CEC Continuing Education Center

DesignNews

PLC-HMI Automation Applications

DAY 2 : Writing plcLib Applications

Sponsored by

Digi-Key Electronics

informa markets

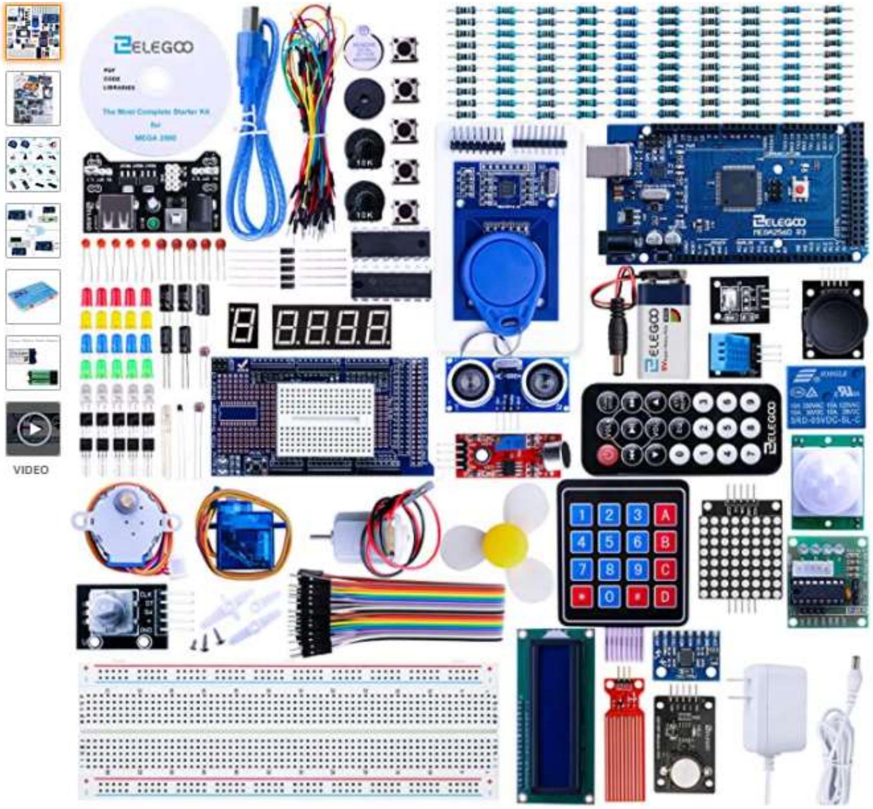
Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.

Don Wilcher

Visit ‘Lecturer Profile’ in your console for more details.

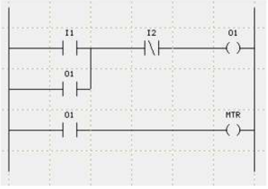
Course Kit:
The ELEGOO Mega 2560 Project: The Most Complete Starter Kit w/Tutorial



Course Components:

ELEGOO UNO R3 2.8 Inches TFT Touch Screen with SD Card Socket w/All Technical Data in CD for Arduino UNO R3

TWTADE SSR-40 DD 40A DC 3-32V to DC 5-60V SSR Solid State Relay + Heat Sink

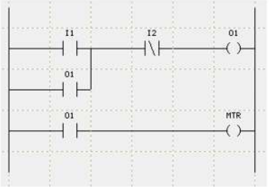


Agenda:

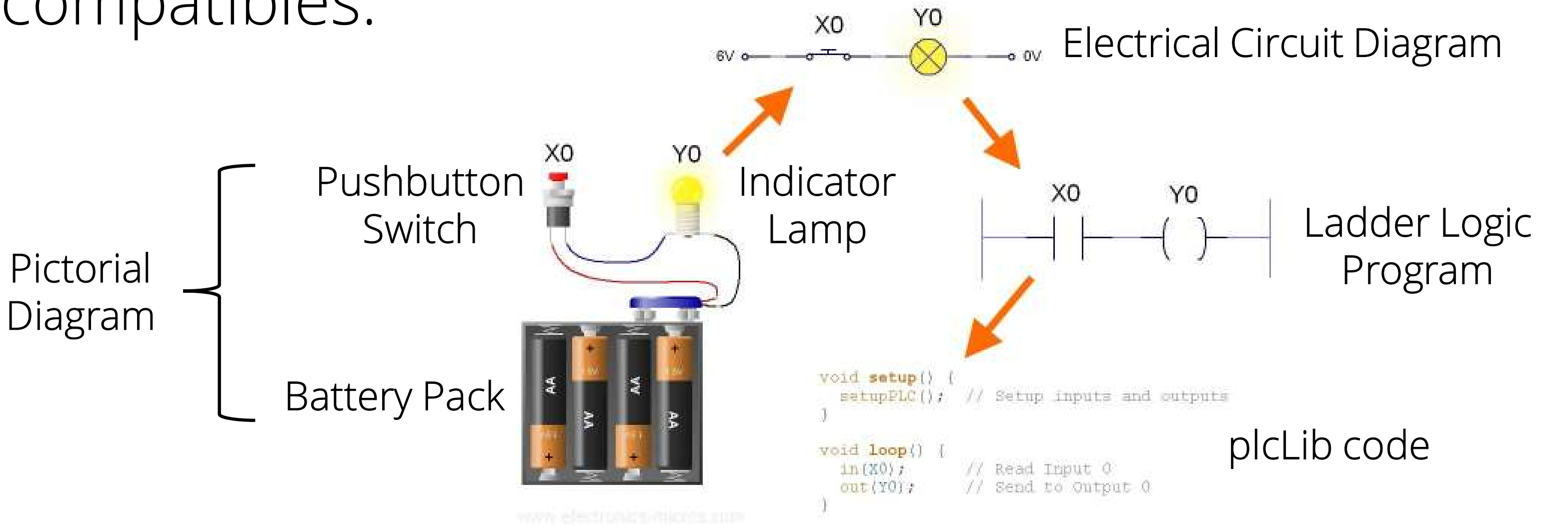
- What is plcLib?
- Basic Logic Gates
- Construction of a Ladder Logic Program
- Lab Activities

  a) Building an Arduino PLC Controller-Concept
  b) plcLib NOT Logic Gate
  c) plcLib AND Logic Gate
  d) plcLib OR Logic Gate

What is plcLib?



A library allowing PLC-style control – oriented software applications to be developed for Arduino and compatibles.



Electrical Circuit Diagram

Pictorial Diagram

Pushbutton Switch

Indicator Lamp

Battery Pack

Ladder Logic Program

plcLib code

```
void setup() {
  setupPLC();  // Setup inputs and outputs
}

void loop() {
  in(X0);       // Read Input 0
  out(Y0);      // Send to Output 0
}
```

Source: plcLib(Arduino) User Guide(https://github.com/wditch/plcLib)

Question 1

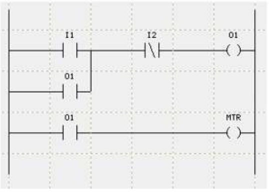
In reviewing slide 7, what electrical components are referenced by X0 and Y0 designators?

a) Battery Pack

b) Pushbutton Switch and Indicator Lamp

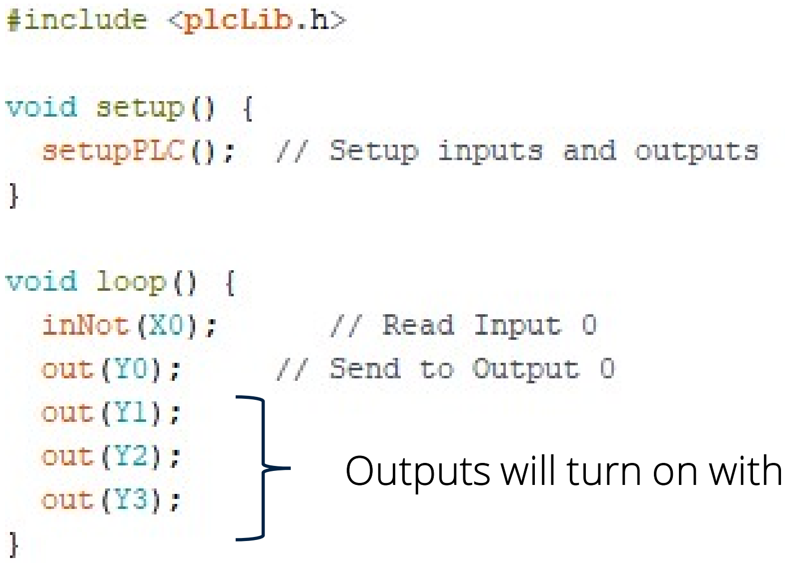
c) None of the Above

What is plcLib?



Instruction List (IL):

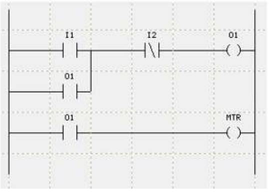
plcLib is similar in structure to Instruction List. Both languages are formatted in a top-down approach



```
#include <plcLib.h>

void setup() {
  setupPLC();  // Setup inputs and outputs
}

void loop() {
  inNot(X0);      // Read Input 0
  out(Y0);     // Send to Output 0
  out(Y1);
  out(Y2);
  out(Y3);
}
```

Outputs will turn on with Y0

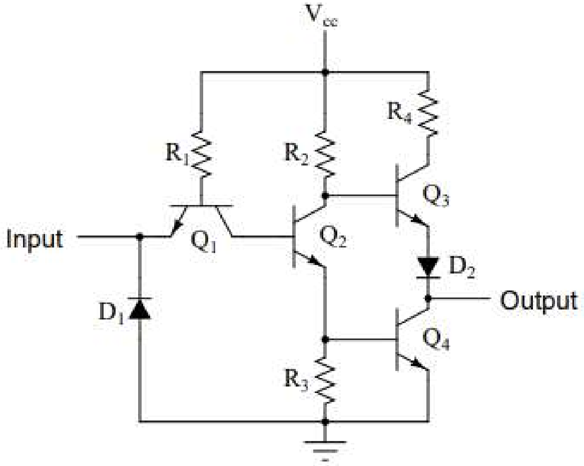
Source: plcLib(Arduino) User Guide (https://github.com/wditch/plcLib)



Basic Logic Gates

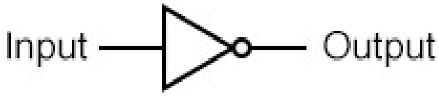
The Basic Logic Gates used in PLC ladder logic programs consists of:

Transistor-Transistor Logic (TTL) Circuit

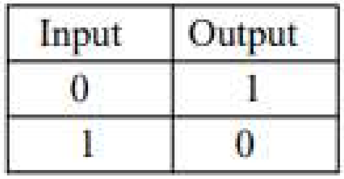


Logic Gate Symbol

NOT or Inverter



Truth Table

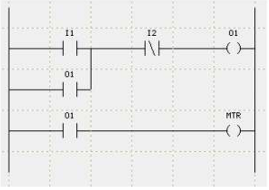


Input	Output
0	1
1	0

Boolean Expression:
Input = (NOT)Output

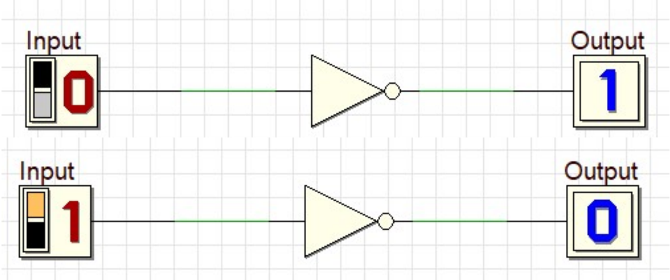
Source: Lessons in Electric Circuits, Volume IV – Digital (https://www.allaboutcircuits.com/assets/pdf/digital.pdf)

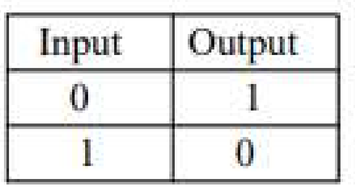
Basic Logic Gates. . .



Functional Operation of the NOT Gate:

NOT

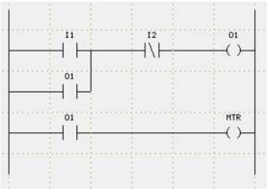




Input	Output
0	1
1	0

Boolean Expression:
Input = (NOT)Output

Source: Lessons in Electric Circuits, Volume IV – Digital (https://www.allaboutcircuits.com/assets/pdf/digital.pdf)

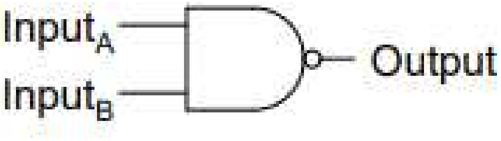


Basic Logic Gates

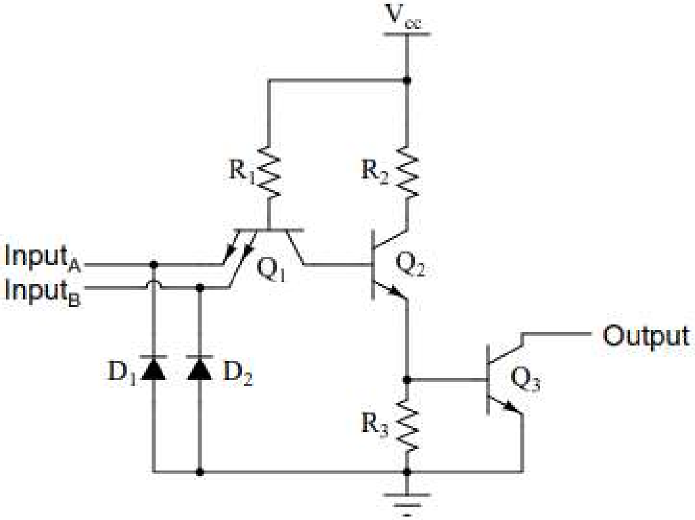
The Basic Logic Gates used in PLC ladder logic programs consists of:

Logic Gate Symbol

NAND

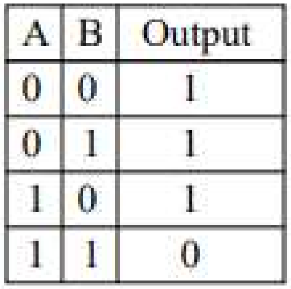


Transistor-Transistor Logic (TTL) Circuit



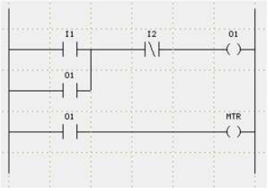
Boolean Expression:
Output = NOT(AB)

Truth Table



A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

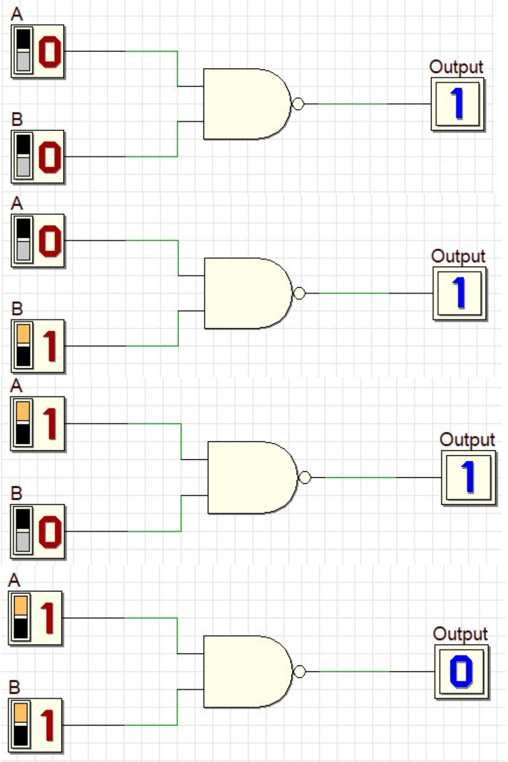
Source: Lessons in Electric Circuits, Volume IV – Digital (https://www.allaboutcircuits.com/assets/pdf/digital.pdf)



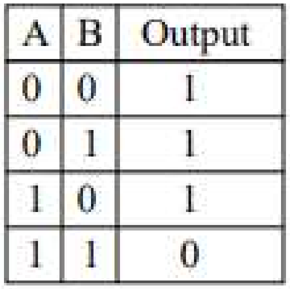
Basic Logic Gates. . .

Functional Operation of the NAND Gate:

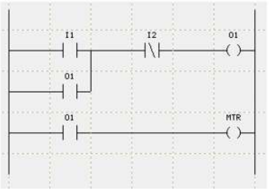
NAND



Boolean Expression:
Output = NOT(AB)



A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

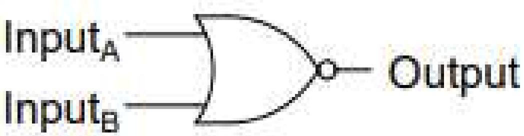


Basic Logic Gates

The Basic Logic Gates used in PLC ladder logic programs consists of:

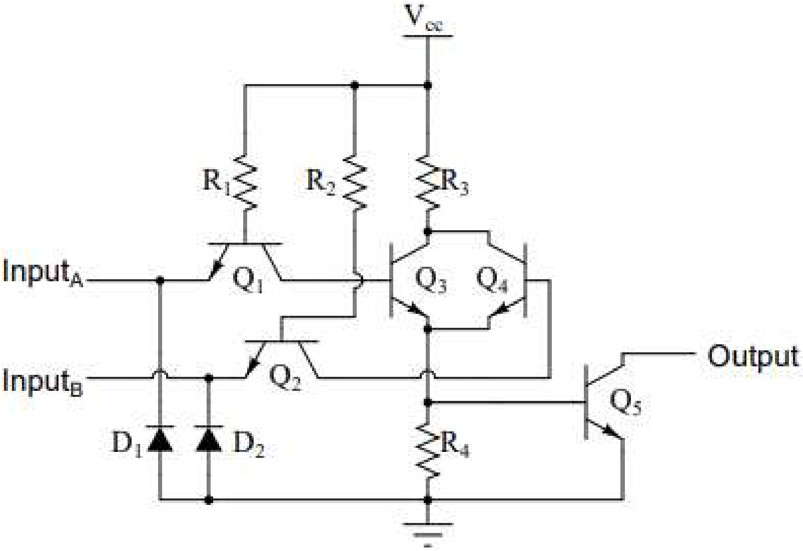
Logic Gate Symbol

NOR

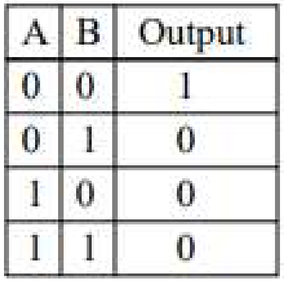


Boolean Expression:
Output = NOT(A+B)

Transistor-Transistor Logic (TTL) Circuit

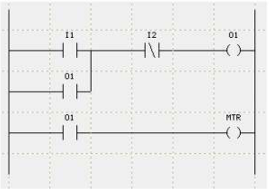


Truth Table



A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

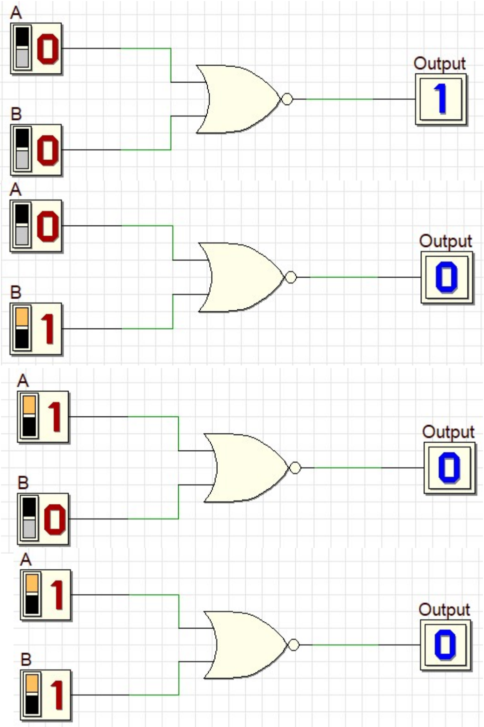
Source: Lessons in Electric Circuits, Volume IV – Digital (https://www.allaboutcircuits.com/assets/pdf/digital.pdf)



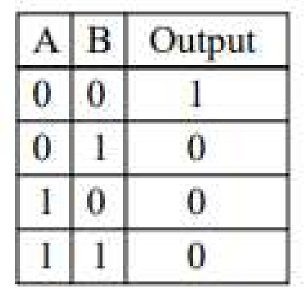
Basic Logic Gates. . .

Functional Operation of the NOR Gate:

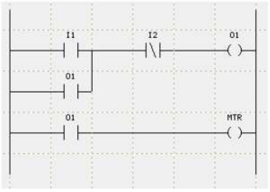
NOR



Boolean Expression:
Output = NOT(A+B)



A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

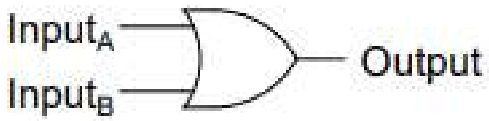


Basic Logic Gates. . .

The Basic Logic Gates used in PLC ladder logic programs consists of:

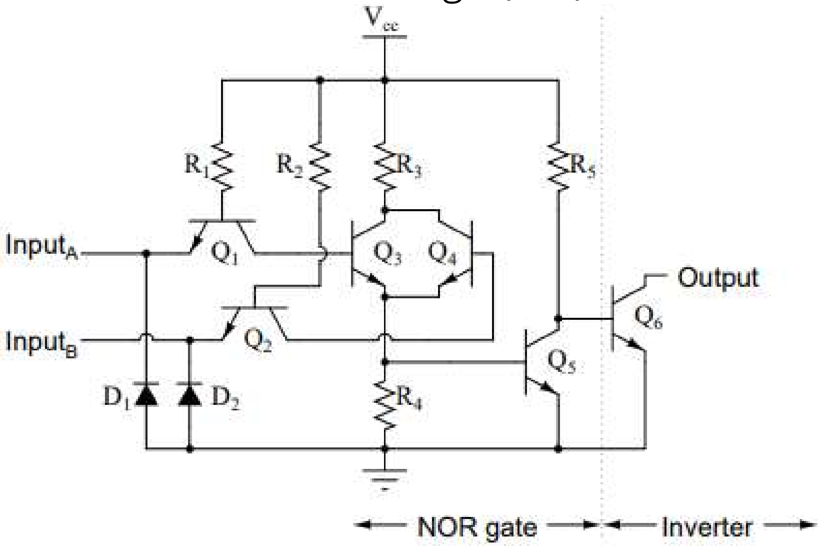
Logic Gate Symbol

OR

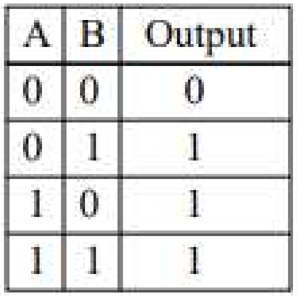


Boolean Expression:
Output = A+B

Transistor-Transistor Logic (TTL) Circuit

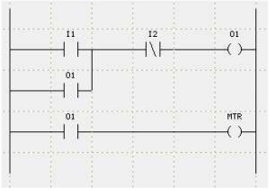


Truth Table



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

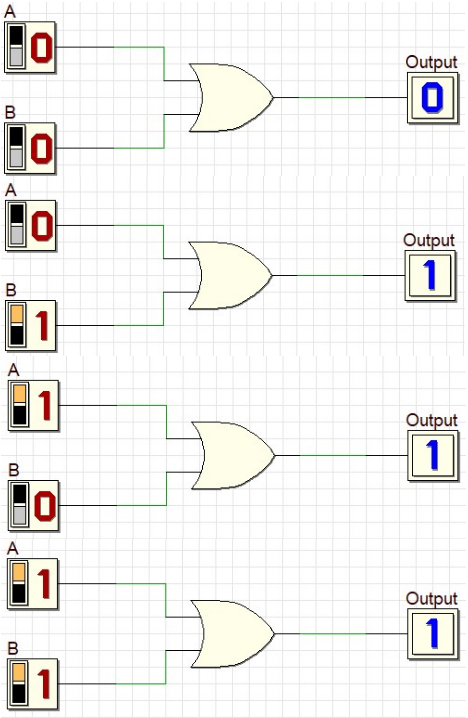
Source: Lessons in Electric Circuits, Volume IV – Digital (https://www.allaboutcircuits.com/assets/pdf/digital.pdf)



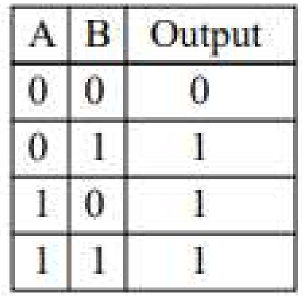
Basic Logic Gates. . .

Functional Operation of the OR Gate:

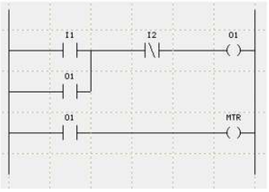
OR



Boolean Expression:
Output = A+B



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

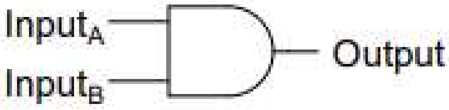


Basic Logic Gates. . .

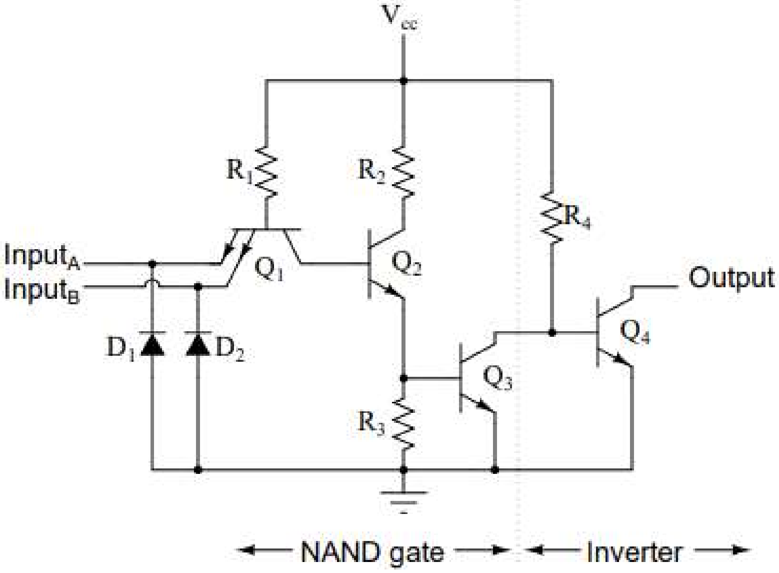
The Basic Logic Gates used in PLC ladder logic programs consists of:

Logic Gate Symbol

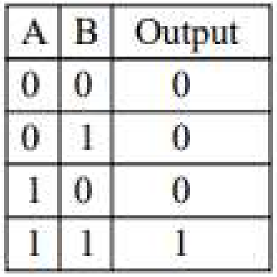
AND



Transistor-Transistor Logic (TTL) Circuit



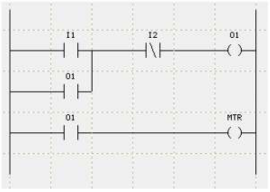
Truth Table



A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Boolean Expression:
Output = AB

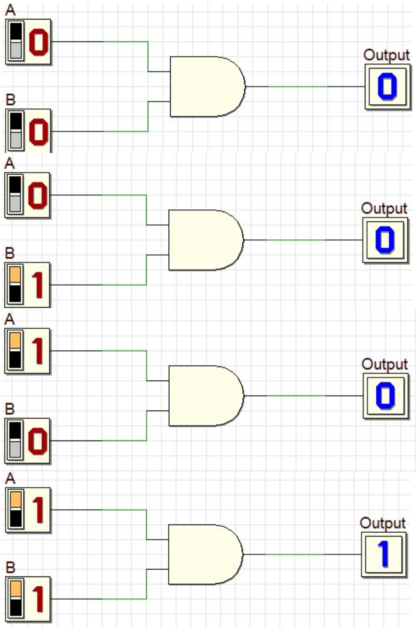
Source: Lessons in Electric Circuits, Volume IV – Digital (https://www.allaboutcircuits.com/assets/pdf/digital.pdf)



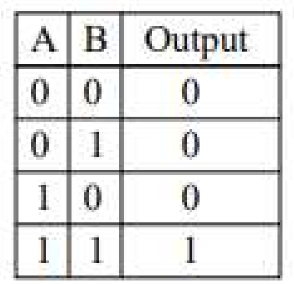
Basic Logic Gates. . .

Functional Operation of the AND Gate:

AND



Boolean Expression:
Output = AB



A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Question 2

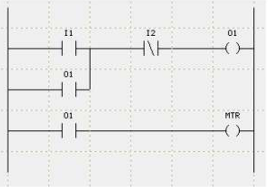
Which logic gate is used to provide signal inversion?

a) AND

b) OR

c) NOT

d) None of the Above

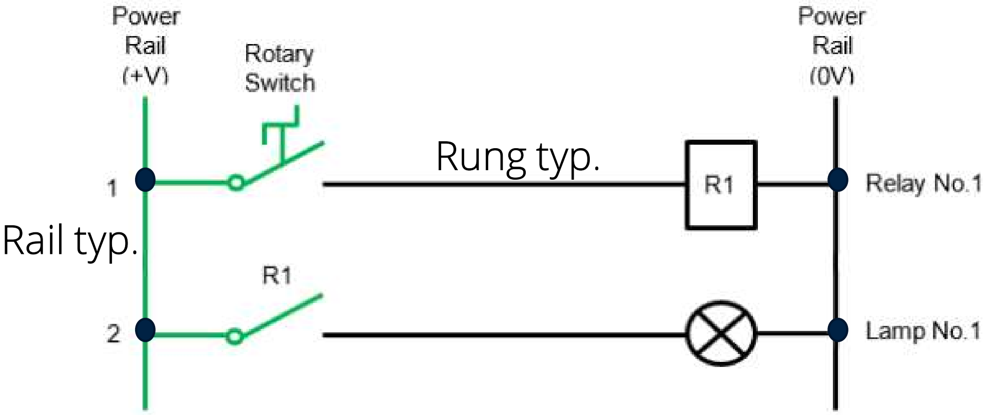


Construction of a Ladder Logic Program

Ladder logic is a programming language is:

- Used to program a PLC (Programmable Logic Controller).
- It is a graphical PLC programming language
- Used to express logic operations with symbolic notation using ladder diagrams

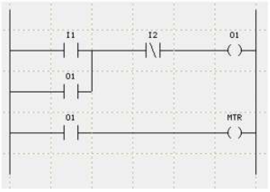
Ladder logic is much like the rails and rungs of a traditional relay logic circuit.



Rung typ.

Rail typ.

Source: Ladder Logic World (https://ladderlogicworld.com/relay-logic-vs-ladder-logic/)

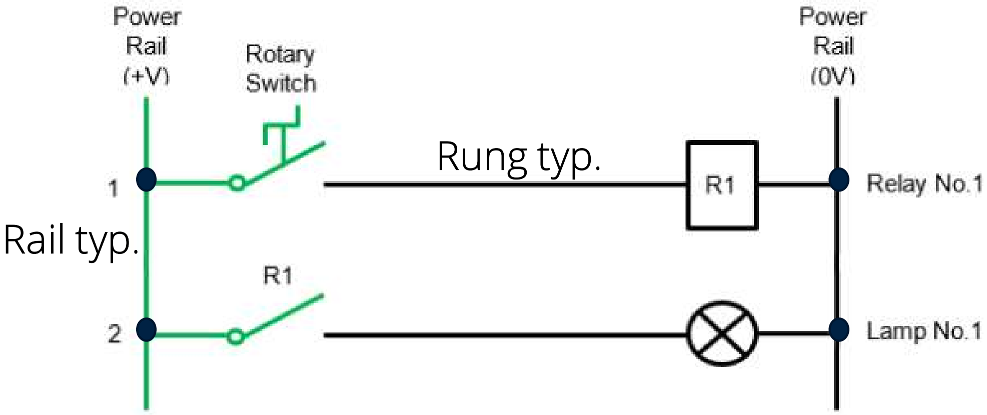


Construction of a Ladder Logic Program. . .

Ladder diagrams have:

- horizontal lines of control logic called rungs
- vertical lines at the start and end of each rung called rails.

The structure of a ladder diagram looks like a ladder, hence the name "ladder diagram".



Source: Ladder Logic World (https://ladderlogicworld.com/relay-logic-vs-ladder-logic/)

Question 3

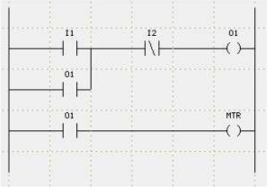
Ladder Diagrams have

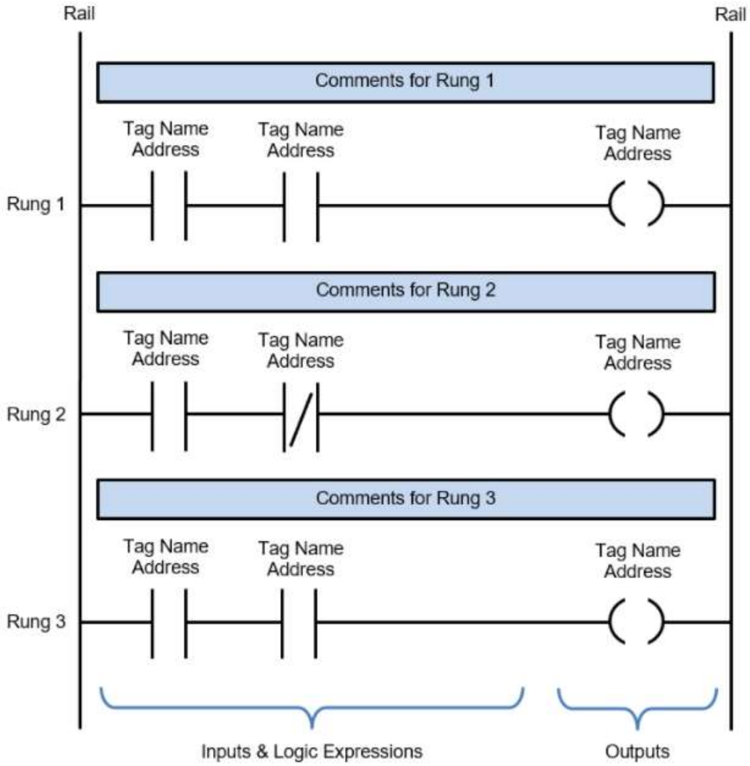
a) horizontal lines of control logic called rungs

b) vertical lines on top and bottom of the diagram

c) None of the Above

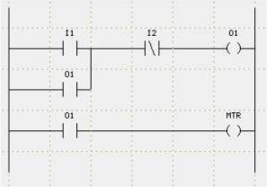
Construction of a Ladder Logic Program. . .



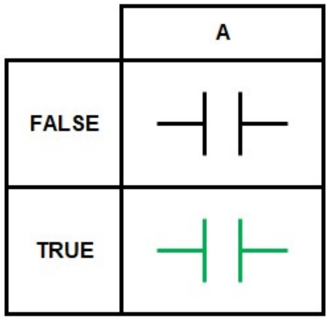


Source: Ladder Logic World (https://ladderlogicworld.com/ladder-logic-basics/)

Construction of a Ladder Logic Program. . .

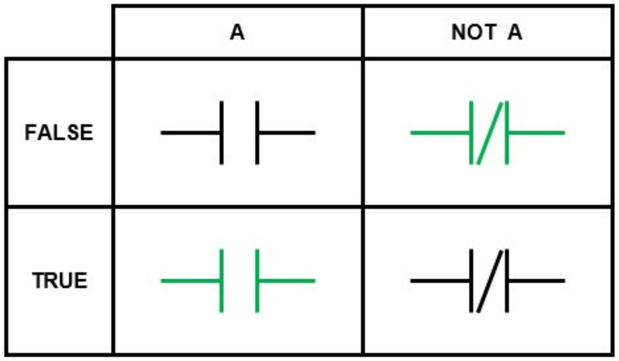


Examine If Closed (XIC)



	A
FALSE	—| |—
TRUE	—| |—

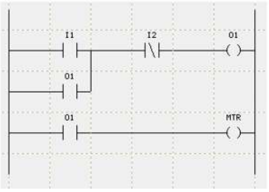
Examine If Open (XIO)



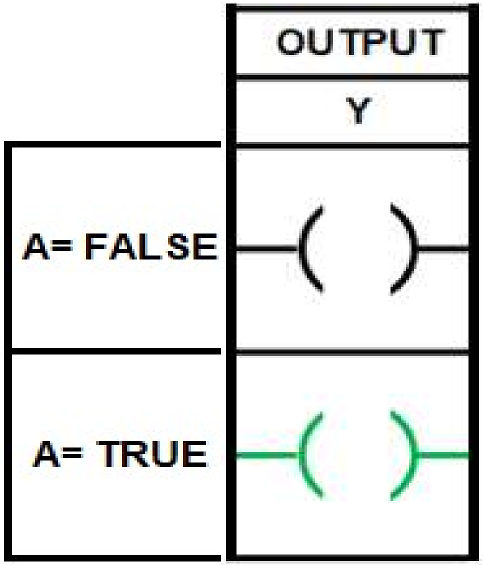
	A	NOT A
FALSE	—| |—	—|/|—
TRUE	—| |—	—|/|—

Source: Ladder Logic World (https://ladderlogicworld.com/ladder-logic-basics/)

Construction of a Ladder Logic Program. . .



Output Energize (OE)



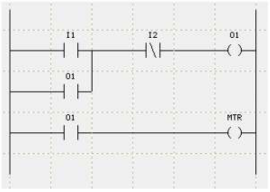
Source: Ladder Logic World (https://ladderlogicworld.com/ladder-logic-basics/)

Question 4

Examine If Closed bit instruction looks at what transitional state of the switching device?

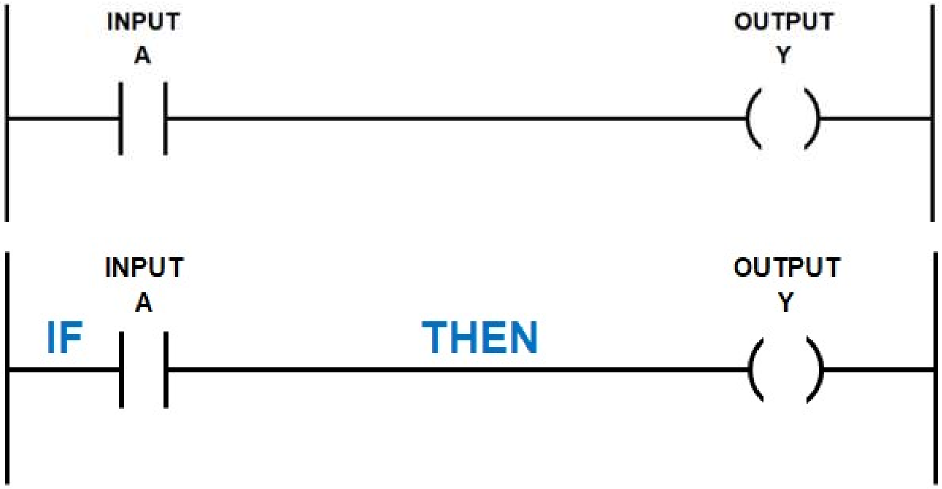
- **a) When the switching device is transitioning from a normally closed to a normally open state**
- **b) When the switching device is transitioning from a normally open to a normally closed state.**
- **c) An intermediate state**
- **d) None of the Above**

Construction of a Ladder Logic Program. . .



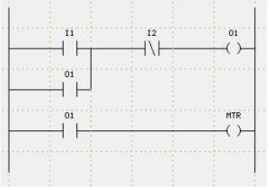
How to Read a Ladder Logic Program

Hello World Ladder Logic Program



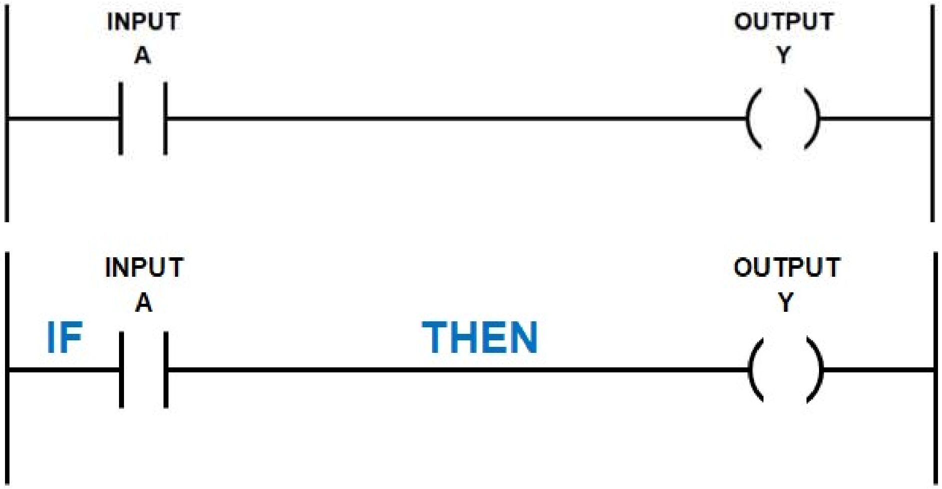
Source: Ladder Logic World (https://ladderlogicworld.com/ladder-logic-basics/)

Construction of a Ladder Logic Program. . .

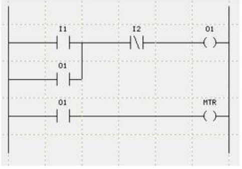


How to Read a Ladder Logic Program

Hello World Ladder Logic Program

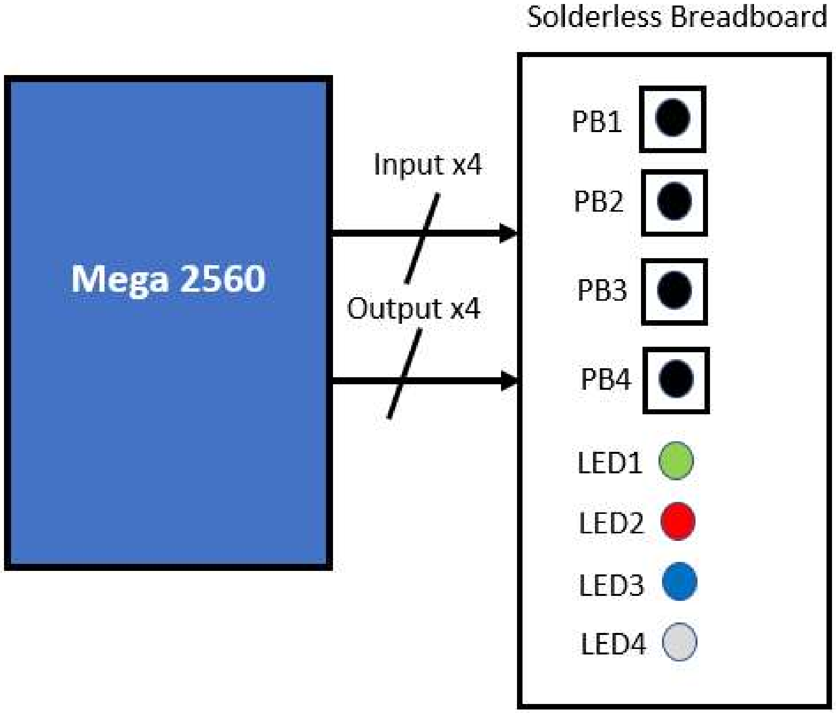


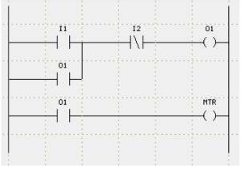
Source: Ladder Logic World (https://ladderlogicworld.com/ladder-logic-basics/)



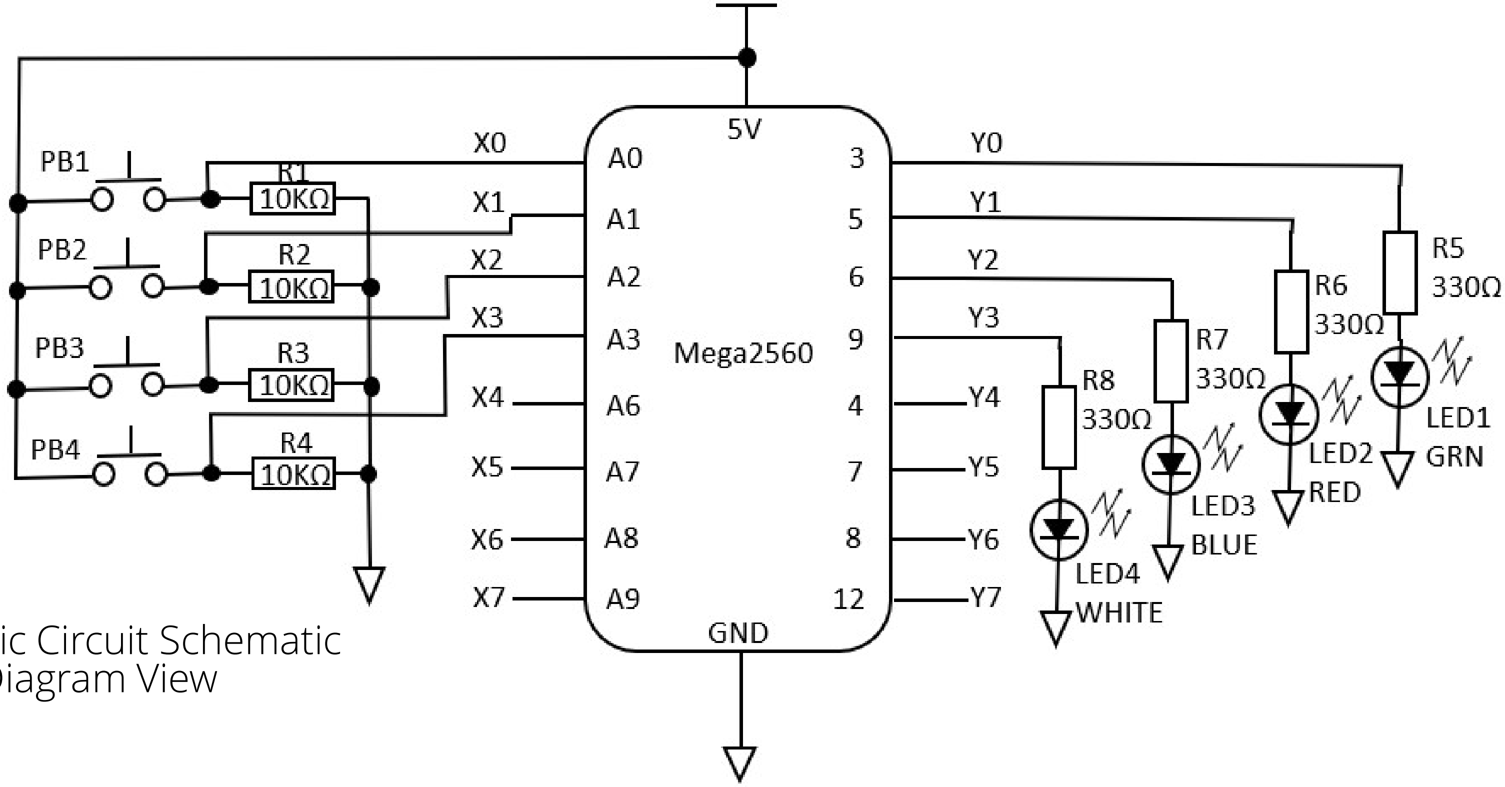
Lab Activities
Building an Arduino PLC Controller - Concept

Functional Block Diagram View

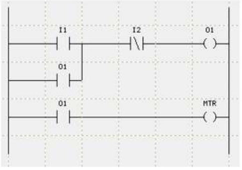




Lab Activities. . .
Building an Arduino PLC Controller - Concept

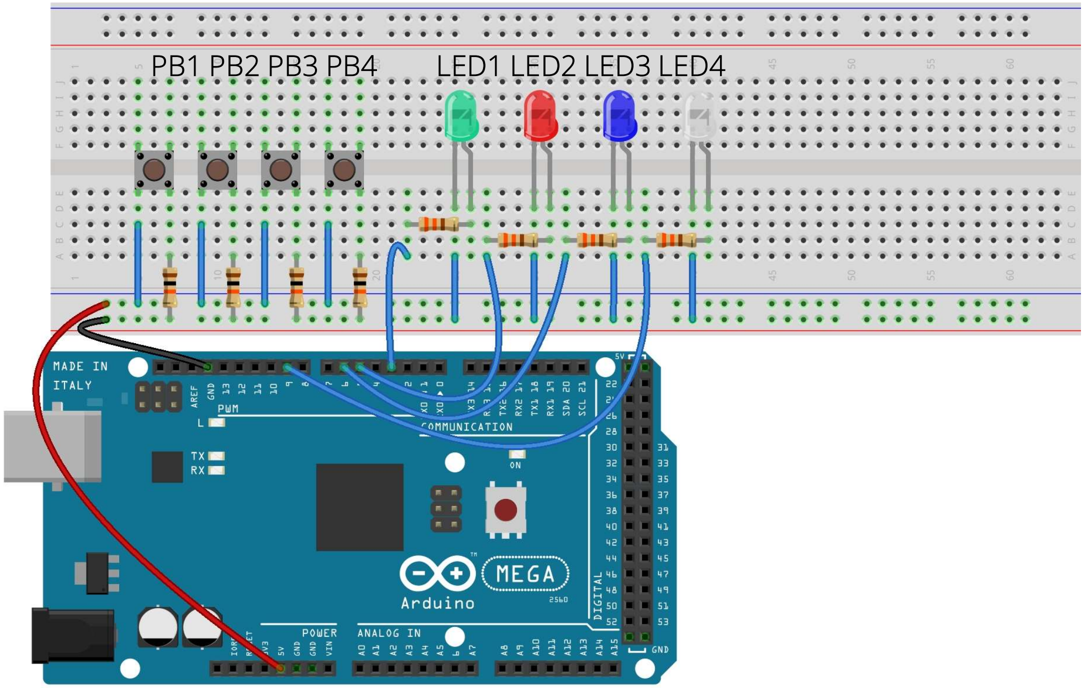


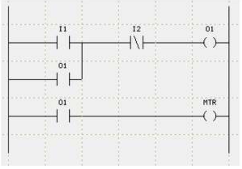
Electronic Circuit Schematic Diagram View



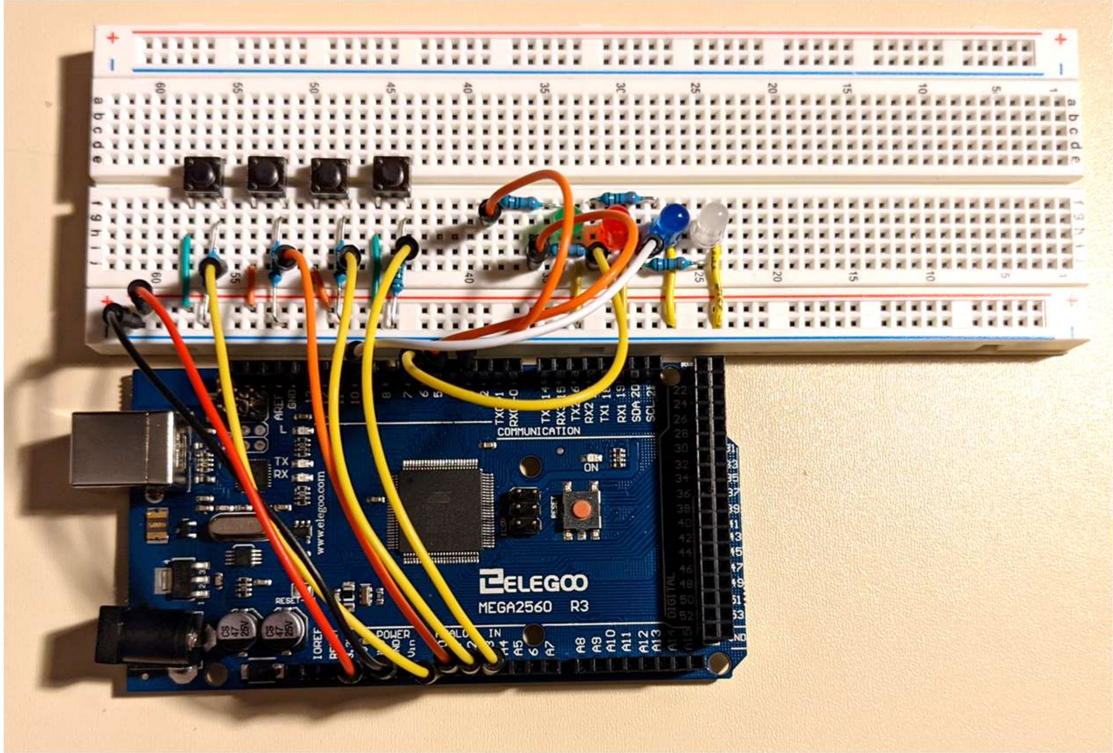
Lab Activities. . .
Building an Arduino PLC Controller - Concept

Breadboard Diagram View

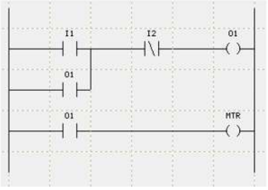




Lab Activities. . .
Building an Arduino PLC Controller - Concept



Actual Wired Breadboard View



Lab Activities:

plcLib NOT Logic Gate

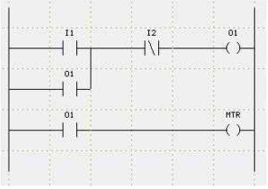
Physical I/O Assignment:

X0 = PB1
Y0 = LED1

```
#include <plcLib.h>

void setup() {
  setupPLC();  // Setup inputs and
outputs
}

void loop() {
  inNot(X0);      // Read Input 0
  out(Y0);     // Send to Output 0
}
```

Invert Input

Output Inverted

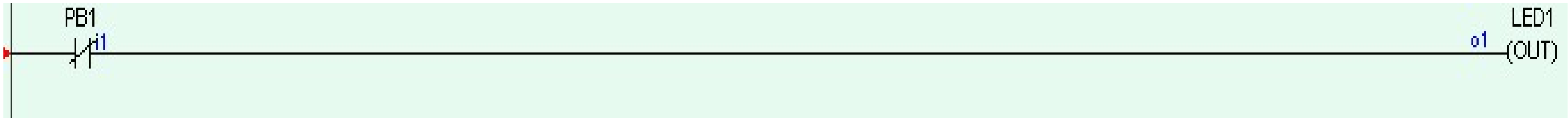


Lab Activities. . .

Physical I/O Assignment:

X0 = PB1
Y0 = LED1

Ladder Logic Program:
NOT Logic Gate

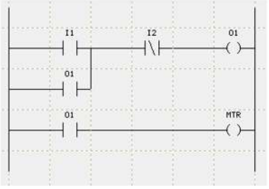


Lab Activities. . .

plcLib AND Logic Gate

Invert Input

Output Inverted



Physical I/O Assignment:

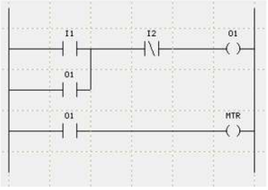
X0 = PB1
X1 = PB2
Y1 = LED2

```
#include <plcLib.h>

void setup() {
  setupPLC();
  }

void loop() {
  in(X0);    //Read input 0
  andBit(X1);// AND with input 1
  out(Y1);   // Send result to output 1

}
```

Input 1

Input 2

Output 1

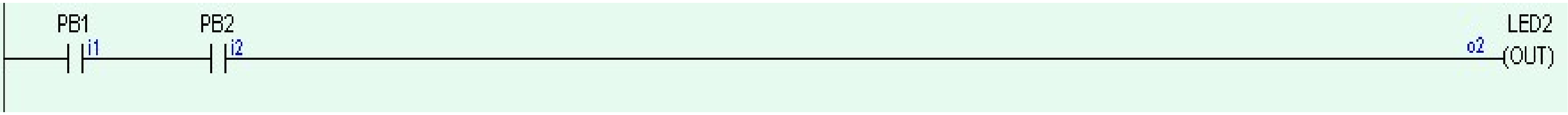


Lab Activities. . .

Physical I/O Assignment:

X0 = PB1
X1 = PB2
Y1 = LED1

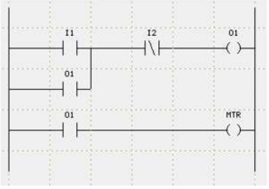
Ladder Logic Program:
AND Logic Gate



Question 5

Referencing the plcLib AND Logic Gate code on slide 36, which instruction will allow a 3-input device to be created?

a) andBit(X1);
b) orBit(X2);
c) andBit(X2);
d) None of the Above



Lab Activities. . .

plcLib OR Logic Gate

Invert Input

Output Inverted

Physical I/O Assignment:

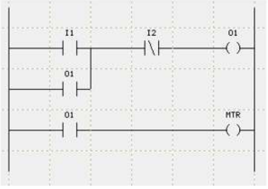
X0 = PB1
X1 = PB2
Y1 = LED2

```
#include <plcLib.h>

void setup() {
  setupPLC();
  }

void loop() {
  in(X0);    //Read input 0
  orBit(X1);// OR with input 1
  out(Y2);   // Send result to output 2
}
```

Input 1

Input 2

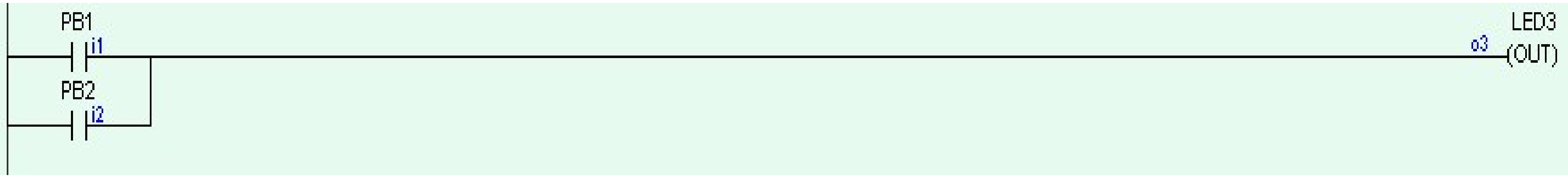
Output 2



Lab Activities. . .

Ladder Logic Program:
OR Logic Gate

Physical I/O
Assignment:
X0 = PB1
X1 = PB2
Y1 = LED3



Thank you for attending

Please consider the resources below:

- plcLib(Arduino) User Guide
  https://github.com/wditch/plcLib
- Lessons in Electric Circuits, Volume IV – Digital
  https://www.allaboutcircuits.com/assets/pdf/digital.pdf
- Ladder Logic World
  https://ladderlogicworld.com/ladder-logic-basics/

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key ELECTRONICS

informa markets