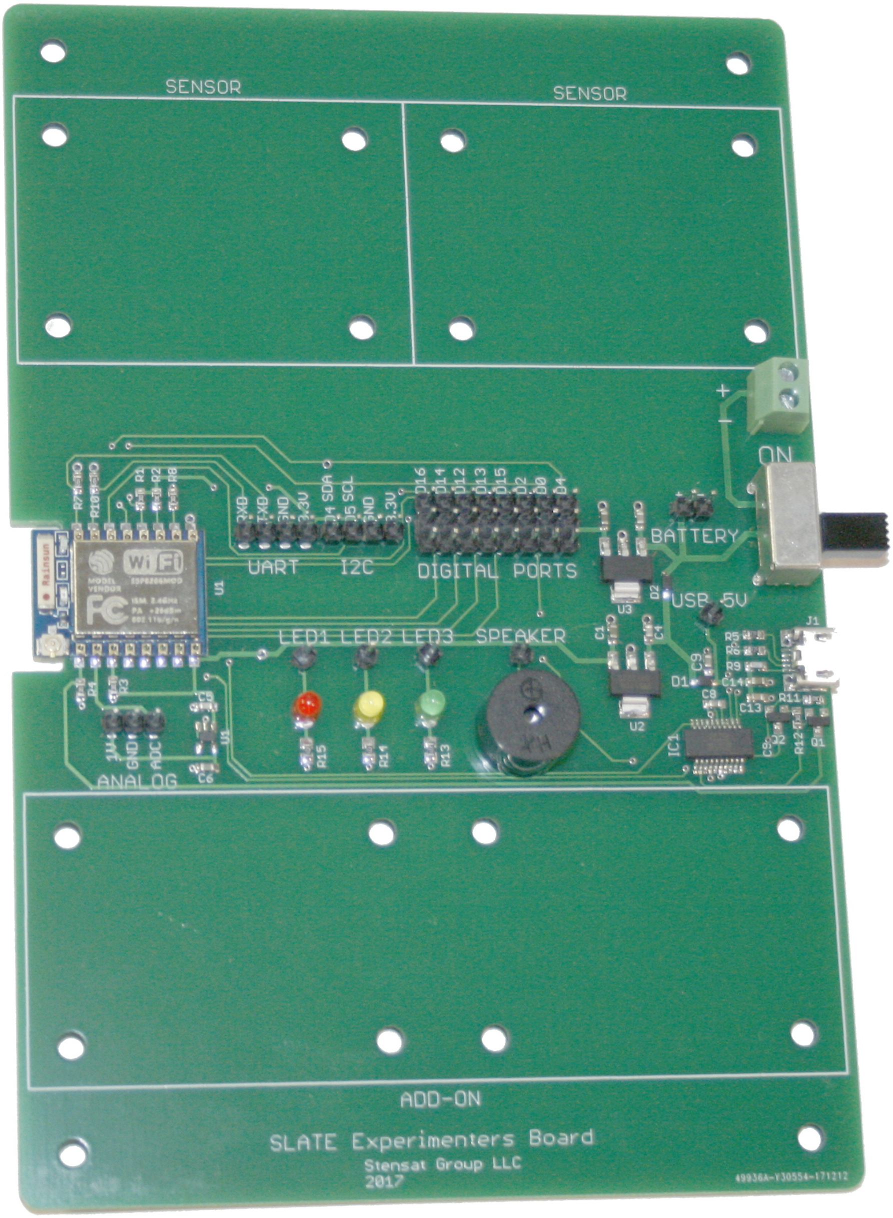
Sten-SLATE ESP Kit



Legal Stuff

Stensat Group LLC assumes no responsibility and/or liability for the use of the kit and documentation.

There is a 90 day warranty for the Sten-SLATE ESP kit against component defects. Damage caused by the user or owner is not covered.

Warranty does not cover such things as over tightening nuts on standoffs to the point of breaking off the standoff threads, breaking wires off the motors, causing shorts to damage components, powering the motor driver backwards, plugging the power input into an AC outlet, applying more than 9 volts to the power input, dropping the kit, kicking the kit, throwing the kit in fits of rage, unforeseen damage caused by the user/owner or any other method of destruction.

If you do cause damage, we can sell you replacement parts or you can get most replacement parts from online hardware distributors.

This document can be copied and printed and used by individuals who bought the kit, classroom use, summer camp use, and anywhere the kit is used. Stealing and using this document for profit is not allowed.

If you need to contact us, go to www.stensat.org and click on contact us.

Electrostatic Warning

The Sten-SLATE Experimenters Kit is sensitive to static electricity. Handle with care and be careful to discharge yourself before handling. Use the silver anti-static bag as an anti-static mat. Touch the bag before handling the processor to reduce the risk of damage.

Work on a flat surface, not the bed, a carpeted floor, other anywhere where static electricity can build up. The warranty does not cover damage due to electrostatic discharge. Fall and winter are times when electrostatic electricity is more prevalent.

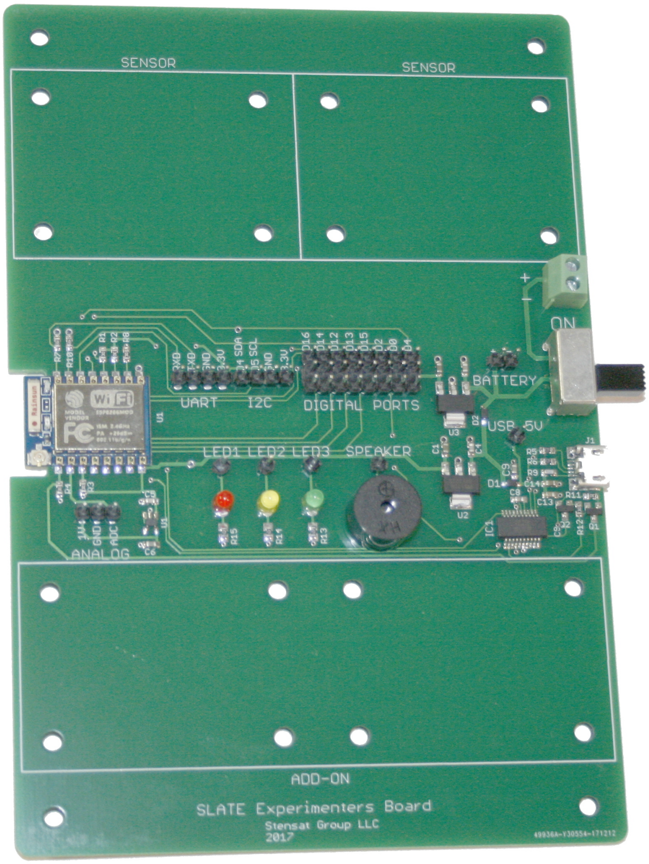
Discharge yourself by touching a metal door frame, a metal lamp that is properly grounded, metal furniture, or some metal structure that is not connected to power.

Parts List

- Electronics Experimenters Base Board
- Sensor 1 Board with Ultrasound, light and temperature sensor
- Sensor 2 Board with accelerometer
- 20 F/F Jumpers
- 8 1/2” 4-40 screws
- 16 4-40 nuts
- 4 rubber pads
- Micro USB cable

Developer Kit

The Sten-SLATE ESP Experimenters Kit includes a 32-bit ARM processor with at least 512 Kbyte of program memory and 64 Kbyte of data memory. The processor operates at 80 MHz. WiFi is integrated into the processor.



The kit includes three LEDs and a speaker integrated. There are positions to install several sensor boards.

There are digital connections that can also operate up to eight servo motors.

There are two digital signals organized with power and ground to provide a convenient access to the I2C bus.

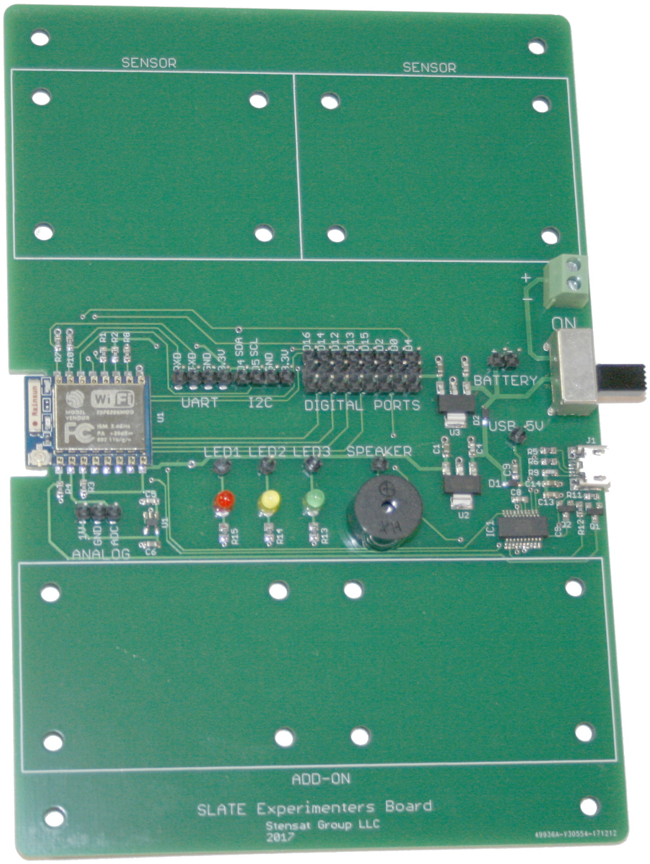
There are another two digital signals organized also with power and ground to provide convenient access to the UART interface. These interfaces will be explained later.

Power is applied through the USB port or through an external battery. For external power, a power switch is made available and a terminal screw block is provided to connect external power. Up to 12 volts can be applied.

The kit is designed to be modular so it can be more easily integrated.

Developer Kit

Take the four rubber pads and mount them into the four corners of the SLATE board. Make sure to not cover any holes.



Overview

In this section, you will be introduced to the processor board electronics and the arduino software.

At the end of this section, you will be able to write software, control things and sense the environment.

Arduino software location: www.arduino.cc

ESP8266 Arduino library information: https://github.com/esp8266/Arduino

Software

The SLATE ESP Experimenters Kit uses the arduino software. This software allows you to write programs, compile them and upload them to the processor. It also allows you to interact with the software running on the processor. Only one program can be installed and run at a time. The processor is small and does not have an operating system. Embedded computers are designed to perform a specific task and not operate like a desktop computer or laptop.

More information about the arduino software can be found at **http://www.arduino.cc**

Loading and Configuring Arduino Software

Download the proper version of the Arduino software from www.arduino.cc. Once downloaded and installed, start the program. Under the **File** menu, select **Preferences.** Find the text entry space to the right of **Additional Board Manager URLS:**

Enter the following into the text area

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Click OK.

Select the menu **Tools** and then **Boards**: and then **Boards Manager**.

A window will open. Scroll down until **esp8266** is located. Click on it and click Install.

The software will load the compiler for the processor board.

For Mac OSX users, you need to install python 3 to run Arduino. Instructions are on page 58.

Configuring The Software

Select the “**Tools**” menu again and then “**Board:**”.

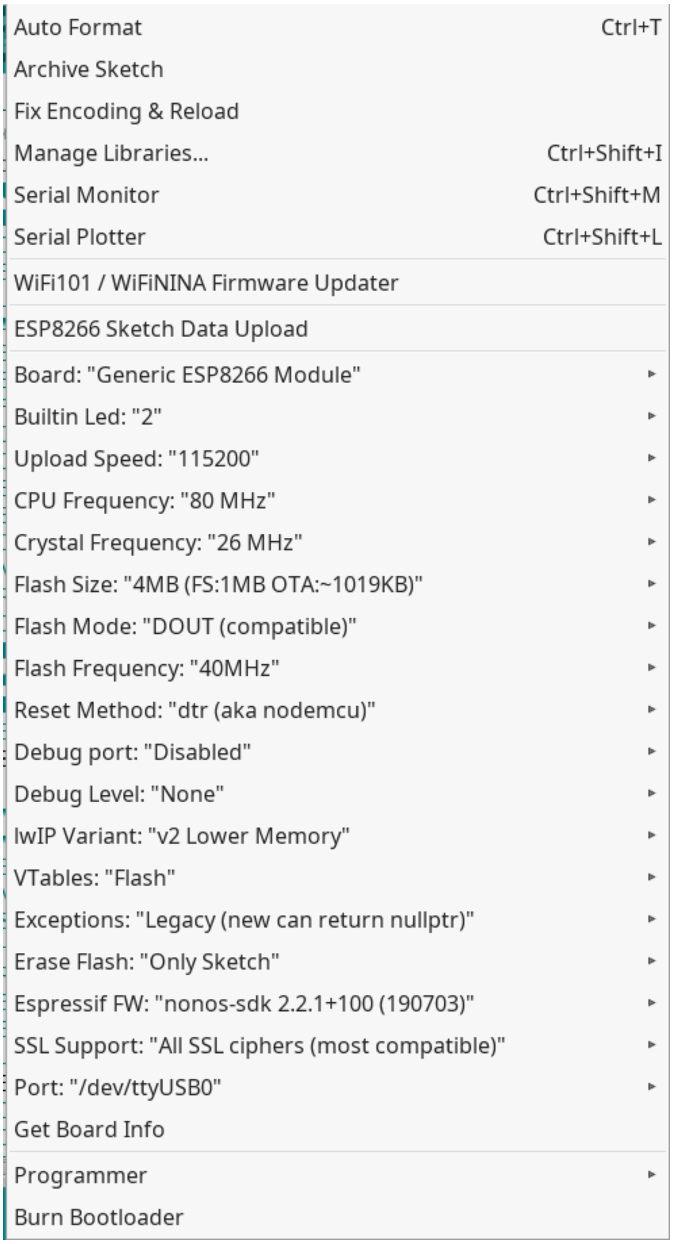
Select “**ESP8266 Boards**”

Select “**Generic ESP8266 Module**”

Go back to “**Tools**” menu and select “**Reset Method**”. Select “**dtr (aka nodemcu)**”

Nothing else needs to be changed. This completes setting up for the processor board.

To the right is a representaion of the Tools Menu. All selections should look like it. Only the Port: selection may be different. The menu may change as newer versions of the software become available. The same setting needs to be set.



Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
ESP8266 Sketch Data Upload
Board: "Generic ESP8266 Module"
Builtin Led: "2"
Upload Speed: "115200"
CPU Frequency: "80 MHz"
Crystal Frequency: "26 MHz"
Flash Size: "4MB (FS:1MB OTA:~1019KB)"
Flash Mode: "DOUT (compatible)"
Flash Frequency: "40MHz"
Reset Method: "dtr (aka nodemcu)"
Debug port: "Disabled"
Debug Level: "None"
lwIP Variant: "v2 Lower Memory"
VTables: "Flash"
Exceptions: "Legacy (new can return nullptr)"
Erase Flash: "Only Sketch"
Espressif FW: "nonos-sdk 2.2.1+100 (190703)"
SSL Support: "All SSL ciphers (most compatible)"
Port: "/dev/ttyUSB0"
Get Board Info
Programmer
Burn Bootloader

Selecting COM Port

The Arduino software communicates with the processor board for uploading code and interaction via a COM port with Windows and a /dev/tty.usbserial-Daxxxxxx with MAC OSX. Under Linux, the COM port is /dev/ttyUSBx where x is a number.

For Linux, the device driver is already part of the operating system.

MAC OSX should include a driver as part of the operating system.

The next two pages explain how to load the device driver for Windows and MAC OS X. Administrative privileges are required for installation. Under Windows, you must run the installation software as an administrator. Under MAC OS X, you will be asked for your login password.

Plug the Slate into the computers USB port.

Windows COM Port

Plug the SLATE into the USB port. If the Serial Port menu does not show any COM ports try the following: Go to http://www.ftdichip.com/Drivers/VCP.htm. In the table on the website, locate the row specifying **`Windows`**. Click on the link **`setup executable`** and download the software. Right click on the icon labeled **`CDM v2.12.00.....`** and select **`Run as Administrator`** in the menu that pops up. Follow the instructions to install. You have to run the installation program as an administrator or the driver will not install. Go back to Arduino and select the COM Port it gives you. Most likely, the COM port will be COM3. If there is more than one COM port, choose the higher number.

Macintosh OS X USB Driver Installation

Go to http://www.ftdichip.com/Drivers/VCP.htm. Select VCP driver (Virtual COM Port) for Mac 64 bit.

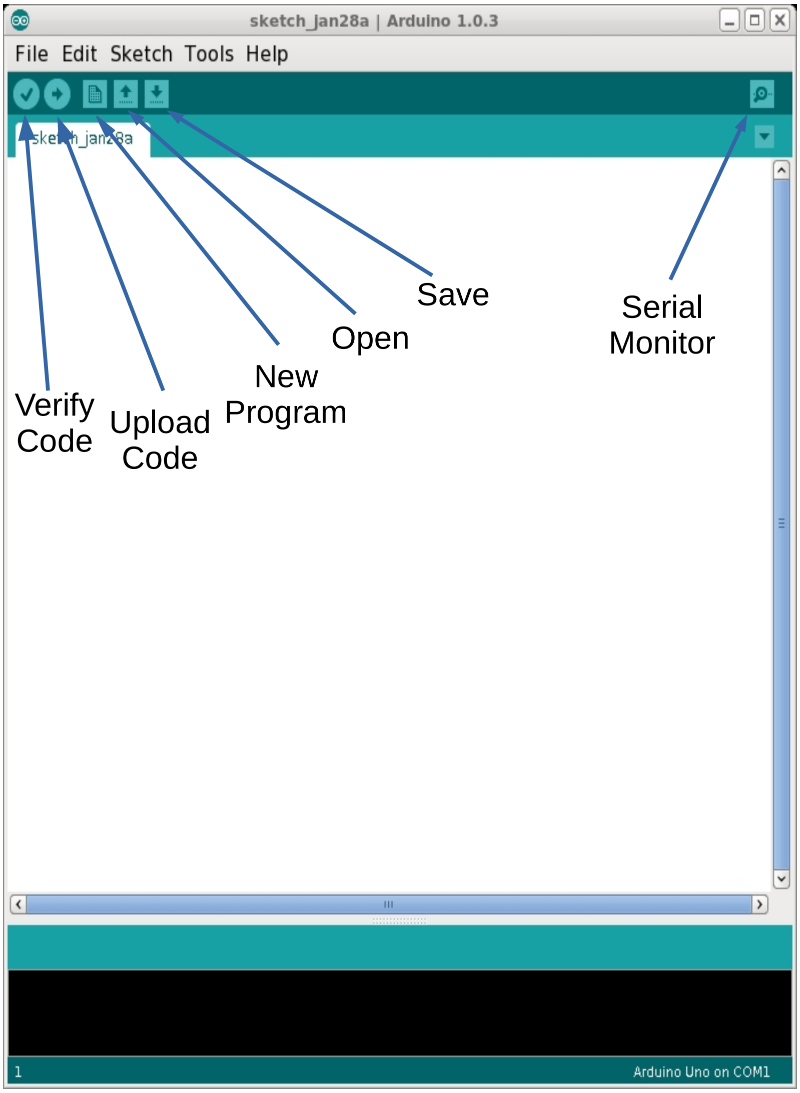
Download and double click on the file. A window will open showing two packages. Double click the package for the OS version you have. Follow instructions for the installation. You do need admin privileges to install. Go back to Arduino and select the Serial Port: /dev/tty.usbserial-Daxxxxxx The xxxxxx will be some combination of letters and numbers

Using Arduino

This is the arduino software. The software will let you enter programs and upload the code to the processor board. The large white area is where the code is entered. The black area below is where error messages will be displayed such as when there is an error in the code or the software cannot upload code for some reason.

The buttons below the menu have different functions. The first is called **Verify Code** and will compile the code and check for errors but not upload the code. The next button will do the same as the first but also upload the code. New Program button opens a new copy of the program allowing you to start writing another program. Open and Save are for opening and saving the code you have written.

Serial Monitor button opens a new window allowing you to interact with the processor. The Serial Monitor window allows the processor to display information and you to send information. This will be used quite a bit in this section.



First Program to Test

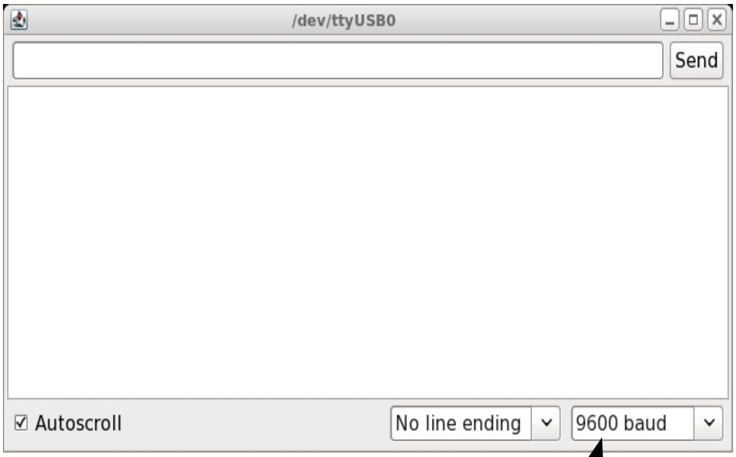
Enter the program in the editor on the right. **Do not copy and paste from the pdf file**. It doesn't work. The compiler is case sensitive so pay attention to capitalized letters. Don't forget to include a semi-colon after each statement as shown.

```
void setup()
{
  Serial.begin(115200);
}

void loop()
{
  Serial.print(“Hello World”);
}
```

Plug the processor board into the USB port. Click on the upload Code button to compile and upload the program. When the status message at the bottom of the window says done uploading, click on the serial monitor button. The Serial Monitor window will open. Most likely, the serial rate will be 9600. Change the serial rate to 115200. The selection is at the lower right corner of the serial monitor window.

Experiment by changing the message. Save your program. Pick a file name.



☑ Autoscroll   No line ending   9600 baud

Change to 115200

Serial Monitor Window

What are Functions

A function is basically a set of instructions grouped together. A function is created to perform a specific task.

The set of instructions for a function are bounded by the curly brackets as seen to the right.

The **`setup()`** function is used to initialize the processor board, variables, and devices.

Objects can include functions. **Serial** is an object with several functions. **begin()** in **`Serial.begin()`** is a function. **print()** is another function that is part of the **Serial** object.

You will notice that some lines end with a semi-colon. This is used to identify the end of an instruction. An instruction can be an equation or function call.

When you create a function such as **`setup()`**, you do not need a semi-colon. When you call a function then a semicolon is needed at the end of the function.

The program is made up of two functions. **`setup()`** function is run at reset, power up or after code upload only once. It is used to initialize all the needed interfaces and any parameters. **`loop()`** function is run after the **`setup()`** function and is repeatedly run hence the name loop.

This program configures the serial interface to send messages at 115200 bits per second.

The message is “Hello World” and is repeatedly displayed.

```
void setup()
{
  Serial.begin(115200);
}

void loop()
{
  Serial.println(“Hello World”);
}
```

`Serial.begin()` is a function that intializes the serial interface and sets the bit rate.
`Serial.println()` sends the specified message over the serial interface and move the cursor to down one line.
`delay(500)` is a command to stop the program for 500 milliseconds.

What is in the Software

In the **setup()** function, it executes the function **Serial.begin(115200);** This function initializes the UART which is connected to the USB port to allow for communications.

In the **loop()** function, it executes the function **Serial.print(“Hello world”);** This function send the text in quotes to the UART. This is displayed in the Serial Monitor window.

The other function is called **delay()**. This function stops the program for a specified period of time. The unit is in milliseconds. The code to the write displays the text every half second.

```
void setup()
{
  Serial.begin(115200);
}

void loop()
{
  Serial.println(“Hello World”);
  delay(500);
}
```

What are Bits and Bytes

Bits are a unit of information in the computer. It has two states:

State 0 is logic level 0 also known as low and represented in the processor as 0 volts. State 1 is a logic level 1 also known as high and represented in the processor a greater than 2 volts.

Bytes are a group of 8 bits. This allows values greater than 1 to be processed. There is a weighting scheme so that numbers can be represented using a byte. The least significant bit has a weight of zero. 2 to the power of 0 is 1. The most significant bit has a weight of seven. 2 to the power of 7 is 128. Add up all the weights of the bits that are set to logic 1 to get the value of the byte.

Byte

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Example:

10111001 equals $2^7 + 2^5 + 2^4 + 2^3 + 2^0 = 185$

128 + 32 + 16 + 8 + 1 = 185

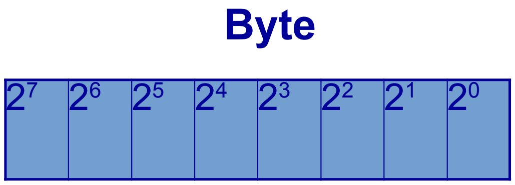
Try a couple binary numbers

01100000

10010101

What are Bits and Bytes

- Example
  - 10111001 equals $2^7 + 2^5 + 2^4 + 2^3 + 2^0 = 185$
  - 128 + 32 + 16 + 8 + 1 = 185
- Try a couple binary numbers
  - 01100000
  - 10010101



Byte

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0