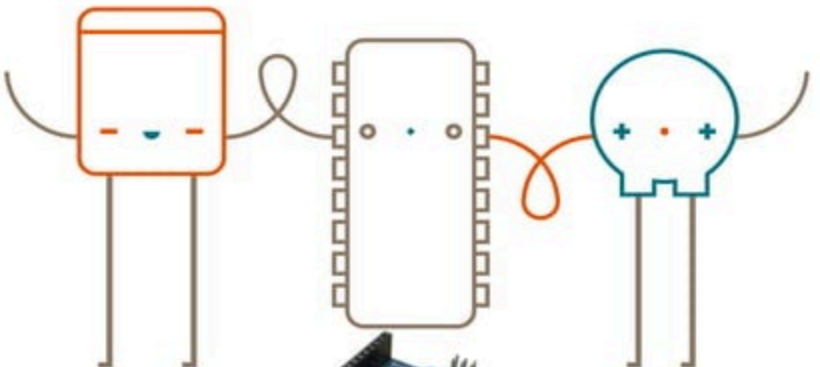
ARDUINO



INTRODUCTION TO ARDUINO

Agenda

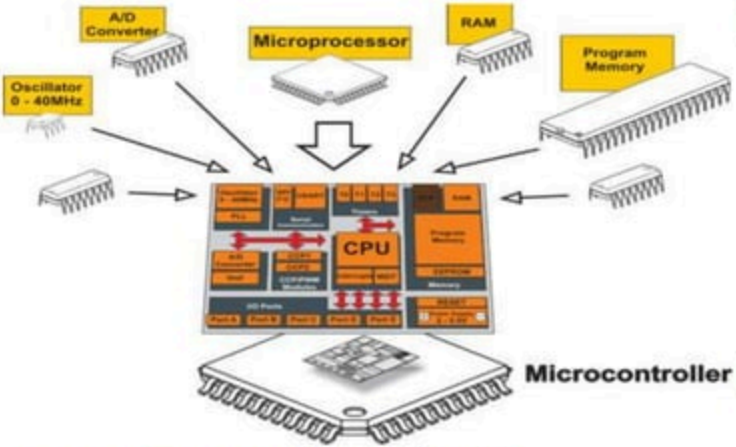
- Introduction:
  1. What is Microcontroller?
  2. What is Arduino?
  3. Types of Arduino
  4. Arduino Uno Board
  5. Arduino Shields
  6. What is Arduino used for?
  7. What can Arduino do?
  8. Why Arduino?
  9. Input/output
  10. Analog/digital
  11. Sensor
  12. Communication
- Programming structure:
  1. Data types
  2. Statement and operators.
  3. Control statements [if, if…else, switch case]
  4. Loop statement [while, for, do …while]
  5. What is function?
  6. What is Arduino Libraries
- Basic Process
- Arduino IDE
- Arduino Language Reference
- Technical Section [Handling Event Using the in-build millis() function]
- Questions???

START INNOVATION HUB

www.starhub.com.ng

INTRODUCTION

What is Microcontroller?



- A small computer on a single chip containing a Central Processor Unit (CPU), flash memory, RAM and input/output interface.
- Used for control purposes, and for data analysis

www.mikroe.com/chapters/view/1

Atmel **Intel**

Microchip

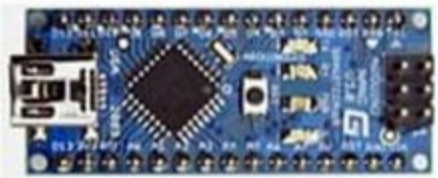
What is Arduino?

open source hardware

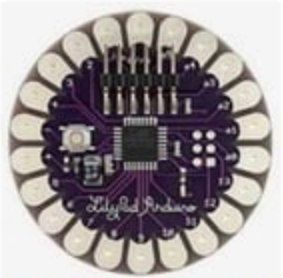
- An open-source electronics platform based on easy-to-use hardware (electronic board) and software (IDE). www.arduino.cc
- A electronic board, with on-board regulated power supply, USB port to communicate with PC, and an Atmel microcontroller chip.
- Anyone can get the details of its design and modify it or make his own.

Types of Arduino

Arduino Uno : 16MHz



Arduino Nano : 16MHz



LilyPad Arduino : 8MHz

Arduino Mega2560 : 16MHz

Arduino Ethernet : 16MHz

Arduino Pro Mini
: 8 (3.3 V)/16
(5 V) MHz

Types of Arduino

Arduino MKR1000 : 48MHz

Arduino 101 : 32MHz

Arduino Zero : 48MHz

Arduino Due : 84MHz

Arduino Yún : 16MHz and 400MHz

Arduino Leonardo : 16MHz

Types of Arduino

Arduino Fio : 8MHz

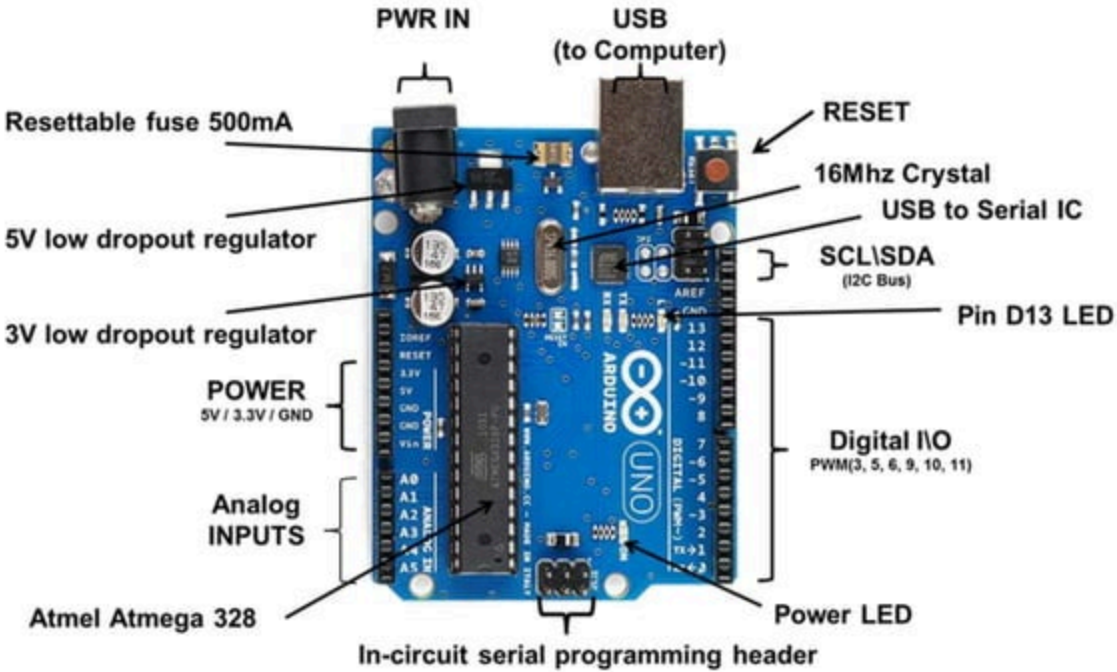
Arduino Pro : 16MHz

Arduino Mega ADK : 16MHz

Arduino Esplora : 16MHz

Arduino Micro : 16MHz

Arduino Uno Board



Arduino Shields

Boards plugged on top of the Arduino PCB to extend its capacities.

What is Arduino used for?

- Physical Computing projects / research
- Interactive Design
- Rapid prototyping

What can Arduino do?

- Sensors (sense things)
  - Push buttons, touch pads, tilt switches.
  - Variable resistors (volume knob / sliders)
  - Photo-resistors (light intensity)
  - Thermistors (temperature)
  - Ultrasound (proximity range finder)
- Actuators (do things)
  - Lights, LED's
  - Motors
  - Speakers
  - Displays (LCD)

Why Arduino?

- It is Open Source, both in terms of Hardware and Software.
- It can communicate with a computer via serial connection over USB.
- It can be powered from USB or standalone DC power.
- It can run standalone from a computer (chip is programmable) and it has memory (a small amount).
- It can work with both Digital and Analog electronic signals. Sensors and Actuators.
- You can build robots, drone, home automation, IoT application, farm management system with Arduino.

Input/output

Input is a signal / information going into the board.

Examples: Buttons Switches, Light Sensors, Flex Sensors, Humidity Sensors, Temperature Sensors, Photo-transistor, etc.

Output is any signal exiting the board.

Examples: LEDs, DC motor, Servo motor, Relay, Stepper motor, Piezo, Buzzer, RGB LED, etc.

Analog/digital

Digital signal are anything that can take specific levels of values with specific offset between each other.
Ex: Square waves

Analog signal are anything that take value between its minimum value and maximum value.
Ex: Temperature, Sine waves, etc.

Sensors

Devices that transforms the physical quantity in electrical value.

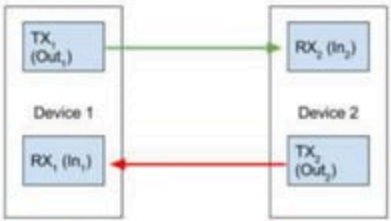
The physical quantity could be

- Light
- Heat
- Motion
- Moisture
- Pressure
- Temperature
- Blood pressure
- Humidity
- Speed
- And others environmental phenomena

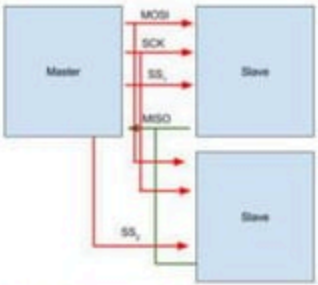
Universal Asynchronous Receiver/Transmitter (UART): is a form of serial communication because data is transmitted as sequential bits

Serial Peripheral Interface (SPI): is a master-slave model, where there is one master device and multiple slave devices.

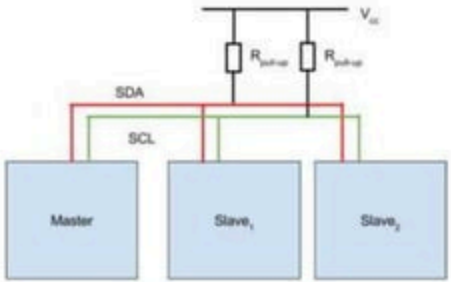
Inter-integrated circuit (I^2C): it connect multiple masters to multiple slaves.



Hardware connection for UART



SPI multiple-slaves connected to a single master



I^2C hardware connection

START INNOVATION HUB

www.starhub.com.ng

ARDUINO PROGRAMMING STRUCTURE

Data Types

Type	Byte length	Range of values
boolean	1	Limited to logic true and false
char	1	Range: –128 to +127
unsigned char	1	Range: 0 to 255
byte	1	Range: 0 to 255
int	2	Range: –32,768 to 32,767
unsigned int	2	Range: 0 to 65,535
word	2	Range: 0 to 65,535
long	4	Range: –2,147,483,648 to 2,147,483,647
unsigned long	4	Range: 0 to 4,294,967,295
float	4	Range: –3.4028235E+38 to 3.4028235E+38
double	4	Range: –3.4028235E+38 to 3.4028235E+38
string	?	A null ('\0') terminated reference type data build from a character array
String	?	An reference data type object
array	?	A sequence of a value type that is referenced by a single variable name

Example:

Int i = 140; char c = 'A'; long u = 234455; float f = 1.56;

unsigned int num = 454;

Statement and Operators

Statement represents a command, it ends with ';'
Example: **int i;**
i=49;
Operators are symbols that used to indicate a specific function:

- Math operators: [+,-,*,/,%,^]
- Logic operators: [==, !=, &&, |||]
- Comparison operators: [==, >, <, !=, <=, >=]

Syntax:

- ';' Semicolon (end statement)
- '{ }' curly braces (block of statement)
- '//' single line comment,
- /*Multi-line comments*/

Compound Operators:

- ++ (increment)
- -- (decrement)
- += (compound addition)
- -= (compound subtraction)
- *= (compound multiplication)
- /= (compound division)

If statement:

```
if(condition)
{
        statements;
}
else if(condition2)
{
        Statements;
}
Else{
        statements;
}
```

Switch statement:

```
switch (x) {
    case 1:
    //do something when x equals 1
    break;
    case 2:
    //do something when x equals 2
    break;
    default:
    // if nothing else matches, do the default
    // default is optional
}
```

Loop Statement

Do… while:

```
do
{
        Statements;
}while(condition); // the statements are run at least once.
```

While:

```
While(condition)
{
        statements;
}
```

for

```
for (int i=0; i <= var; i++){
        statements;
}
```

A body of code designed to solve a particular task.

```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

What is Arduino Library?

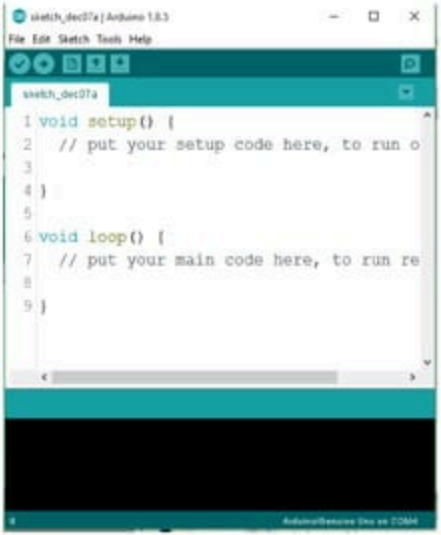
- **Arduino** environment can be extended through the use of **libraries**.
- **Libraries** provide extra functionality for use in sketches.
  - Working with hardware
  - Manipulating data.
- A number of **libraries** come installed with the IDE.
  - Download
  - Create your own.

Basic Process

- Design the circuit:
  - What are electrical requirements of the sensors or actuators?
  - Identify inputs (analog inputs)
  - Identify digital outputs
- Write the code
  - Build incrementally
    - Get the simplest piece to work first
    - Add complexity and test at each stage
    - Save and Backup frequently
  - Use variables, not constants
  - Comment explicitly

Arduino IDE

Download the Arduino IDE
(The program used to write code and uploading it to arduino boards) from:
http://arduino.cc/en/Main/Software



Arduino Language Reference

Reference Language | Libraries | Comparison | Changes

Language Reference

Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

Structure

- setup()
- loop()

Control Structures

- «

Variables

Constants

- HIGH | LOW
- INPUT | OUTPUT | INPUT_PULLUP

Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

http://arduino.cc/en/Reference/HomePage

Handling Multiple Event Using the Arduino in-build millis() function:

Make a 4 LEDs to blink at different rate of 1 sec, 0.5 sec, 0.25 sec and 0.1sec in an infinite loop.

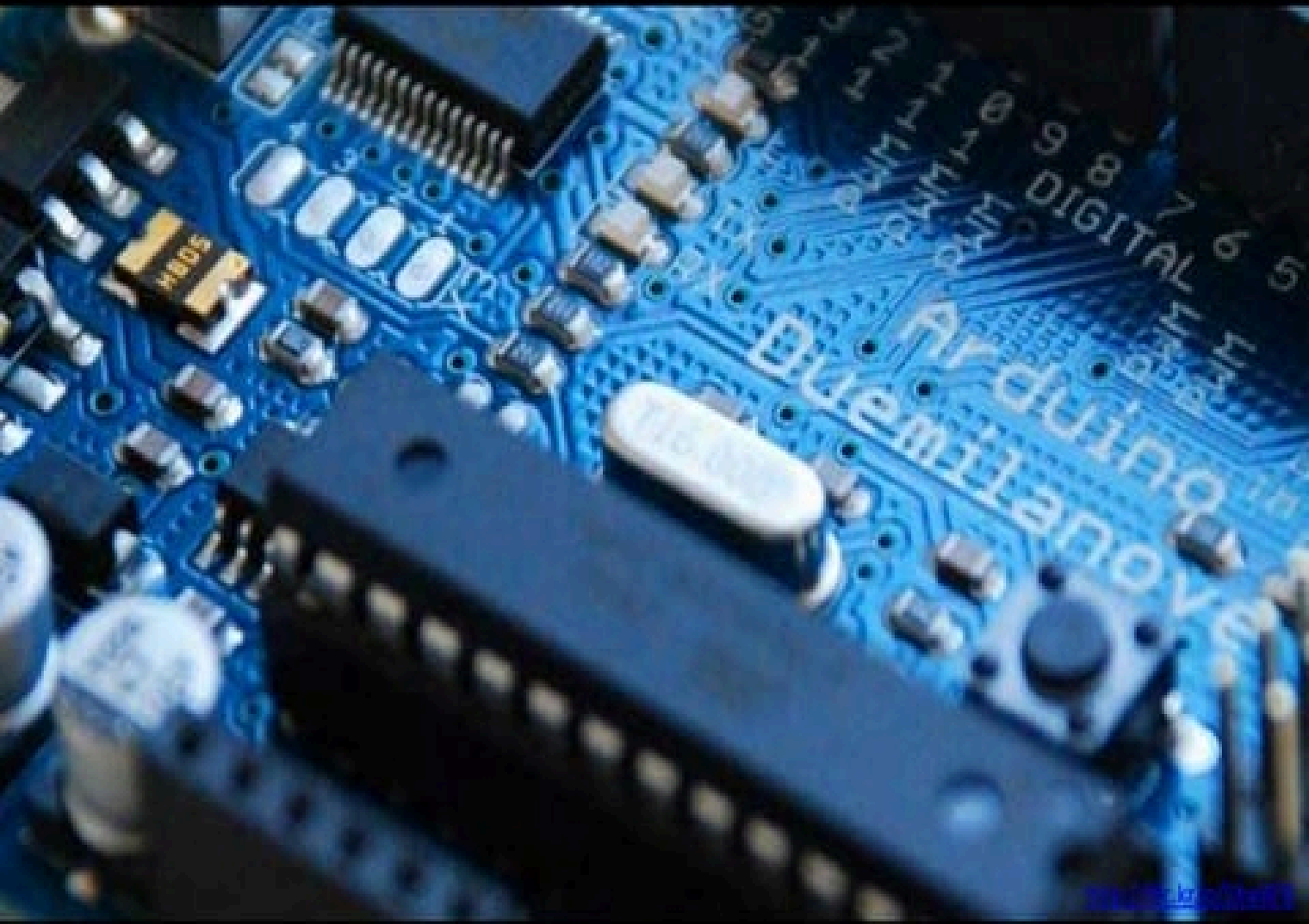
Download the demo code from

https://github.com/emotexplanet/Arduino_Multiple_Event_with_millis_function

Questions???

Thank You

Introduction to Arduino



Omer Kilic | omer@tinkersoc.org

Arduino Programming

(C) 2014 James Lewis
james@baldengineer.com

Arduino Pin Description

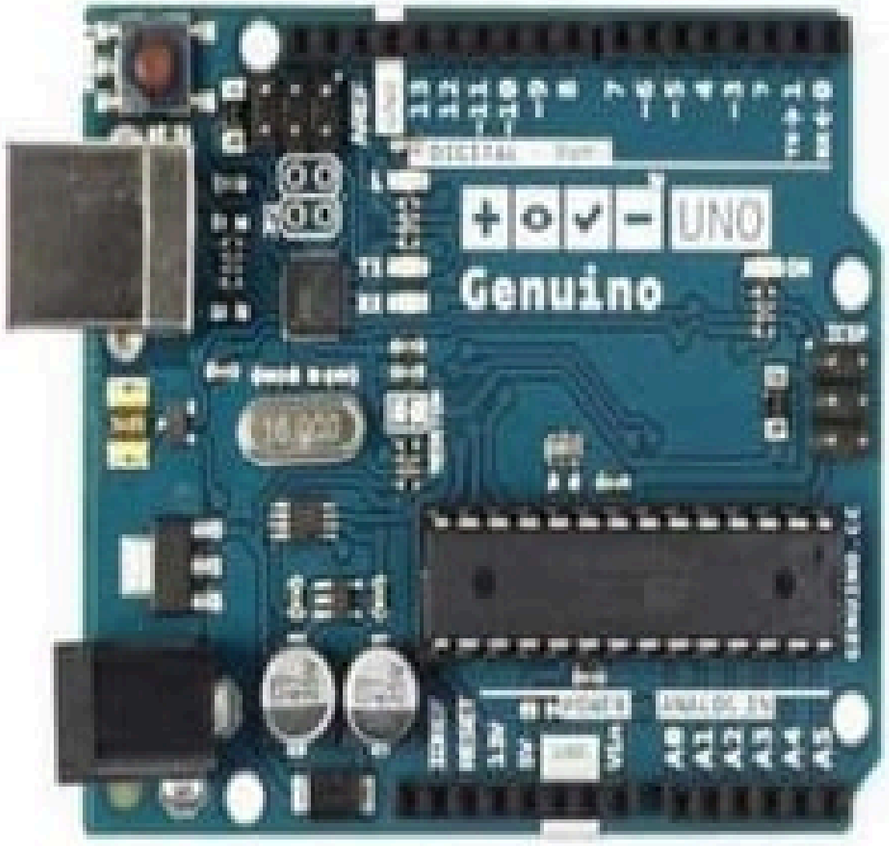
BY: NIKET CHANDRAWANSHI

HTTP://WWW.NIKETCHANDRAWANSHI.ME/


ARDUINO-UNO
- An Open Source Hardware
Submitted by,
Jerin K John
[jerinisready@gmail.com]
S7 CS - 12141009
Do it yourself !!!


ARDUINO
WORKSHOP

Introduction to Arduino/Genuino Uno



By: Yeo Kheng Meng (yeokm1@gmail.com)

Startathon (20 Feb 2016)

https://github.com/SustainableLivingLab/intro-to-arduino

FABTECH TECHNICAL CAMPUS
COLLAGE OF ENGINEERING,
SANGOLA

PRESENTATION ON

SUMMER TRAINING ON

ARDUINO

PRESENTED BY :

PHADTARE RAVI DATTATRYA

ROLL NO.- 21

Line follower competition.

Introduction to...

Arduino

IEEE MÂNA UNIVERSITY STUDENT BRANCH