

ARDUINO SESSION 1

Topics:

Microcontrollers

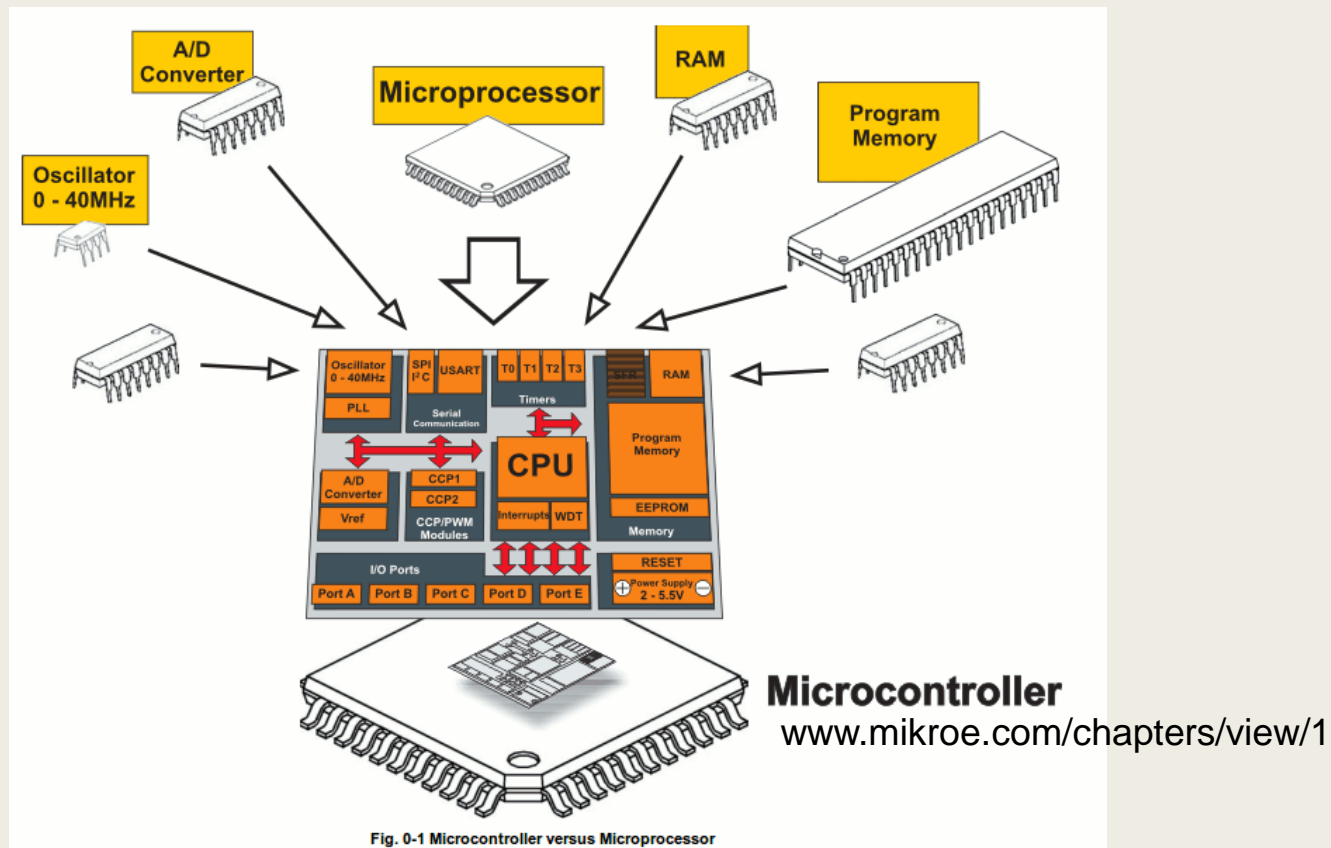
Programming Basics: structure and variables

Digital Output

Analog to Digital Conversion

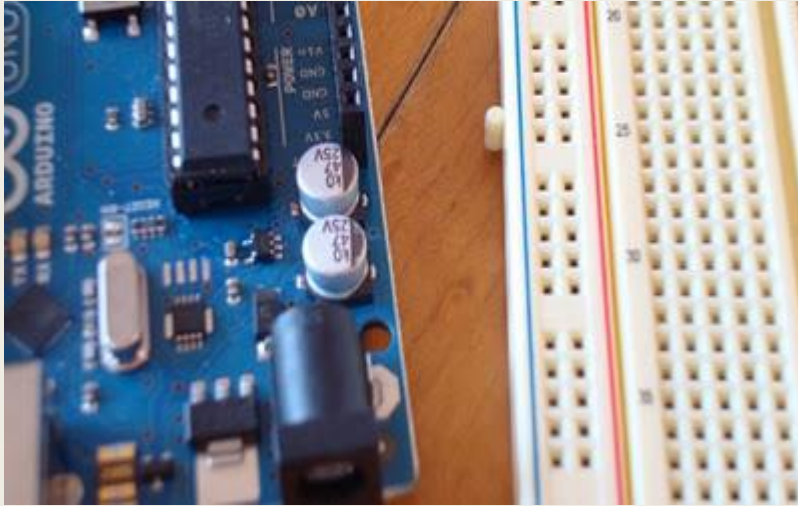
*Dr. Swarna Priya RM,
Associate Professor, SITE,*

What is a Microcontroller



- A small computer on a single chip
 - containing a processor, memory, and input/output
- Typically "**embedded**" inside some device that they control
- A microcontroller is often small and low cost

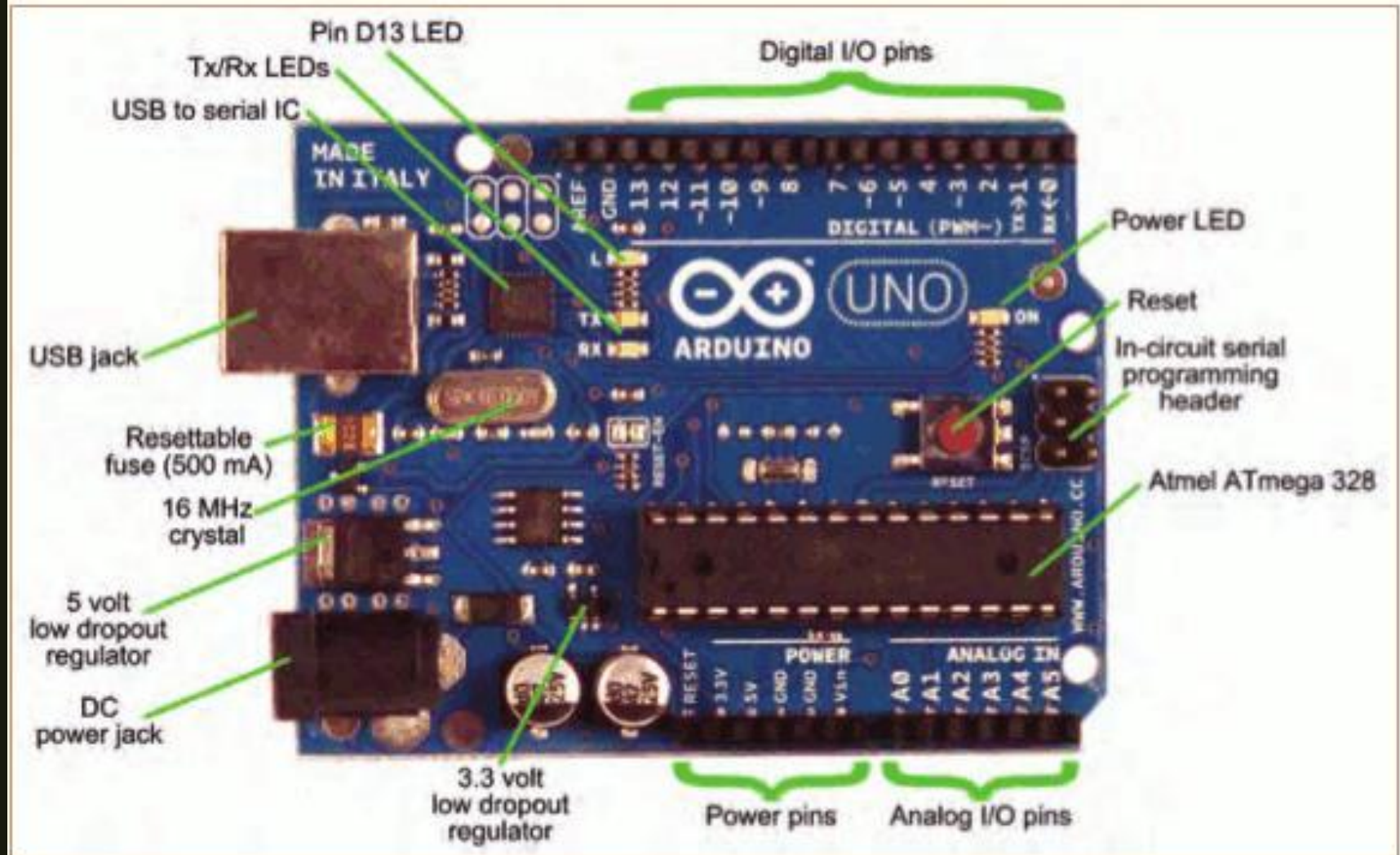
What is a Development Board



- A printed circuit board designed to facilitate work with a particular microcontroller.

- Typical components include:
 - power circuit
 - programming interface
 - basic input; usually buttons and LEDs
 - I/O pins

The Arduino Development Board



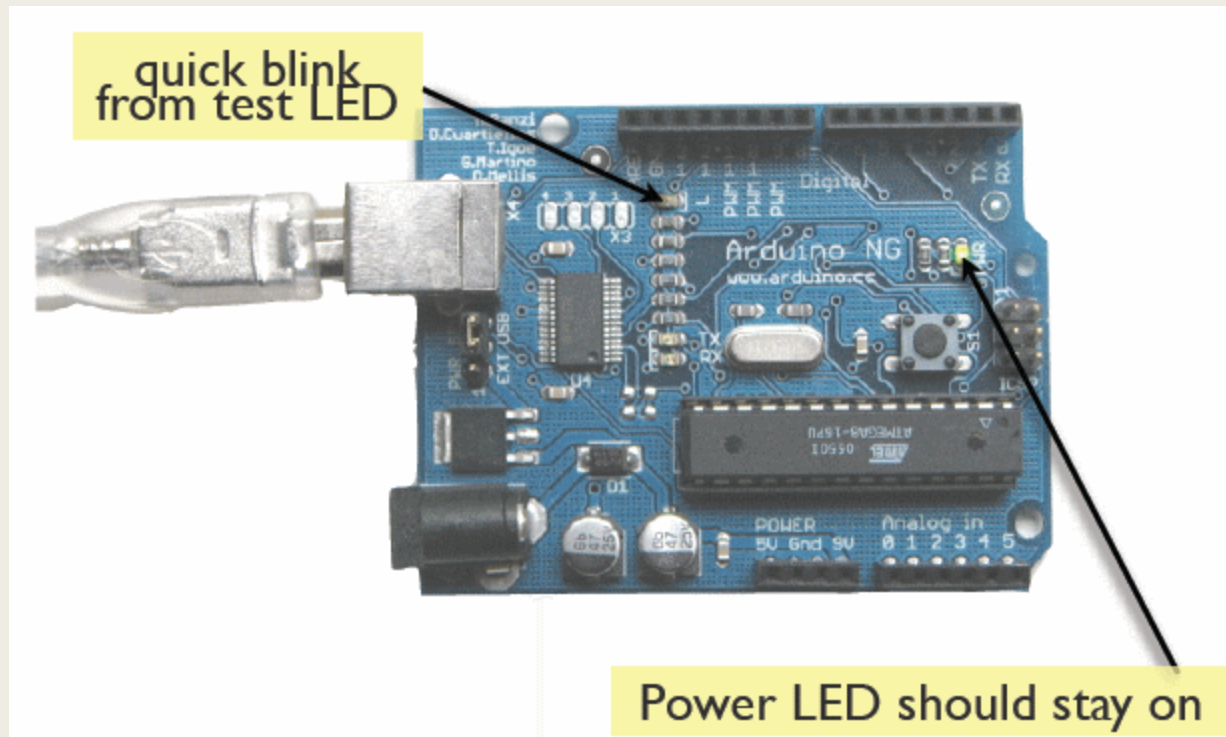
Making-robots-with-arduino.pdf

Getting Started

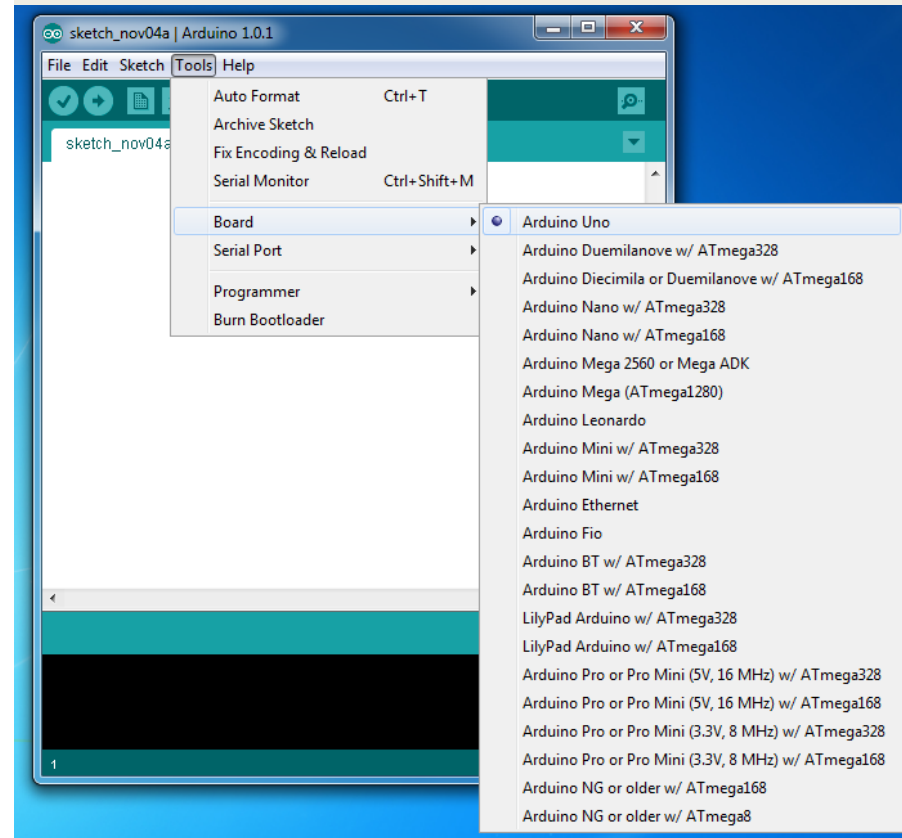
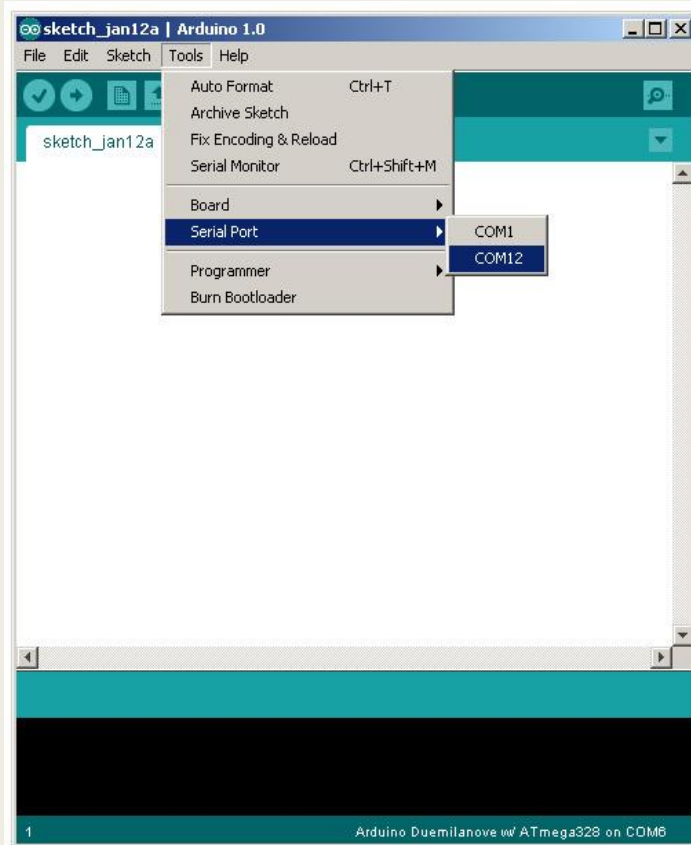
Check out: <http://arduino.cc/en/Guide/HomePage>

- 1. Download & install the Arduino environment (IDE)***
- 2. Connect the board to your computer via the USB cable***
- 3. If needed, install the drivers Launch the Arduino IDE***
- 4. Select your board***
- 5. Select your serial port***
- 6. Open the blink example***
- 7. Upload the program***

Try It: Connect the USB Cable



Select Serial Port and Board



Status Messages

Uploading worked

Done uploading.

Binary sketch size: 1110 bytes (of a 14336 byte maximum)

Size depends on
complexity of your sketch

Wrong serial port selected

```
Serial port '/dev/tty.usbserial-A4001qa8' not found. Did you select the
java.awt.EventQueue$DispatchThread.run(EventDispatchThread.java:118)
)
at
java.awt.EventQueue$DispatchThread.run(EventDispatchThread.java:118)
```

Wrong board selected

```
Wrong microcontroller found. Did you select the right board from the T
Binary sketch size: 000 bytes (of a 14336 byte maximum)
avrdude: Expected signature for ATMEGA8 is 1E 93 07
Double check chip, or use -F to override this check.
```

nerdy cryptic error messages

Using Arduino

- Write your sketch
- Press Compile button (to check for errors)
- Press Upload button to program Arduino board with your sketch

Try it out with the “Blink” sketch!

Load “File/Sketchbook/Examples/Digital/Blink”

```
void setup() {  
  pinMode(ledPin, OUTPUT); // sets t  
}  
void loop() {  
  digitalWrite(ledPin, HIGH); // sets t  
  delay(1000); // waits  
  digitalWrite(ledPin, LOW); // sets t  
  delay(1000); // waits  
}
```



compile

Done compiling.



upload



TX/RX flash



sketch runs

Terminology

“*sketch*” – a program you write to run on an Arduino board

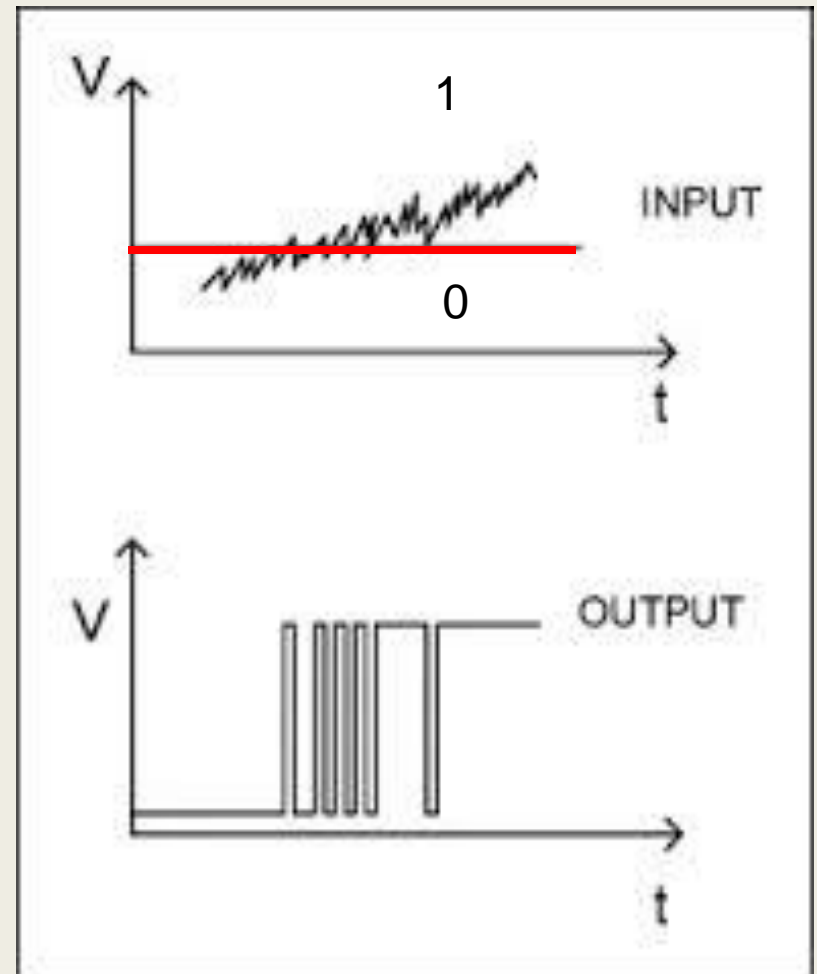
“*pin*” – an input or output connected to something.
e.g. output to an LED, input from a knob.

“*digital*” – value is either HIGH or LOW.
(aka on/off, one/zero) e.g. switch state

“*analog*” – value ranges, usually from 0-255.
e.g. LED brightness, motor speed, etc.

Digital Input / Output

Digital IO is binary valued—it's either *on* or *off*, 1 or 0



Getting started with Programming

A Little Bit About Programming



- Code is case sensitive
- Statements are commands and must end with a semi-colon
- Comments follow a `//` or begin with `/*` and end with `*/`

Bare minimum code

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

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

Bare minimum code

- **setup** : It is called only when the Arduino is powered on or reset. It is used to initialize variables and pin modes
- **loop** : The loop functions runs continuously till the device is powered off. The main logic of the code goes here. Similar to while (1) for micro-controller programming.

PinMode

- A pin on arduino can be set as input or output by using pinMode function.
- `pinMode(13, OUTPUT);` // sets pin 13 as output pin
- `pinMode(13, INPUT);` // sets pin 13 as input pin

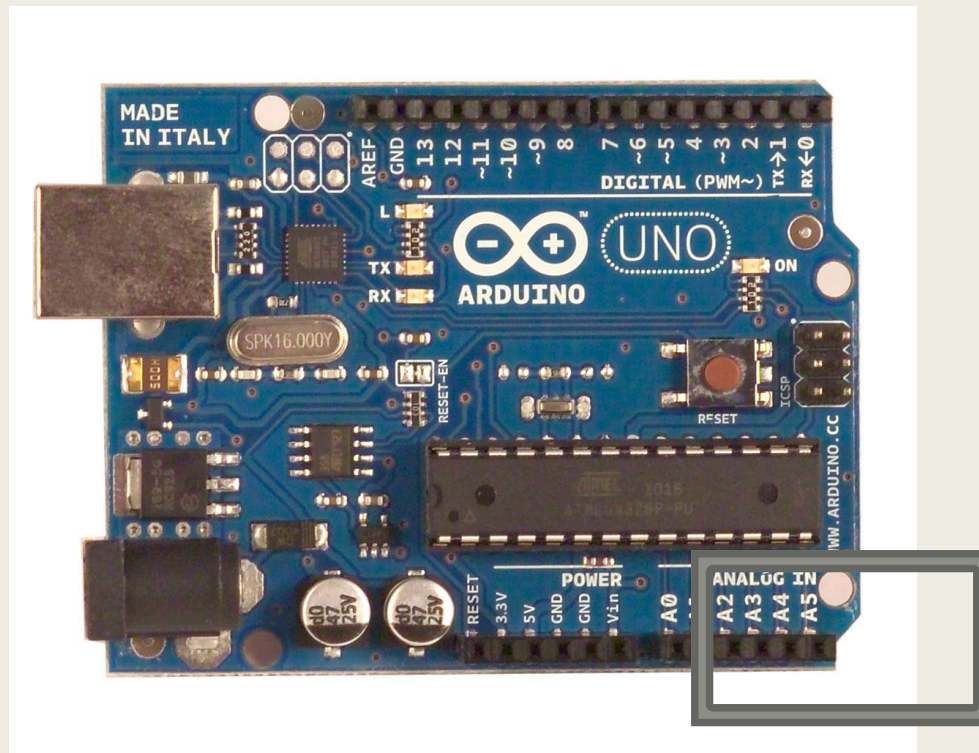
Reading/writing digital values

- `digitalWrite(13, LOW);` // Makes the output voltage on pin 13 , 0V
- `digitalWrite(13, HIGH);` // Makes the output voltage on pin 13 , 5V

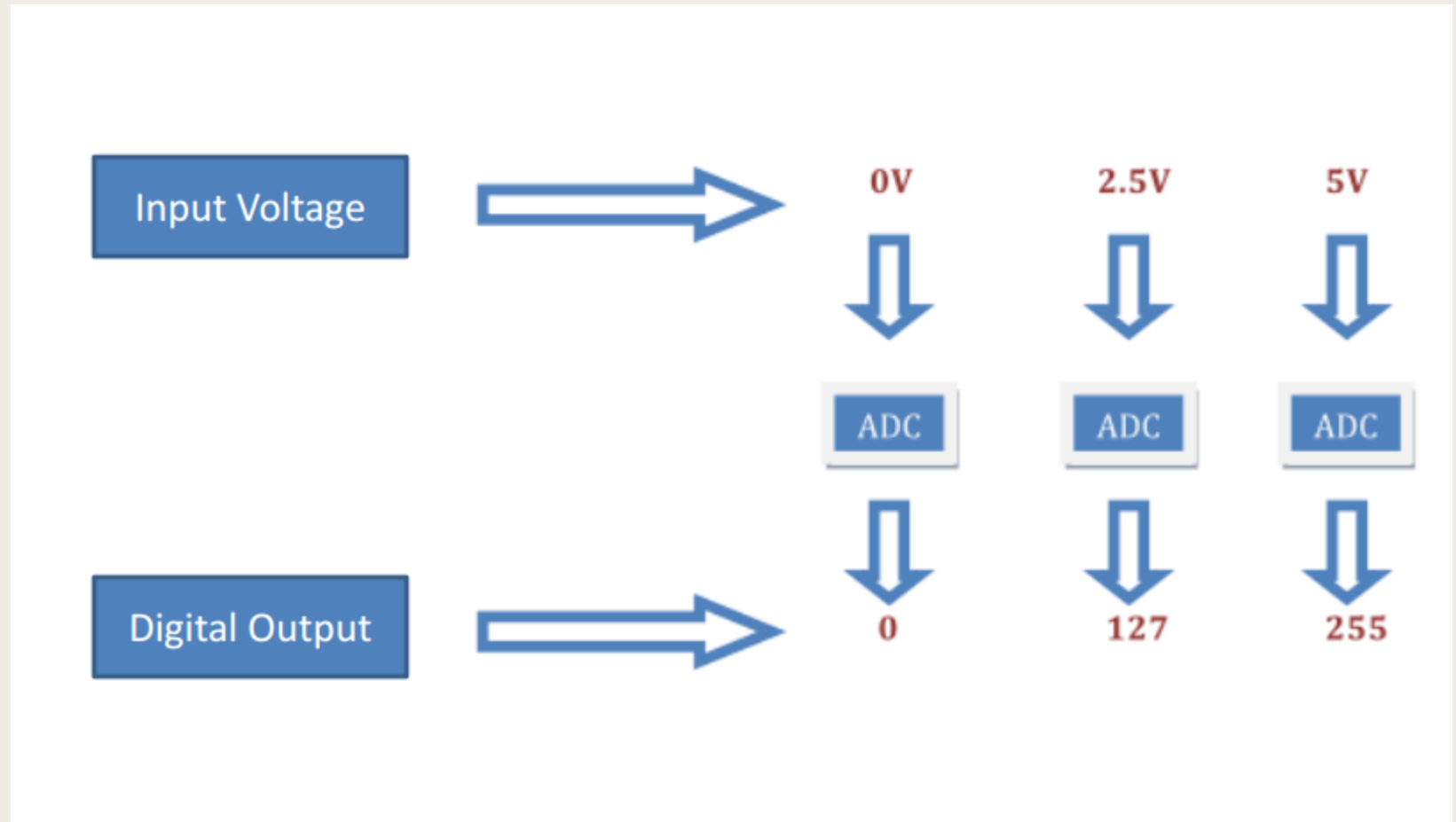
Analog to Digital Conversion

- What is analog ?
 - *It is continuous range of voltage values (not just 0 or 5V)*
- Why convert to digital ?
 - *Because our microcontroller only understands digital.*

ADC in Arduino Uno



Converting Analog Value to Digital



ADC in Arduino

- The Arduino Uno board contains 6 pins for ADC
- 10-bit analog to digital converter
- This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023

Reading/Writing Analog Values

- `analogRead(A0);` // used to read the analog value from the pin A0
- `analogWrite(2,128);`

Our First Program

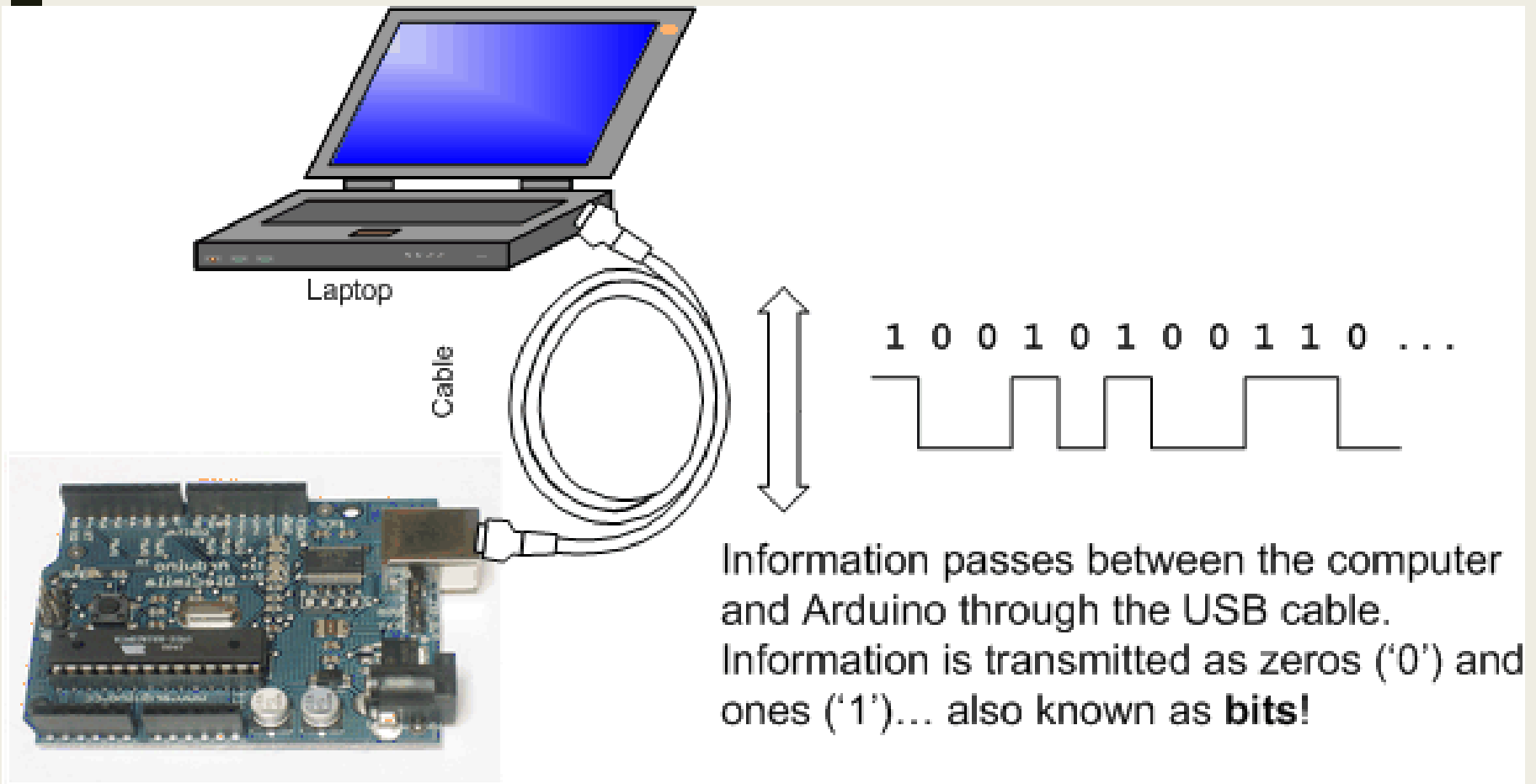
Blink

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  // set the LED on  
  delay(1000);             // wait for a second  
  digitalWrite(13, LOW);   // set the LED off  
  delay(1000);             // wait for a second  
}
```

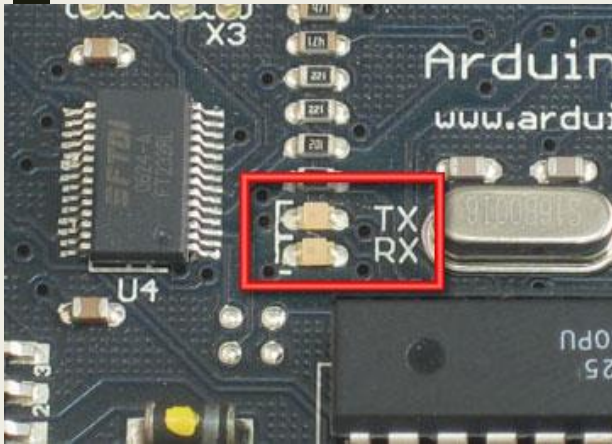
Arduino Timing

- `delay (ms)`
 - Pauses for a few milliseconds
- `delayMicroseconds (us)`
 - Pauses for a few microseconds

Serial Communication



Serial Communication



- **Compiling** turns your program into binary data (ones and zeros)
- **Uploading** sends the bits through USB cable to the Arduino
- The two LEDs near the USB connector blink when data is transmitted
 - **RX** blinks when the Arduino is receiving data
 - **TX** blinks when the Arduino is transmitting data

Some Commands

- `Serial.begin()`
 - e.g., `Serial.begin(9600)`
- `Serial.print()` or `Serial.println()`
 - e.g., `Serial.print(value)`
- `Serial.read()`
- `Serial.write()`

ADC Example

// These constants won't change. They're used to give names to the pins used:

`const int analogInPin = A0; // Analog input pin`

`const int analogOutPin = 9; // Analog output pin that the LED is attached to`

`int sensorValue = 0; // value read from A0`

`int outputValue = 0; // value output`

`void setup() {`

`// initialize serial communications at 9600 bps:`

`Serial.begin(9600);`

`}`

```
void loop() {  
  // read the analog in value:  
  sensorValue = analogRead(analogInPin);  
  // map it to the range of the analog out:  
  outputValue = map(sensorValue, 0, 1023, 0, 255);  
  // change the analog out value:  
  analogWrite(analogOutPin, outputValue);  
  
  // print the results to the serial monitor:  
  Serial.print("sensor = " );  
  Serial.print(sensorValue);  
  Serial.print("\t output = ");  
  Serial.println(outputValue);  
  
  // wait 2 milliseconds before the next loop  
  // for the analog-to-digital converter to settle  
  // after the last reading:  
  delay(2);  
}
```


Good References

www.arduino.cc

www.ladyada.net/learn/arduino

www.EarthshineElectronics.com

■ Contact:

- *Dr. Swarna Priya RM /Prof. Kumaresan*
 - SJT 111 (A10) / SJT 116 (A24)
 - swarnapriya.rm@vit.ac.in, pkumaresan@vit.ac.in
 - 9486120585, 9842582301