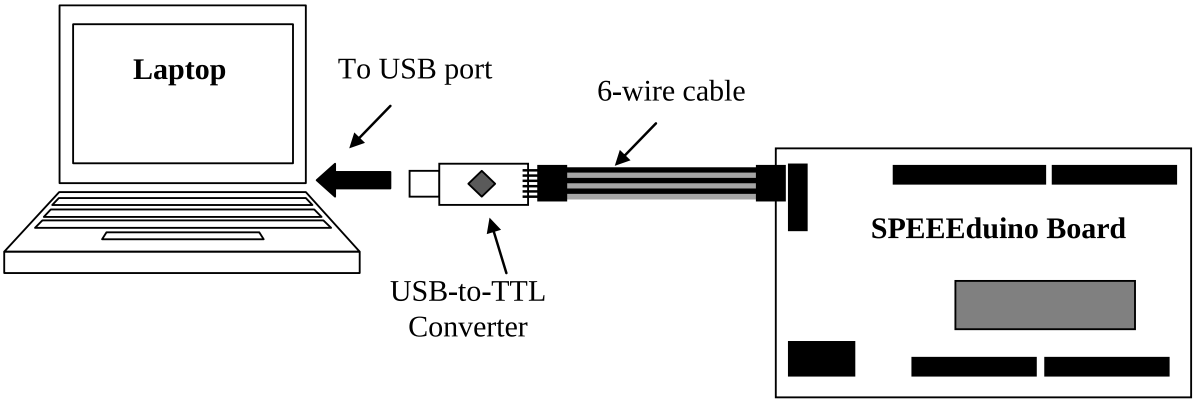
e5 comms: SPEEEduino + ESP8266

This experiment assumes Arduino IDE is installed on the computer/laptop, SPEEEduino is set up, and SSID&Password for SPEEEduino to connect via WiFi.

Recap: The SPEEEduino





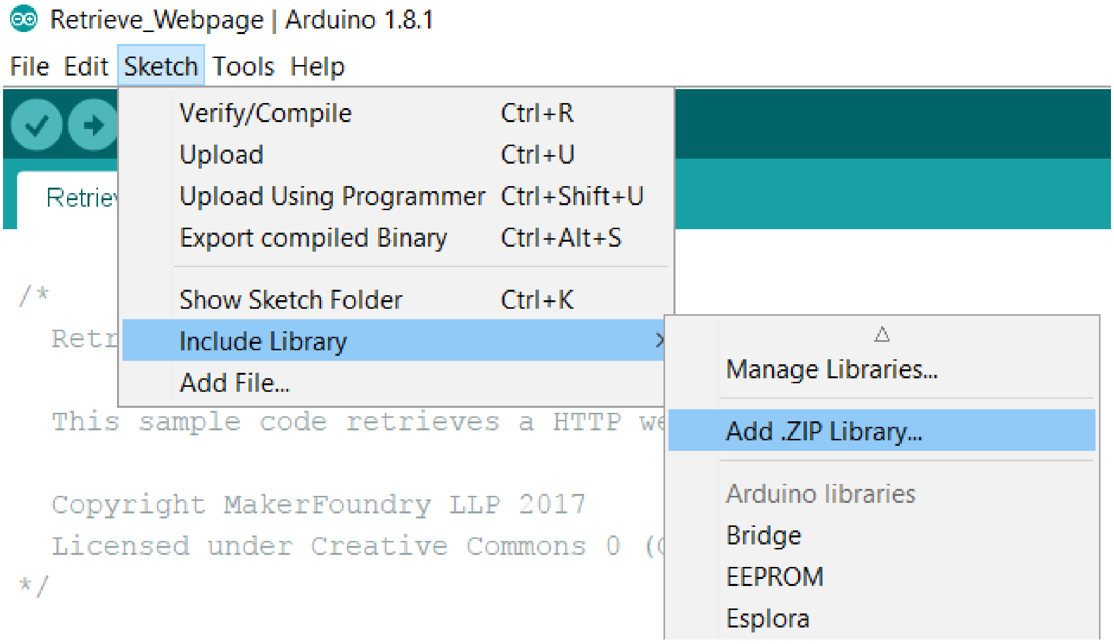
Connection diagram between SPEEEduino and Computer

Task2: Setup Driver for USB-TTL (if not yet completed)

1. Plug in the USB TTL converter into host computer
2. Determine USB TTL driver is installed on host computer
    a. Open the Device Manager window.
    b. Under the “Ports (COM & LPT)” directory
    c. Locate “CP2102 USB to UART Bridge Controller”
    d. Logo with exclamation mark on device needs the driver to be installed.
3. The USB TTL used is manufactured by Silicon Labs
4. Part number of USB-TTL is useful to identify the drivers to install
5. Determine the OS of the host computer
6. Go to https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers and download the zip file that meets the host’s operating system.
7. Unpackage the zip file to folder of choice
8. Return to the "Device Manager" of computer (with USB-TTL converter plugged in).
9. Right-click "CP2102..." and choose "Update driver".
10. Navigate to the folder where the CP2102 driver is stored.
11. The computer will install the driver for the CP2102 USB-TTL converter.
12. Driver installation is completed, the newly added device will be renamed to “Silicon Labs CP210x USB to UART Bridge (COM?)”.
13. Observe the COM port number (e.g. COM3).
14. SPEEEduino board is connected to the computer via this COM port

Task3: Setup SPEEEduino Low Level Library (if not yet completed)

1. Navigate to https://drive.google.com/file/d/0B1b6Ob0_FnFLbHZMTGREMXFmQU0/view
2. Download the SPEEEduino Low Level Library
3. Install the library to Arduino IDE
    a. Click on Sketch->Include Library->Add .zip library



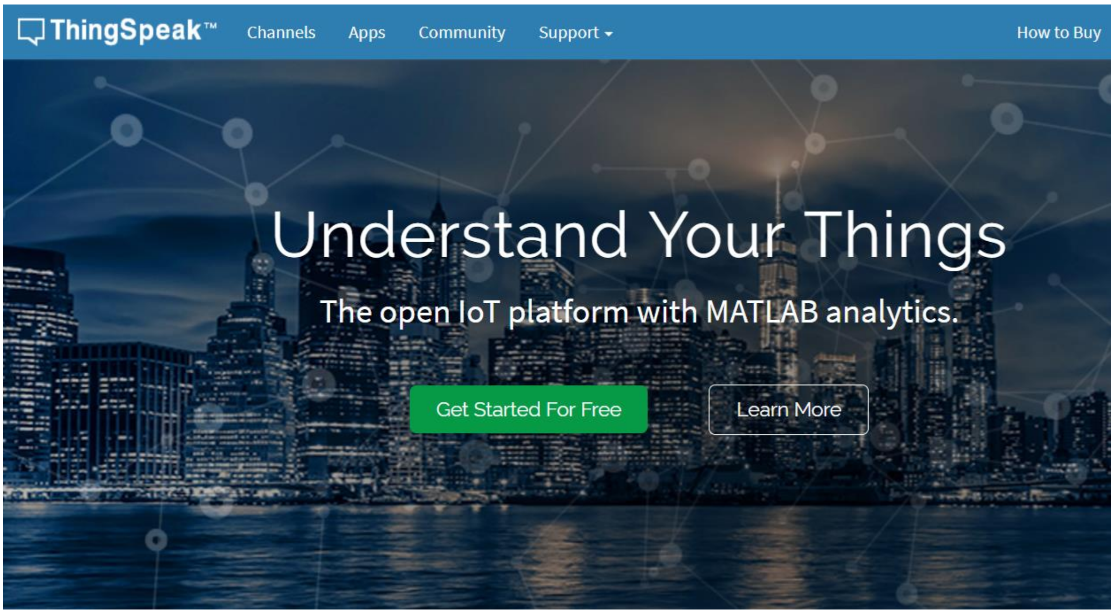
Retrieve_Webpage | Arduino 1.8.1
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries...
Add .ZIP Library...
Arduino libraries
Bridge
EEPROM
Esplora
Retrie

```
/*
  Retr
  This sample code retrieves a HTTP we
  Copyright MakerFoundry LLP 2017
  Licensed under Creative Commons 0 ((
*/
```

b. Locate SPEEEduino Low Level Library zip file
c. Click add
d. Verify library is added

Task5: Signup for Thingspeak account

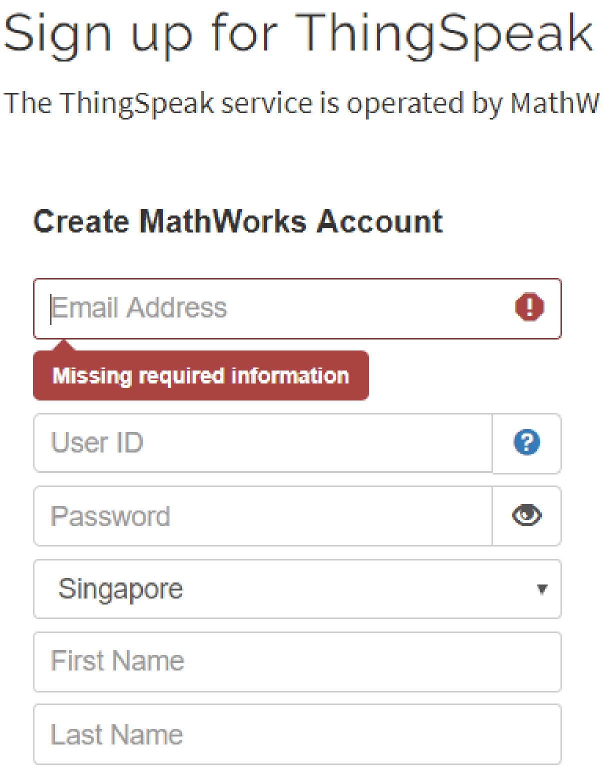
1. Navigate to www.thingspeak.com



Understand Your Things

The open IoT platform with MATLAB analytics.

2. Click on Get Started For Free, and fill in the sign up form.



Sign up for ThingSpeak

The ThingSpeak service is operated by MathW

Create MathWorks Account

Email Address

Missing required information

User ID

Password

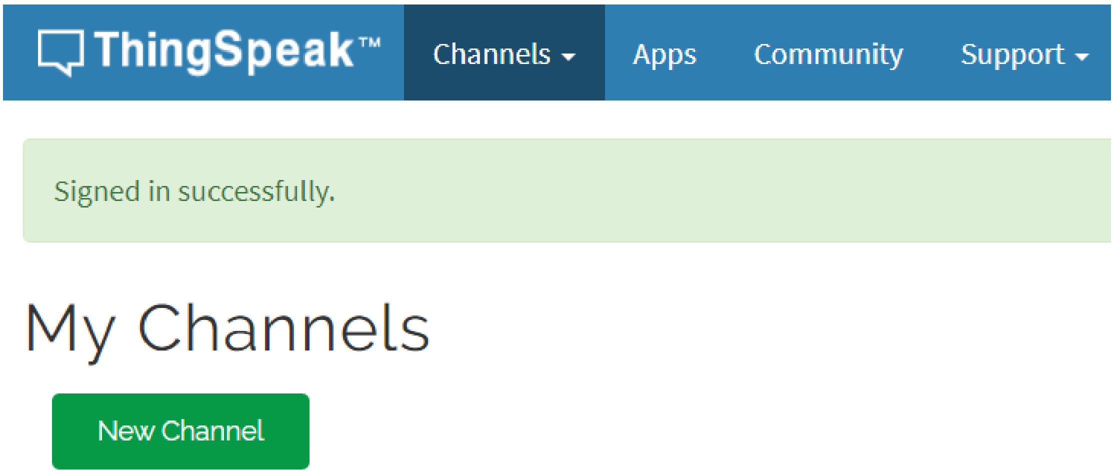
Singapore

First Name

Last Name

a. Remember to verify the account by clicking on the verification URL sent by MathsWork to the email used to create account

3. Create new channel

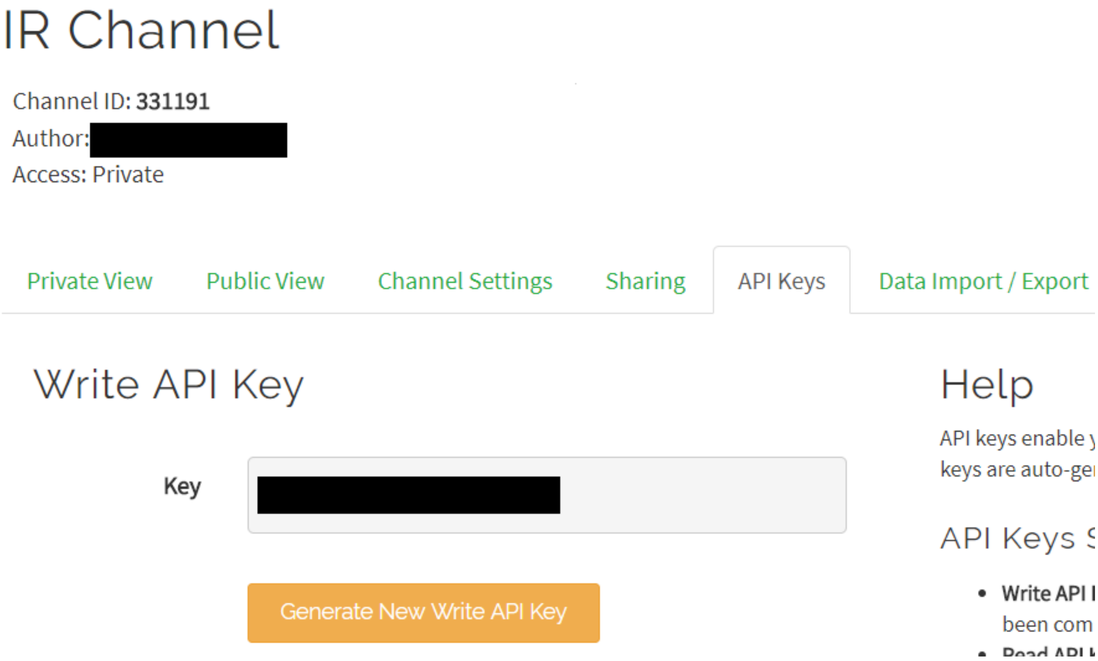


Signed in successfully.

My Channels

   a. Click on Channels->New Channel
   b. Fill in Name, and field1 used
   c. Take note on the limit of channels and fields on free account

4. Locate Write API Key



IR Channel

Channel ID: **331191**
Author: [redacted]
Access: Private

Write API Key

Key [redacted]

   a. Channel->API Keys
   b. Copy Write API Key into notepad
   c. This Write API key will be used by MCU to update ThingSpeak Channel
   d. At each Generate New Write API Key, all IoT devices with old Write API Key need to be updated with the newly generated Write API Key.

Task 6: Send data to ThingSpeak using SPEEEduino

1. Connect USB-TTL to SPEEEduino and computer
2. Navigate to File -> Examples -> SPEEEduino_Low_Level_Library -> IR_Sensor_ThingSpeak
3. Modify the SSID, Password to the correct parameters
4. Modify the Write API key to the correct parameter from task4

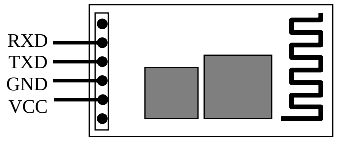
5. Download the program to SPEEEduino
6. Observe the output at Thingspeak channel at 30seconds interval
7. Modify this sample code to read data from LM35 connected on A0 of SPEEEduino, and transmitted to Thingspeak
8. Observe the output at Thingspeak channel at 30seconds interval

e5 comms: Bluetooth

Bluetooth modules are used for short distance (<30 metres), low volume, and wireless data communications. Many electronic devices are equipped with Bluetooth connectivity:

Mobile phones, Laptops, Bluetooth headphones/speakers

Adding Bluetooth module to SPEEEduino introduces wireless connectivity to SPEEEduino board, allowing it to communicate with other smart devices.



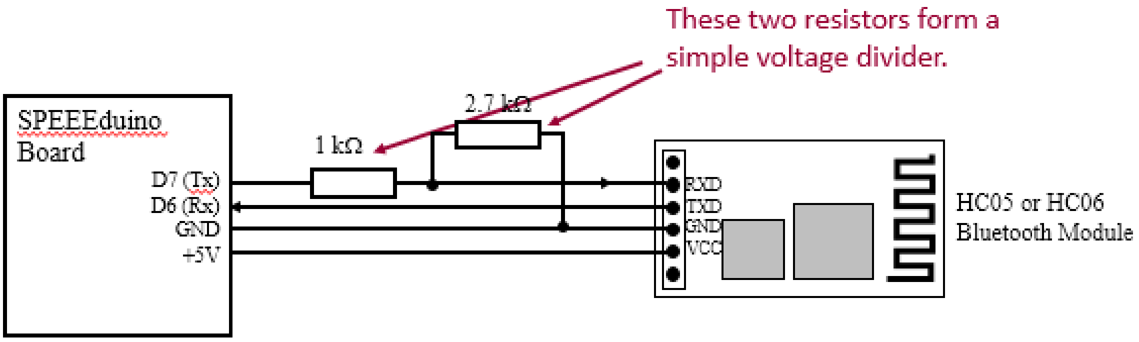
HC-05 Bluetooth module has 6 pins, but only 4 used: TXD, RXD, VCC, GND

HC-05 module are supposed to work with logic signal of 0V (logic ‘0’) and 3.3V (logic ‘1’). SPEEEduino board, where logic ‘1’ = 5V.

Q1: What is required to make TTL compatibility between SPEEEduino and HC05?

____

Task1: make a voltage divider circuit for HC05 and SPEEEduino



These two resistors form a simple voltage divider.

Task2: Pairing of HC05 with ArduDroid (Android App)

Note: Please take note on the physical address of the HC05, and select the corresponding HC05 device using your ArduDroid android App

1. Launch the ArduDroid App on your Android phone.
2. Press the "Menu" button of your Android phone.
3. Press "Connect me to a Bluetooth device". Choose your HC05 device.
4. If successful, the App should show a pop-up message, saying "Connected to _____". The LED on the HC05 module will start blinking in the pattern of “0..0……0..0…….”.

Task3: PBIL

1. Locate open source example for HC05 for Arduino Uno
2. Download and/or modify the example for the use with SPEEEduino
3. Use ArduDroid in Task2 to control pin13 of LED.