

# CSE423: Embedded System Summer-2020



## Understanding Arduino Code/Sketch (1)

---

```
void setup()  
{  
}  
void loop()  
{  
}
```

But it compiles without error!

# Today's Lecture

---

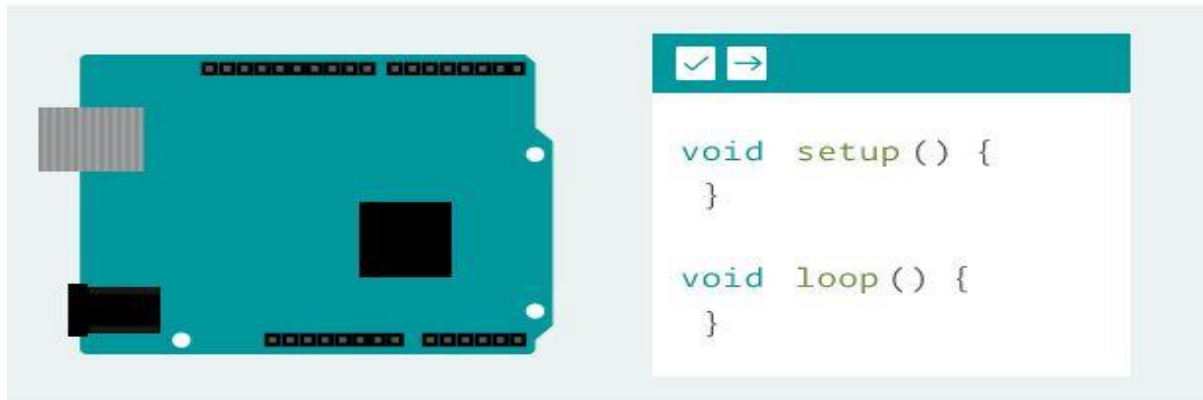


- *Understanding Arduino Code (Sketch) step by step*
- *Implementation of code*

# Understanding Codes



- ❑ Based on C/C++
- ❑ The basic function has two required sub-routines:
  - `void setup()`
  - `void loop()`
- ❑ **Includes** and **define** as defined as C



# Understanding Codes

---



- A simple program: Do Nothing

```
void setup()  
{  
}  
void loop()  
{  
}
```

**But it compiles without error!**

# Understanding Codes



- ❑ Link: <https://www.arduino.cc/en/Reference/HomePage>

The screenshot shows the Arduino Reference website. The navigation menu at the top includes: HOME, BUY, SOFTWARE, PRODUCTS, LEARNING, COMMUNITY, SUPPORT, and SIGN IN. The 'LEARNING' menu is expanded, showing sub-items: GETTING STARTED, TUTORIALS, REFERENCE (highlighted), and EDUCATION. A grey callout box with the text 'Click on the Reference Tab.' has an arrow pointing to the 'REFERENCE' tab. Below the navigation, the page title is 'Language Reference' and the main content area is divided into three columns: Structure, Variables, and Functions. The 'Structure' column lists 'setup()' and 'loop()'. The 'Variables' column lists 'Constants' (HIGH | LOW, INPUT | OUTPUT | INPUT\_PULLUP, LED\_BUILTIN, true | false, integer constants, floating point constants) and 'Data Types'. The 'Functions' column lists 'Digital I/O' (pinMode(), digitalWrite(), digitalRead()) and 'Analog I/O' (analogReference(), analogRead()).

# Understanding Codes: Constants



- **Digital:**
  - **HIGH** | **LOW** (Defining Pin Levels)
  - **true** | **false** (Defining Logic Levels)
  
- **GPIO Configuration:**
  - **INPUT** | **OUTPUT** (Defining Pin Levels)
  
- **Numerical:**
  - **Integer:** **B11010101**, **123**, **-57**, **0x3C**
  - **Float:** **-1.2**, **1.7e5**, **-62E-12**

# Understanding Codes: Data Types



<b>int</b> <small>16b signed</small>	<b>long</b> <small>32b signed</small>	<b>char</b> <small>8b " " signed</small>	<b>float</b> <small>32b signed</small>
<b>unsigned int</b> <small>16b unsigned</small>	<b>unsigned long</b> <small>32b unsigned</small>	<b>boolean</b> <small>1b/8b binary</small>	<b>double</b> <small>32b signed</small>
<b>word</b> <small>16b unsigned</small>	<b>byte</b> <small>8b unsigned</small>	<b>string</b> <small>8b*nchar " " signed</small>	<b>array</b> <small>8b*nelem unsigned</small>
<b>short</b> <small>16b signed</small>	<b>unsigned char</b> <small>8b signed</small>	<b>String</b> <small>?? signed</small>	<b>void</b> <small>0b signed</small>

# Understanding Codes: **Basic Operators**

---

Arithmetic	Comparison	Logic/Pointer	Bitwise	Assignment
=	==	&&	&	++
+	!=			--
-	<	!	^	+=
*	>		~	--=
/	<=	*var	<<	*=, /=
%	>=	&var	>>	&=,  =





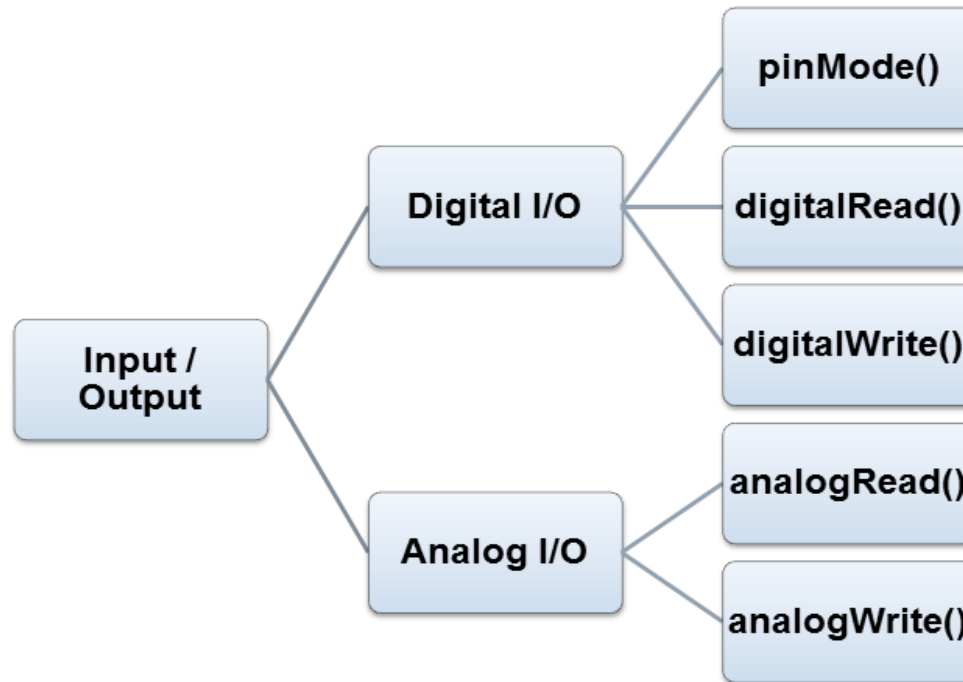
# Understanding Codes: **Flow controls**

---

- `if`
- `if...else`
- `for`
- `switch case`
- `while`
- `do... while`
- `break`
- `continue`
- `return`
- `goto`



# Understanding Codes: **Basic I/O**



# Understanding Codes: Digital I/O



## □ 1. pinMode()

### pinMode()

#### Description

Configures the specified pin to behave either as an input or an output.

#### Syntax

```
pinMode(pin, mode)
```

#### Parameters

pin: the number of the pin whose mode you wish to set

mode: `INPUT`, `OUTPUT` (see the [digital pins](#) page for a more complete description of the functionality.)

#### Returns

None

```
int ledPin = 13;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

# Understanding Codes: Digital I/O



## □ 2. digitalWrite( )

### digitalWrite()

#### Description

Write a **HIGH** or a **LOW** value to a digital pin.

NOTE: If you do not set the `pinMode()` to **OUTPUT**, and connect an LED to a pin, when calling `digitalWrite(HIGH)`, the LED may appear dim.

#### Syntax

`digitalWrite(pin, value)`

#### Parameters

pin: the pin number

value: **HIGH** or **LOW**

#### Returns

none

```
int ledPin = 13;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

# Understanding Codes: Digital I/O

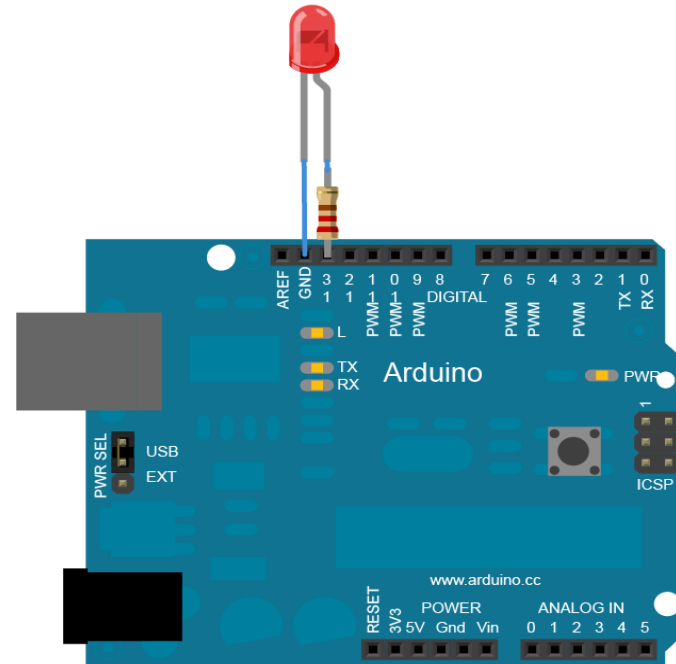


## □ 2. digitalWrite()

```
int ledPin = 13;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

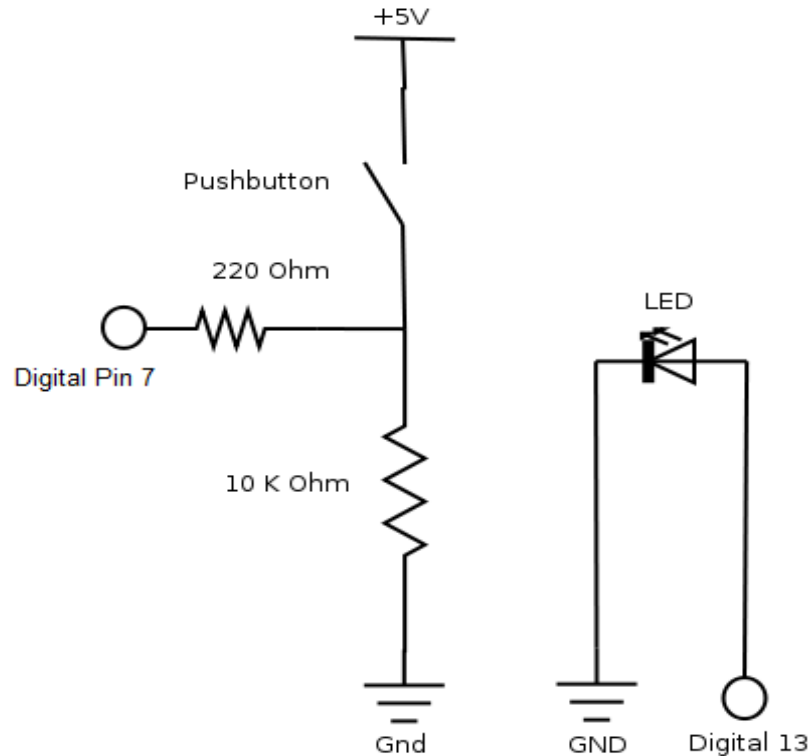
void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```



# Understanding Codes: Digital I/O



## □ 3. digitalRead()

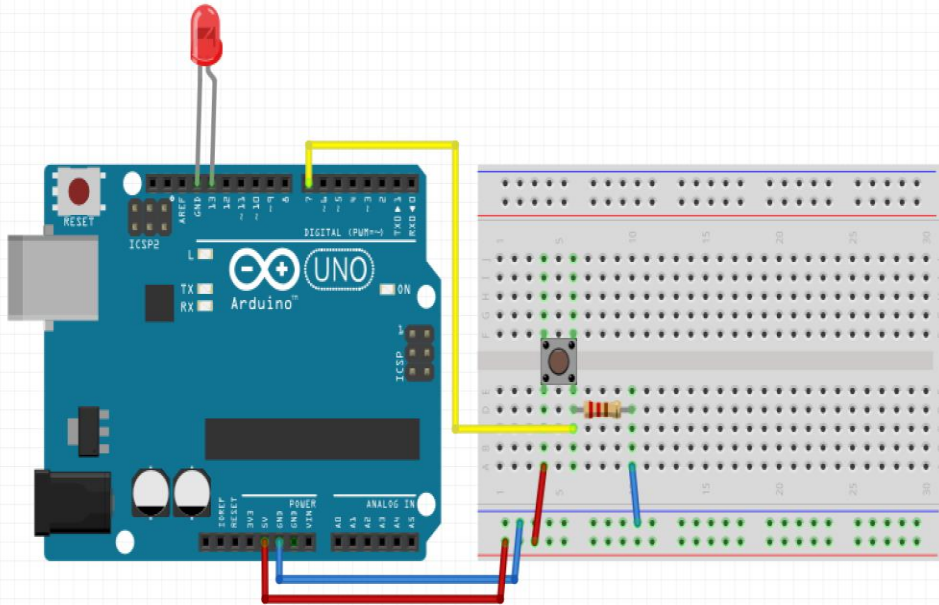


```
int ledPin = 13;
int inPin = 7;
int val = 0;

void setup()
{
  pinMode(ledPin, OUTPUT);
  // sets the digital pin 13 as output
  pinMode(inPin, INPUT);
  // sets the digital pin 7 as input
}

void loop()
{
  val = digitalRead(inPin);
  // read the input pin
  digitalWrite(ledPin, val);
  // sets the LED to the button's value
}
```

# Understanding Codes: Digital I/O



```
int ledPin = 13;
int inPin = 7;
int val = 0;

void setup()
{
  pinMode(ledPin, OUTPUT);
  // sets the digital pin 13 as output
  pinMode(inPin, INPUT);
  // sets the digital pin 7 as input
}

void loop()
{
  val = digitalRead(inPin);
  // read the input pin
  digitalWrite(ledPin, val);
  // sets the LED to the button's value
}
```