



ARDUINO PROGRAMMING & ITS APPLICATIONS

Day-One

Zuhaib A. Shaikh,
Asst. Prof., CSE Deptt., QUEST
Web: zuhaib-shaikh.neocities.org



Outline

Day	Activity
1	<ul style="list-style-type: none">i. Introduction to Arduino development board and Arduino IDEii. C/C++ language overviewiii. Basic input / output with Arduinoiv. Overview of Proteus simulation software for Arduino simulation
2	<ul style="list-style-type: none">i. Interfacing and glowing LEDs with different patternii. Interfacing push button and piezo buzzeriii. Interfacing a temperature sensor with Arduino
3	<ul style="list-style-type: none">i. Familiarization with Serial Monitor for input and for outputii. Interfacing LDR sensor with Arduinoiii. Interfacing PIR motion sensor with Arduino
4	<ul style="list-style-type: none">i. Interfacing Arduino with LCD (16x2), relay and Servo motorii. Interface Arduino with Sonic Sensor for obstacle detection
5	<ul style="list-style-type: none">i. Interfacing shift register and 7-segment display with Arduinoii. Interfacing HC-05 Bluetooth module with Arduinoiii. Driving GSM modem with Arduino
<p>Prerequisites:</p> <ul style="list-style-type: none">- Knowledge of C++- Knowledge of basic electronic components	



Introduction

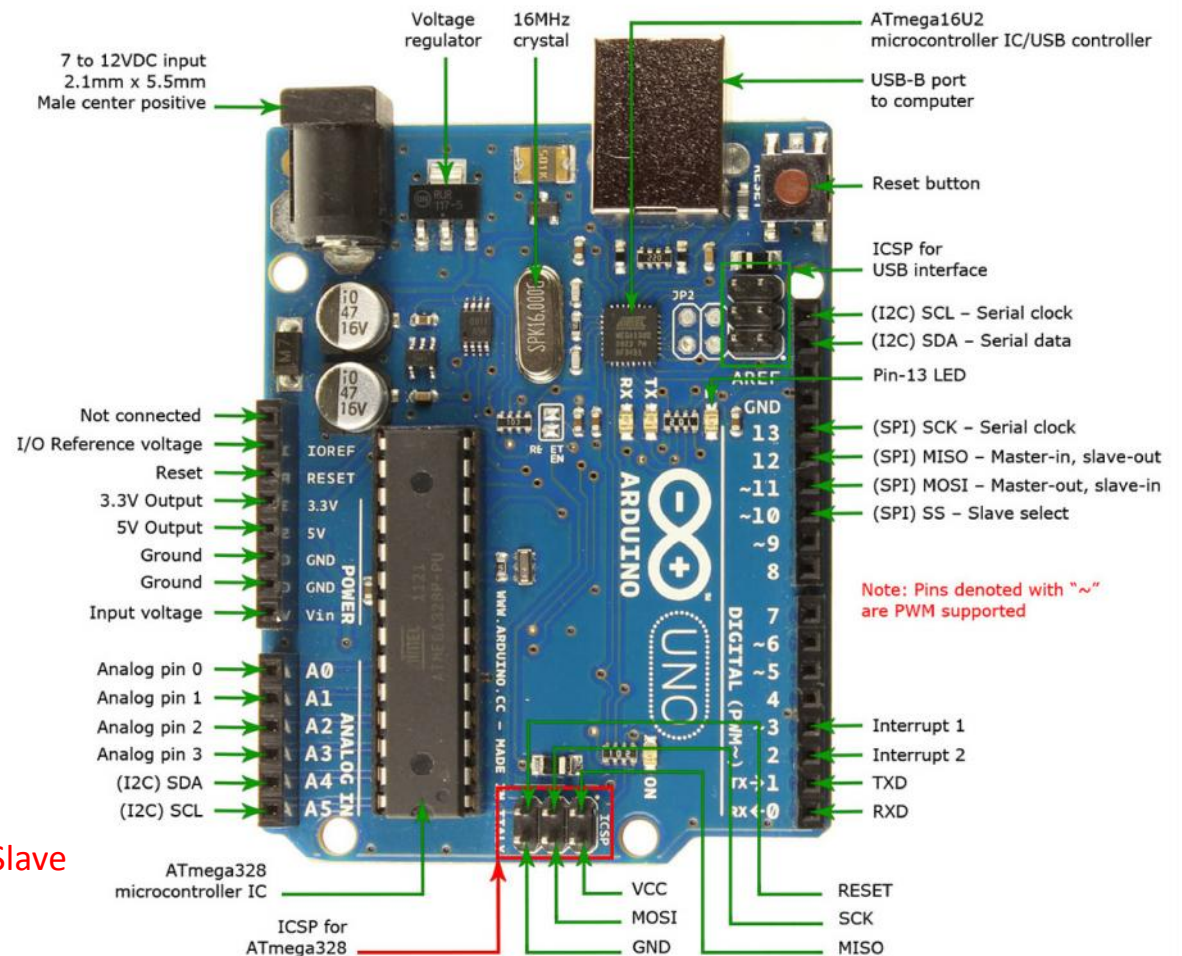
- Arduino (a Italian company) is opensource HW/SW company
 - Designs development board for microcontrollers and microcontroller kits
 - Its IDE supports C/C++ like language for writing and burning the code (HEX file) into Arduino Board
- Useful for learning, experiments and prototyping
- Different Models with different microcontrollers:
 - Arduino Uno
 - Arduino Mega
 - Arduino Nano
 - Arduino Due, etc,
- Selection of particular model depends on requirements
- Many shield and sensors are compatible with Arduino

Arduino Uno

- Basic board, suitable for most of the projects and mostly supported by Arduino shields
- It has:
 - ATmega Microcontroller
 - Digital I/O pins (0 to 13)
 - Pins with tilde (~) can generated PWM analog signals
 - Power pins
 - Analog pins (A0 to A5)
 - Control Pins
 - Power connector
 - USB Port (for connecting with PC)
 - LEDs (Power, TX, RX & L-for pin 13)

*I2C – Inter-integrated Component for Master-Slave Communication

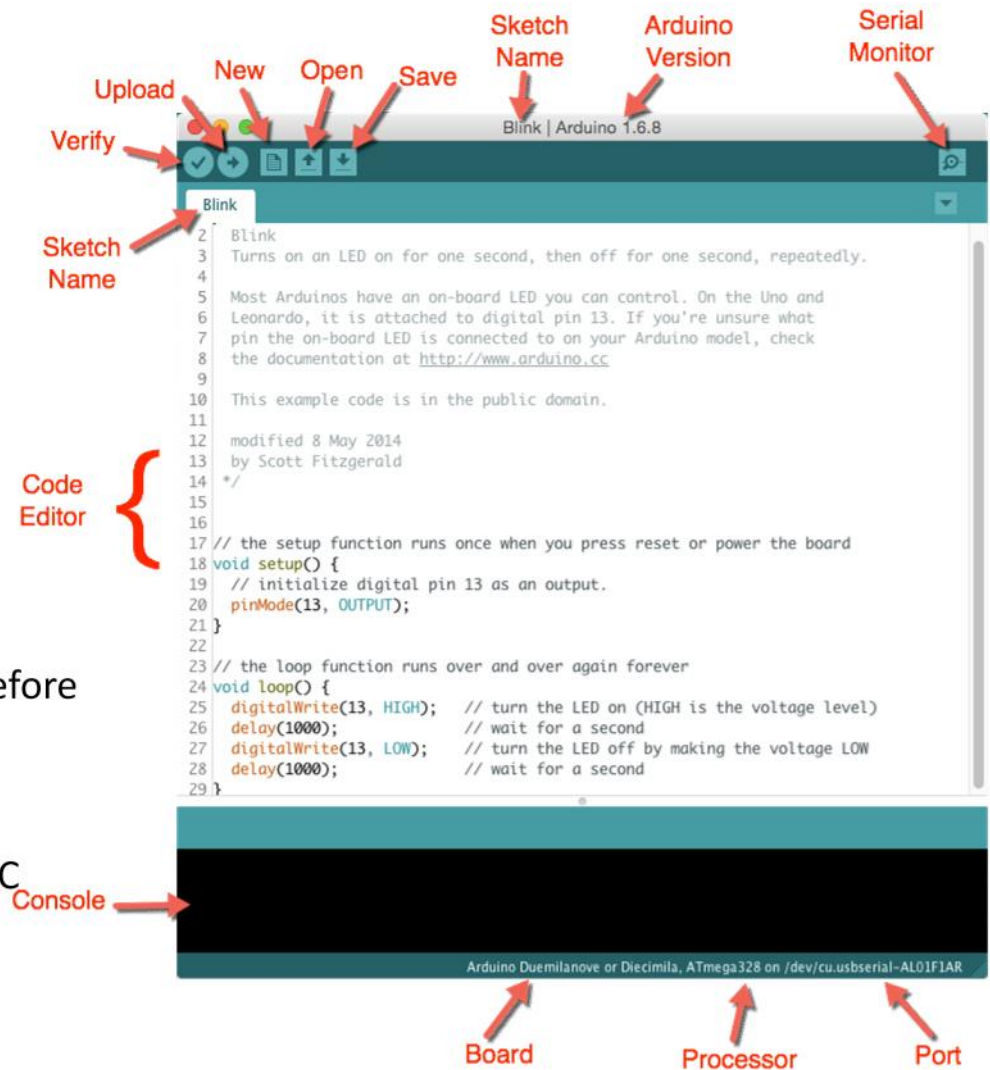
*ICSP – In-Circuit Serial Programming





Arduino IDE

- Allows to write C++ like programs (sketches), compile those and burn those on Arduino board
- IDE is divided into
 - Command Area
 - Title Bar
 - Menu Items
 - Icons
 - Text Area
 - Message Window
- Port and Board
 - IDE supports many Arduino boards
 - Connected as serial COM port
 - Port and Board should be selected before working on project
- Serial Monitor
 - IDE allows to input/output from/to PC & Arduino using IDE





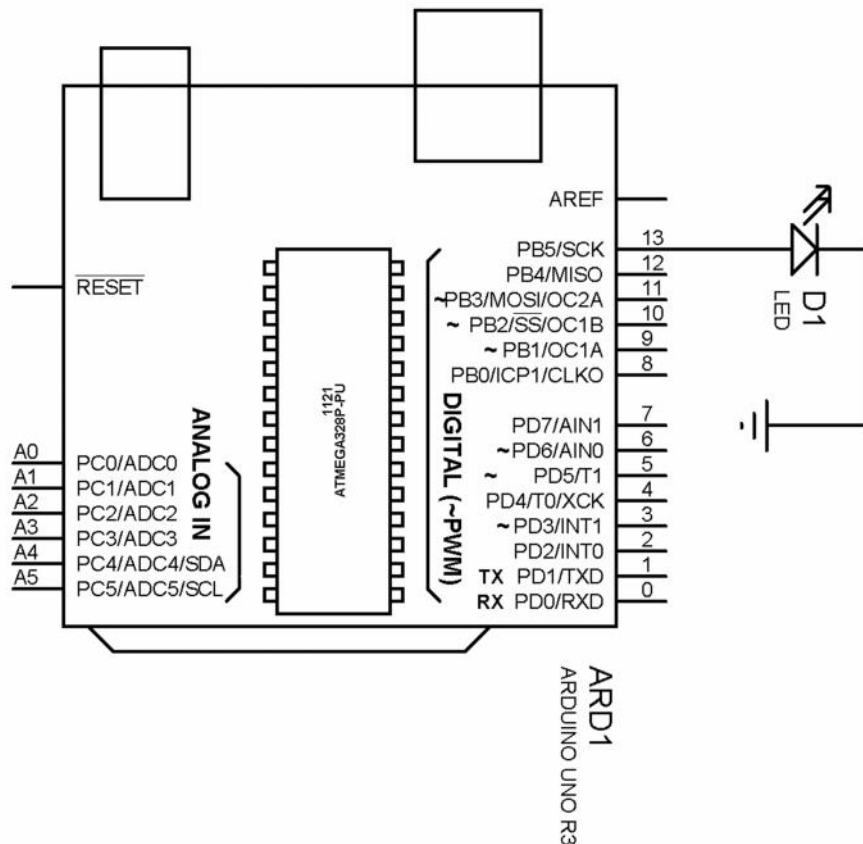
Arduino C++

- Arduino C++ is based on traditional C++ except standard library
- Similar keywords, control structures, primary and secondary datatypes etc.
- Arduino library
 - Setup function
 - The function which is to be executed first for only once
 - This function is usually used to set certain things like Digital I/O pins, Analog pins, baud rate etc.
 - Loop function
 - The function which is to be executed repeatedly
 - All instructions are written inside loop function which are meant to be executed infinite times
 - Basic I/O functions
 - **pinMode**(*pin_no, mode*) – sets the digital pin in **INPUT** or **OUTPUT** mode (mostly in setup function)
 - **digitalWrite**(*pin_no/reference, value*) – assigns **HIGH** or **LOW** value to particular pin number or its reference until interrupted
 - **Variable = digitalRead**(*pin_no/reference*) – Reads digital value from particular pin number or its reference to defined variable
 - **Serial.begin**(*baud_rate*) – To set serial baud rate for serial monitor
 - **Serial.print**(*string_and/or_variable*) and **serial.println**(*string_and/or_variable*) – prints string (enclosed in double quotation) and/or variable value on serial monitor without and with new line respectively
 - **delay**(*milli_seconds*) – describe microcontroller to do nothing (wait state) in specific time defined in milli-seconds



Example 1: LED Blink

Circuit



Code

```
/*
Arduino Blink LED Sketch
*/

void setup()
{
  Serial.begin(9600);           //Setting baud rate
  Serial.println("LED Blink"); //Displaying a message
  pinMode(13, OUTPUT);         // set digital pin 13 to output
}

void loop()
{
  digitalWrite(13, HIGH);      // turn on digital pin 13
  delay(1000);                 // pause for one second
  digitalWrite(13, LOW);       // turn off digital pin 13
  delay(1000);                 // pause for one second
}
```

* LED L is connected to pin number 13 for testing purpose by default



Verifying and uploading sketch

- Once code is written, it is necessary to compile (verify) the code to generate HEX file
- If code is without errors, its hex file will be temporary stored on PC by displaying following information:
 - Size of the program (in bytes)
 - Size of total Arduino memory (in bytes)
- HEX (output) file can be uploaded on Arduino using Upload button
 - TX & RX LEDs will blink during upload
 - Arduino board will start executing HEX file once upload is done
- If the sketch has syntax errors
 - Display error message by mentioning the exact or expected error
 - Compilation and uploading can't be continued until errors are removed

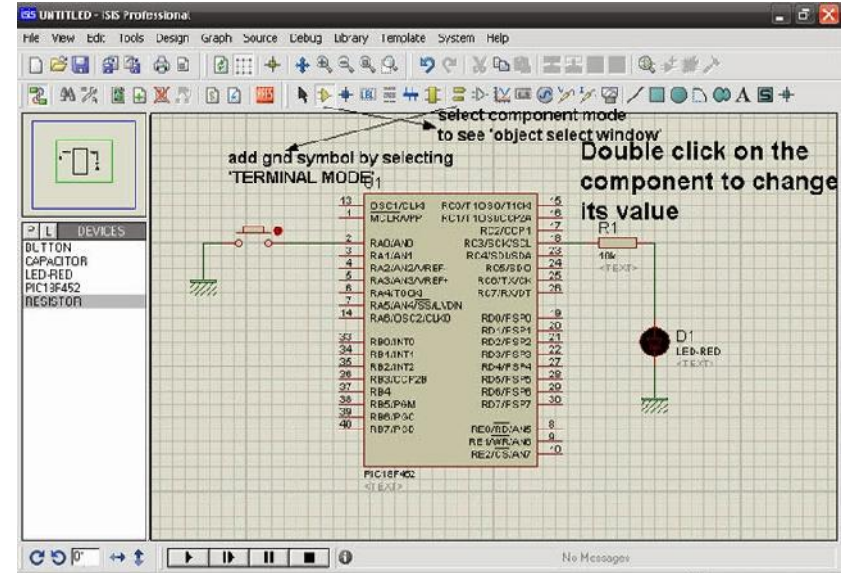
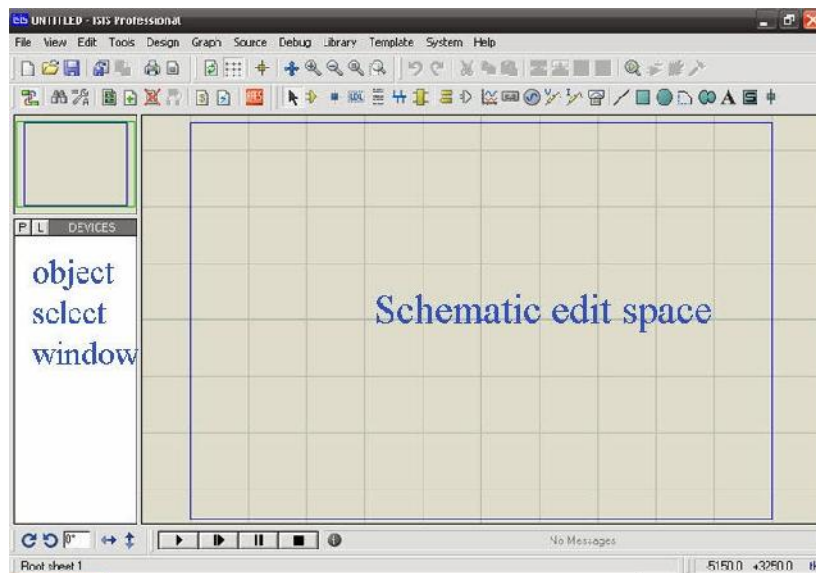
```
Done compiling.  
  
Binary sketch size: 1,076 bytes (of a 32,256 byte maximum)  
17 Arduino Uno on COM14
```

```
expected ';' before '}' token  
  
blink.cpp: In function 'void loop()':  
blink:16: error: expected ';' before '}' token  
17 Arduino Uno on COM14
```




Simulating on Proteus

- The example can also be simulated on Proteus simulation software rather than physical implantation
- ISIS(Intelligent Schematic Input System), a tool of Proteus allows users to design and simulate electronic circuits easily

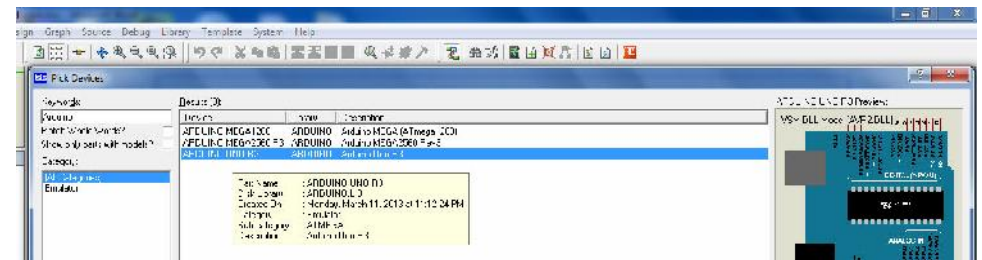
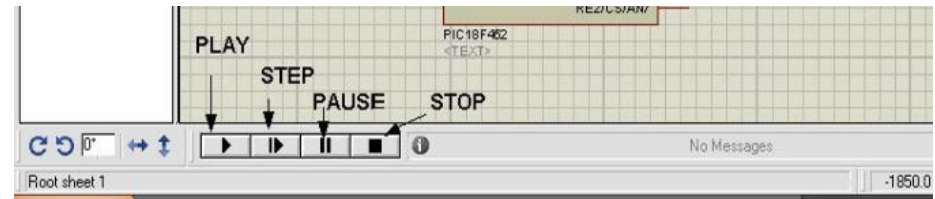
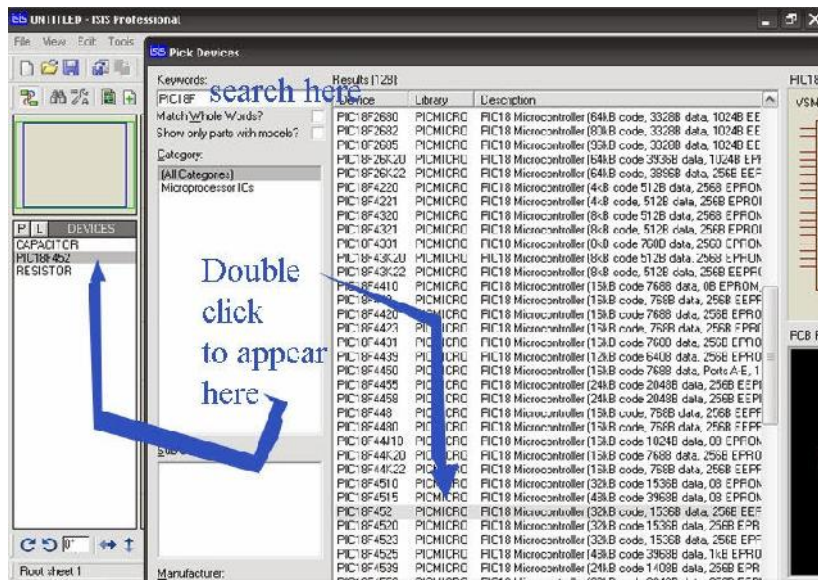


- ISIS supports wide variety of components
- Selected components are visible in object select window, while complete circuit is in schematic edit space
- Components (including power source) can be selected via menu buttons



Simulating on Proteus

- The components from Pick device window can be search via their names or model numbers

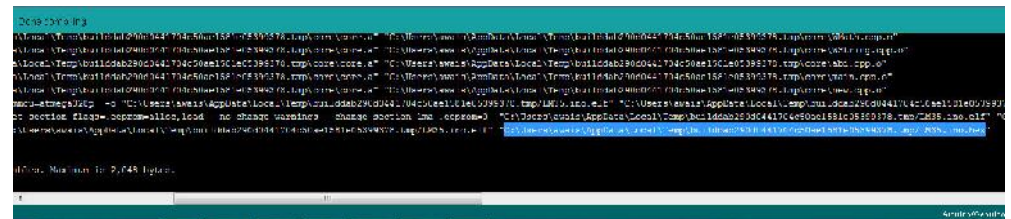
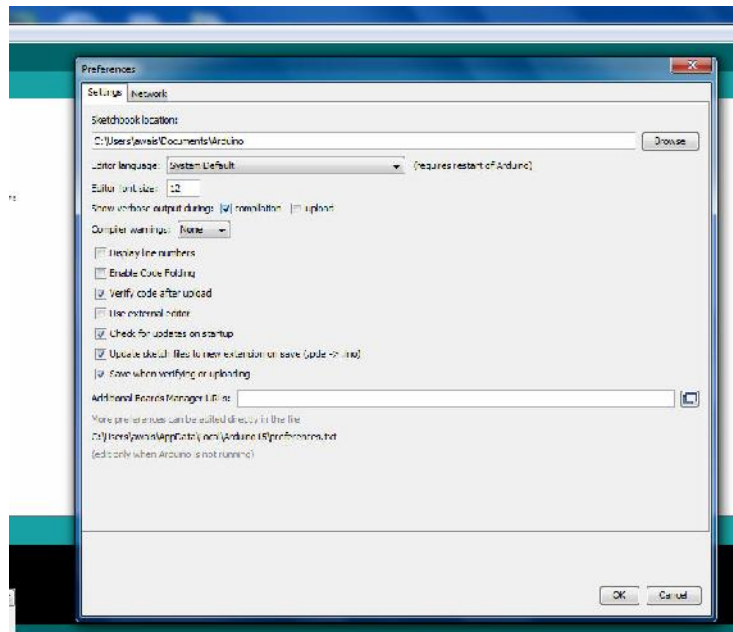


- The simulation can be started, paused and stopped using the buttons at bottom of ISIS
- ISIS-Proteus 7 doesn't not provide Arduino Library by default, therefore:
 - Download Arduino Library and extract it
<https://www.theengineeringprojects.com/ArduinoProjects/Arduino%20Library%20for%20Proteus.rar>
 - Copy the extracted files into (ISIS is needed to be closed):
C:\Program Files (x86)\Labcenter Electronics\Proteus 7 Professional\LIBRARY
- After including the Arduino library, Arduino board can be added pick device window by writing name of the board.



Simulating on Proteus

- Arduino module now needs the HEX file for simulation.
- HEX file can be obtained via Arduino IDE
 - Open preference window from File tab in IDE
 - Under setting tab, apply locate "**Show verbose output during...**" and check the **compilation** box



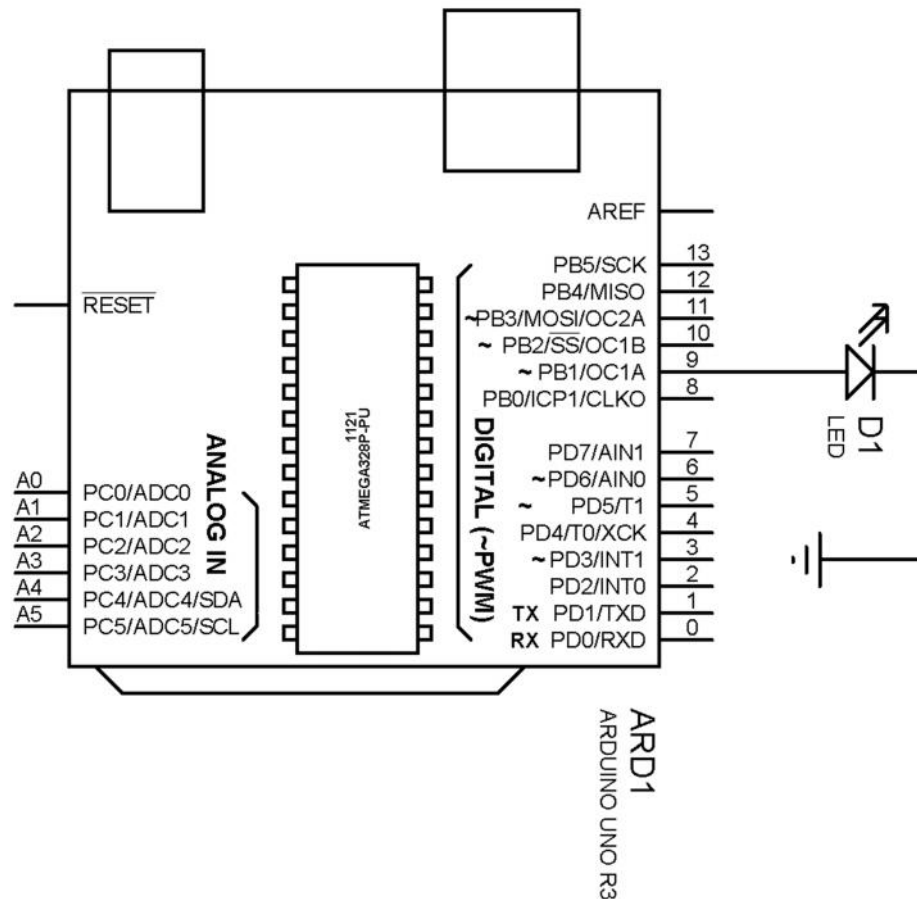
Name	Date modified	Type	Size
core	09-Mar-16 5:40 PM	File folder	
libraries	09-Mar-16 5:40 PM	File folder	
preproc	09-Mar-16 5:40 PM	File folder	
sketch	09-Mar-16 5:40 PM	File folder	
build.options.json	09-Mar-16 5:40 PM	JSON File	1
LM35.ino.eep	09-Mar-16 5:40 PM	EEP File	1
LM35.ino.elf	09-Mar-16 5:40 PM	ELF File	71
LM35.ino.hex	09-Mar-16 5:40 PM	HEX File	12
LM35.ino.with_bootloader.hex	09-Mar-16 5:40 PM	HEX File	13

- After verifying the sketch, HEX file can be located from the path shown in message window
- File path can be viewed by pasting the path in Run dialog box
- HEX file is then to be included in Proteus by double clicking on Arduino & selecting HEX file
- Simulation can be started after completing the circuit and adding HEX File



Example 2: LED Fade

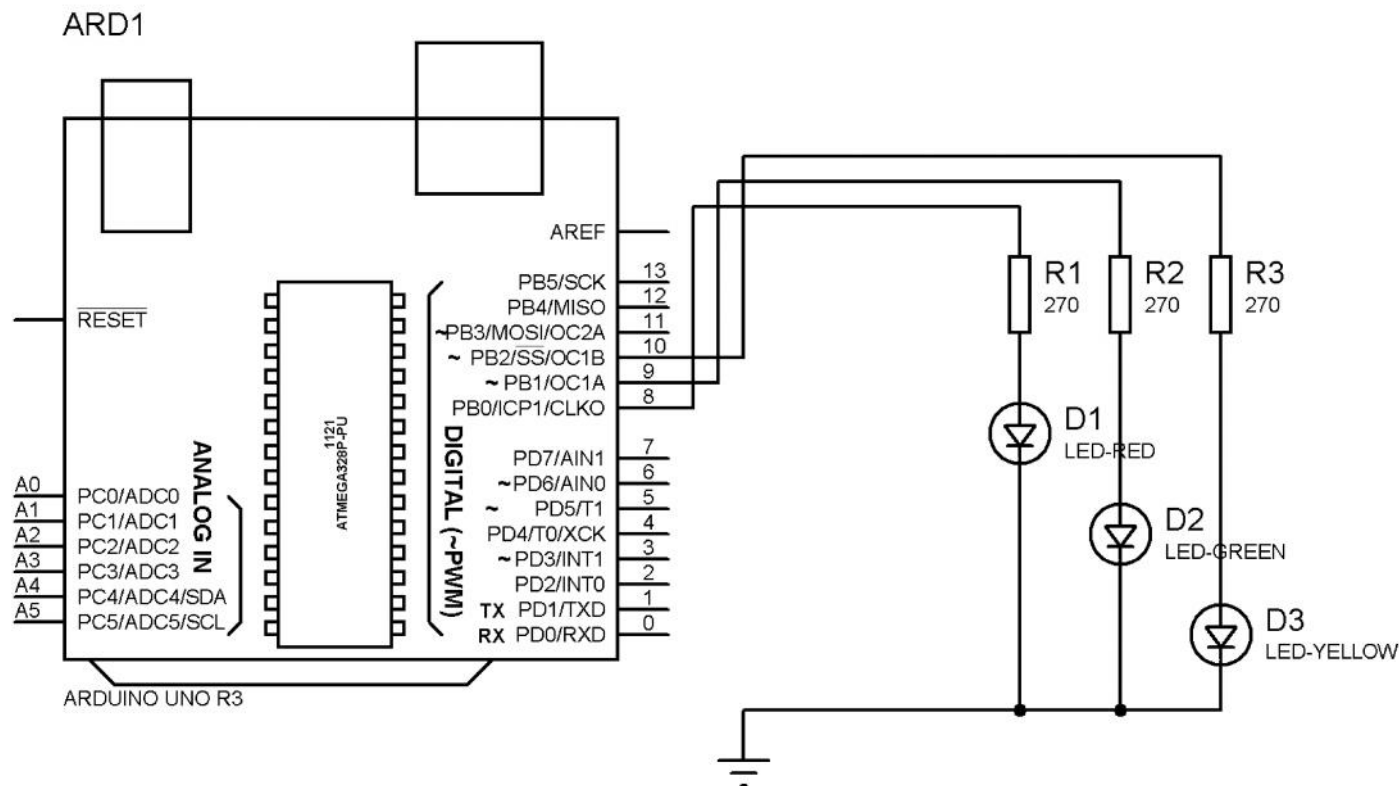
- Connect LED with PWM pin (e.g. pin 9) and with GND
- The example code can be opened from *File>Examples>01.Basics>Fade*





Example 3: Traffic Signal (Exercise)

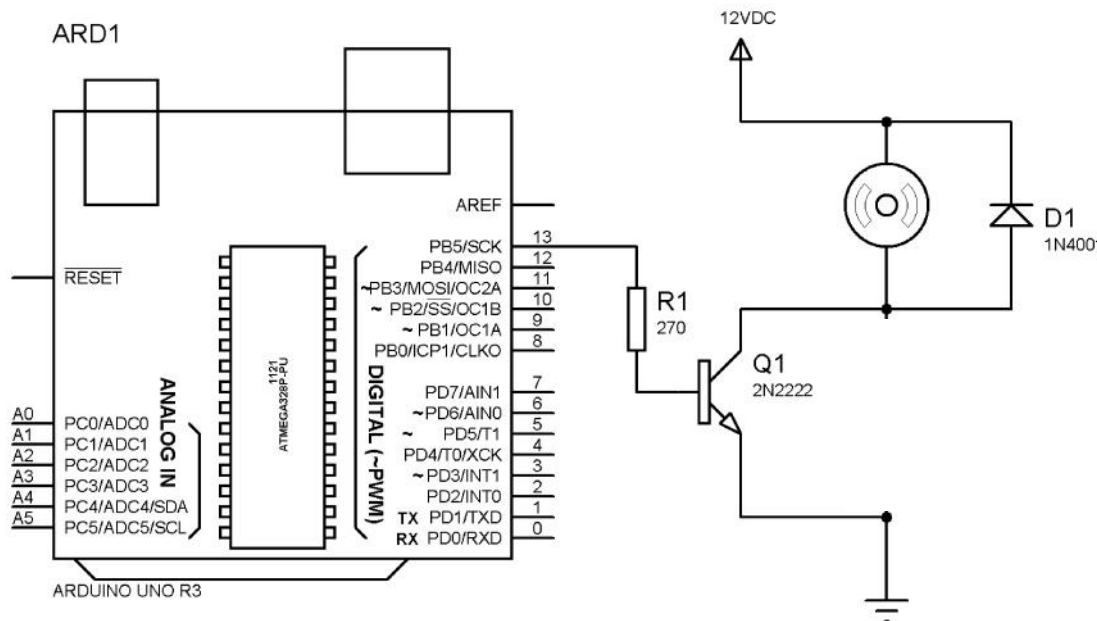
- Connect 3 LEDs with digital pins (e.g. pin 9) using 270ohm resistors and with GND
- Write program to glow those LEDs with different time periods, i.e. RED = 3s, YELLOW = 0.5s and GREEN = 4s
- The glowing order should be like Red – Yellow – Green – Yellow - Red





Example 4: Controlling DC motor

- DC motor has 2 terminals +ve & -ve. Motor start working when those terminals are connect to DC supply.
- Components:
 - 12V DC power supply (5V in case you have a 5V DC motor) X 1
 - 1N4001 or equivalent back EMF blocking diode X 1
 - 2N2222 NPN transistor X 1
 - 5V or 12V DC motor X 1
 - 270 ohm resistor X 1

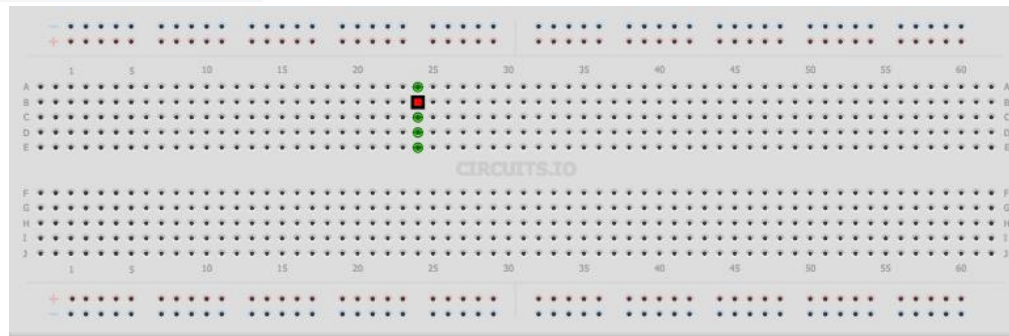
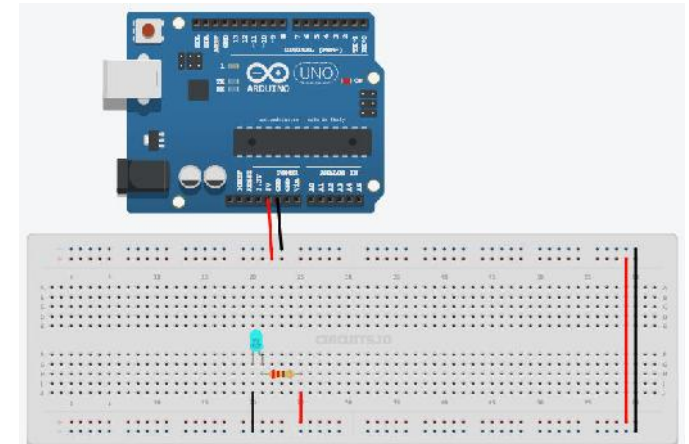
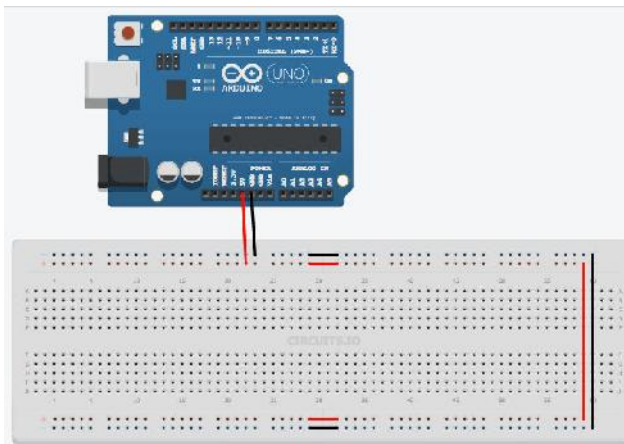


```
int motorPin = 13;
void setup()
{
  pinMode(motorPin, OUTPUT);
}
void loop()
{
  digitalWrite(motorPin,HIGH);
  delay(5000);
  digitalWrite(motorPin,LOW);
  while(1);
}
```



Using Breadboard

- Allows to design circuit connections without permanently soldering those
- Divided into matrix of rows and columns
- Columns of single block are connected with each other
- While, 2-piece or 4-piece *rails* are use to be connected with power supply to provide to components





Exercise

- Simulate a ISIS-Proteus Project using Arduino with eight LEDs.
- Blink them in an endless loop in the sequence:
1-2-3-4-5-6-7-8-7-6-5-4-3-2-1-2-3-...
- The delay b/w steps should be following by three combinations
 - one second
 - Half second
 - Quarter second



Questions

