



DesignNews

Embedded Controls Development with OpenPLC

DAY 1 : Introduction to OpenPLC

Sponsored by

DigiKey

 **informa**markets

Webinar Logistics

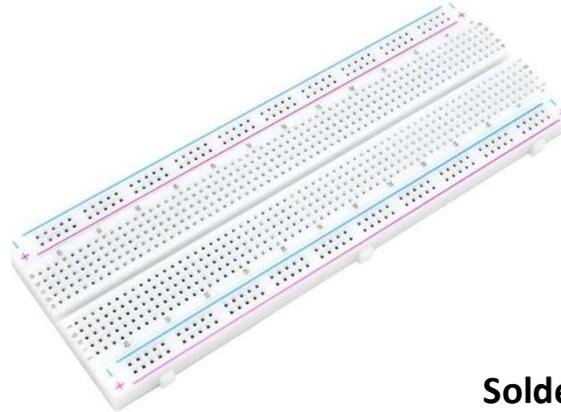
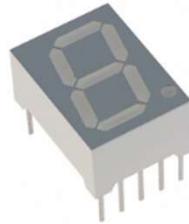
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.



Dr. Don Wilcher

Visit 'Lecturer Profile' in your console for more details.

Course Kit and Materials

ESP32 WROOM32D DEVKITC**Solderless Breadboard x2****Adafruit Parts Pal Kit****L298N Motor Drive Controller****7 Segment LED Display,
Common Cathode****Solderless Breadboard
Power Supply Module with
9V Battery Clip Power Cable**

Agenda:

- Traditional Programming Methods
 - a) C-Language
 - b) C++ Language
- OpenPLC Overview
 - a) IEC61131-3 Functional Programming Language Specification
 - b) Basic BIT Instructions
 - c) Ladder Logic Basics
- Lab: Hello World Demo

Research Perspective

“Embedded electronics is a subfield of electronics that can unite the power of programming with the power of electronics” (Zemmouri et al., 2023).

Traditional Programming Languages

- A traditional programming language is a programming language that is used to write code in a text-based format.
 - a) Often, a compiled language – Translated into a machine language before being executed.
 - b) Compiled languages are faster because of the machine language translation process.
 - c) Interpreted language – A programming language that allows the source code to be executed line by line.

Question 1

A compiled language is translated into a _____ language before being executed.

- a) blockly code**
- b) assembly**
- c) machine**
- d) text-based**

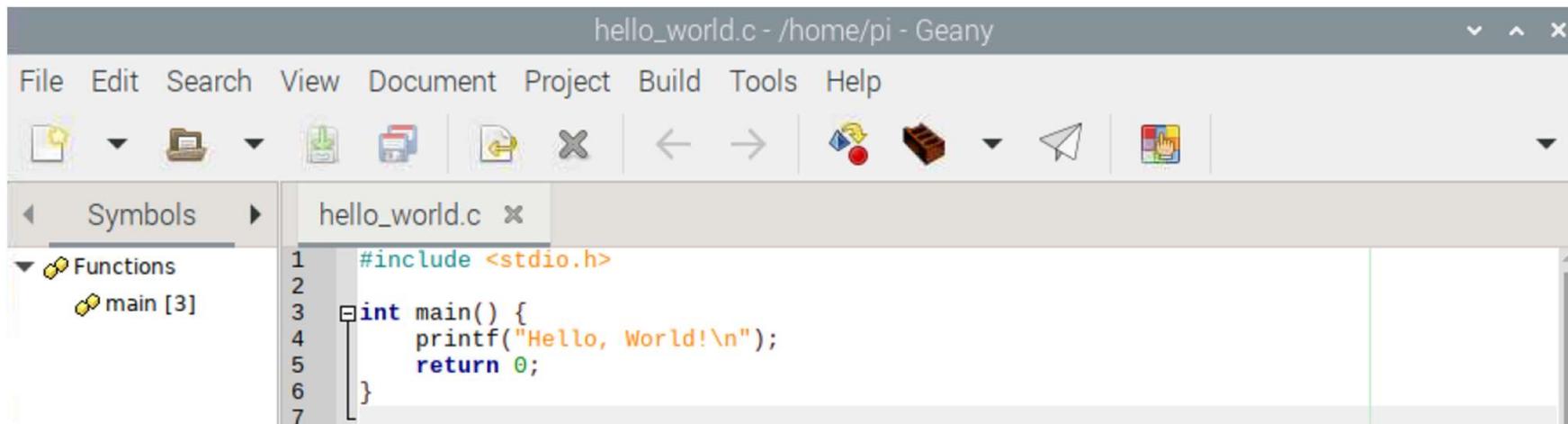


Traditional Programming Languages...

- Examples of traditional programming languages are:
 - a) C
 - b) C++
 - c) Python
 - d) MicroPython

Traditional Programming Languages...

Here is an example of a Hello World Traditional Program using C-language.



The screenshot shows a code editor window titled "hello_world.c - /home/pi - Geany". The menu bar includes File, Edit, Search, View, Document, Project, Build, Tools, and Help. The toolbar contains various icons for file operations and editing. The main editor area displays the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello, World!\n");
5     return 0;
6 }
7
```

The left sidebar shows a "Symbols" view with a tree structure under "Functions" containing "main [3]".

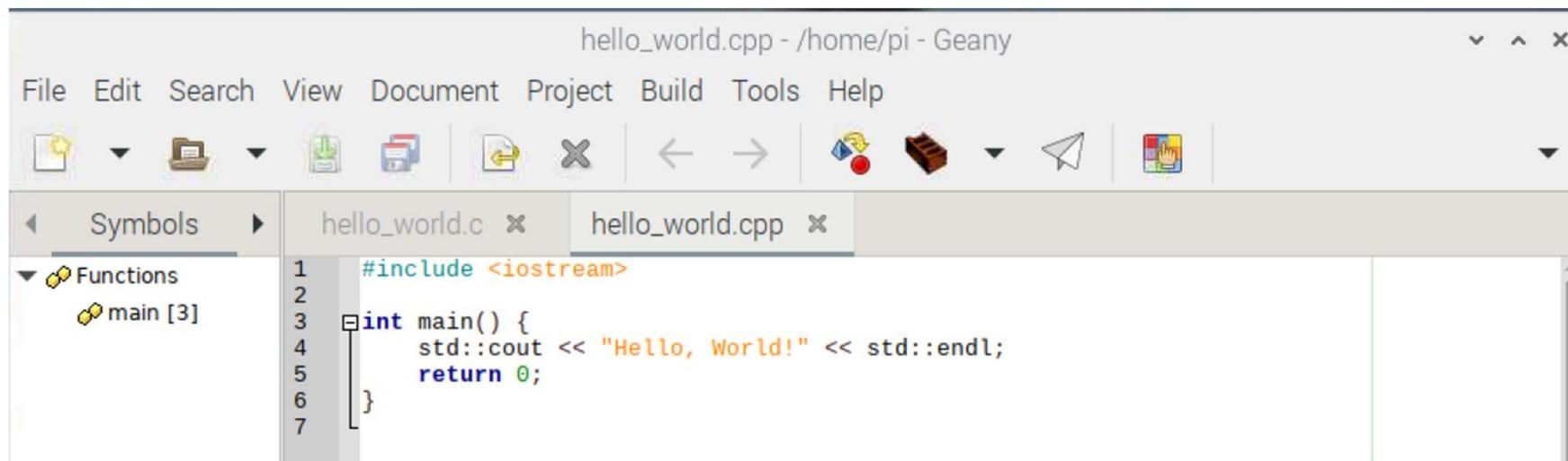
Traditional Programming Languages...

Compiling and Executing the code using the gcc Compiler.

```
pi@raspberrypi:~ $ gcc -o myprog hello_world.c
pi@raspberrypi:~ $ ./myprog
Hello, World!
pi@raspberrypi:~ $ █
```

Traditional Programming Languages...

Here is an example of a Hello World Traditional Program using C++ language.



The screenshot shows a code editor window titled "hello_world.cpp - /home/pi - Geany". The menu bar includes File, Edit, Search, View, Document, Project, Build, Tools, and Help. The toolbar contains various icons for file operations and editing. The editor displays the following C++ code:

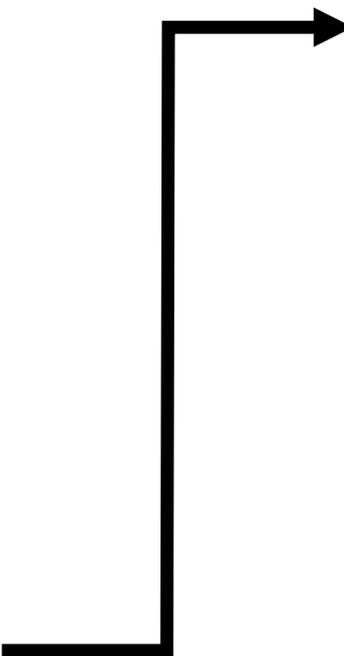
```
1  #include <iostream>
2
3  int main() {
4      std::cout << "Hello, World!" << std::endl;
5      return 0;
6  }
```

The left sidebar shows a "Symbols" view with a tree structure containing "Functions" and "main [3]".

Traditional Programming Languages...

Set up for Build Commands

Build	Tools	Help
Compile		F8
Build		F9
Lint		
Make		Shift+F9
Make Custom Target...		Shift+Ctrl+F9
Make Object		Shift+F8
Next Error		
Previous Error		
Execute		F5
Set Build Commands		



Set Build Commands

#	Label	Command	Working directory	Reset
C++ commands				
1.	Compile	g++ -Wall -c "%f"		🔧
2.	Build	g++ -Wall -o "%e" "%f"		🔧
3.	Lint	cppcheck --language;		🔧
	Error regular expression:			🔧
Independent commands				
1.	Make	make		🔧
2.	Make Custom Target...	make		🔧
3.	Make Object	make %e.o		🔧
4.				🔧
	Error regular expression:			🔧
<i>Note: Item 2 opens a dialog and appends the response to the command.</i>				
Execute commands				
1.	Execute	"/%e"		🔧
2.				🔧

%d, %e, %f, %p, %l are substituted in command and directory fields, see manual for details.

Cancel OK

Traditional Programming Languages...

To compile and execute the code

1st. Click to compile code:



Build	Tools	Help
Compile		F8
Build		F9
Lint		
Make		Shift+F9
Make Custom Target...		Shift+Ctrl+F9
Make Object		Shift+F8
Next Error		
Previous Error		
Execute		F5
Set Build Commands		

2nd. Click to Execute code:

Question 2

To execute C++ code using Geany, the first step is to

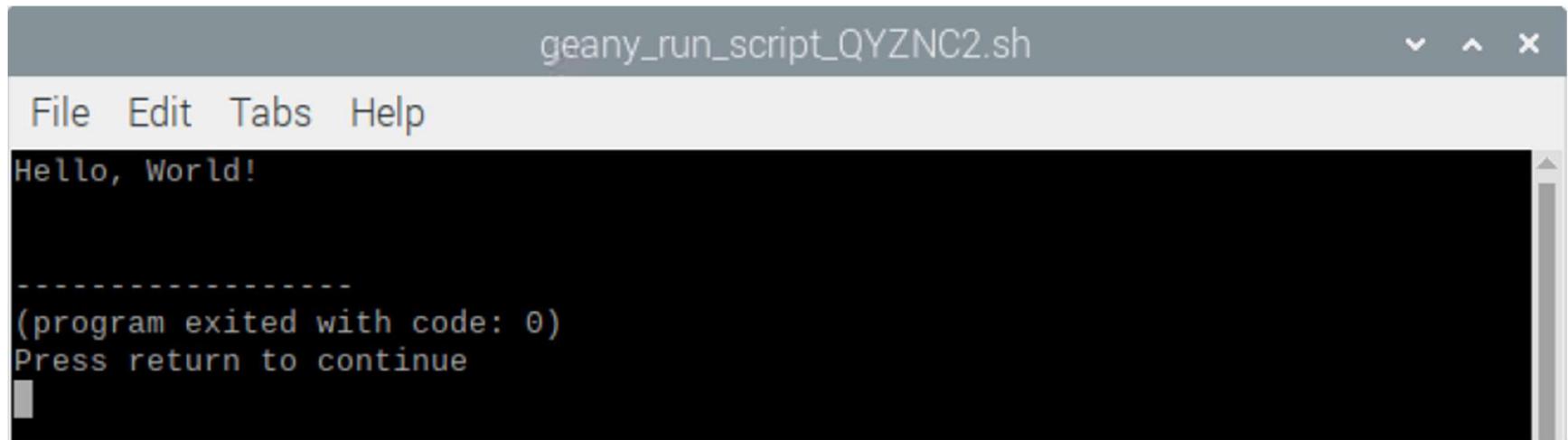
_____.

- a) Build
- b) Execute
- c) Compile
- d) None of the above



Traditional Programming Languages...

Executed Code- Output



```
geany_run_script_QYZNC2.sh
File Edit Tabs Help
Hello, World!
-----
(program exited with code: 0)
Press return to continue
```

Traditional Programming Languages...

Example of Google Colaboratory-
Python Executed Code-Output

```
▶ 1 print("Hello World!")  
2  
Hello world!
```

Traditional Programming Languages...

Example of a MicroPython
Executed Code-Output

```
Shell
File "main.py", line 8, in <module>
KeyboardInterrupt:
MicroPython v1.19.1-852-g9ea64a36a on 2023-02-03; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>> print("Hello World!")
Hello World!
>>> |
```

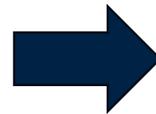
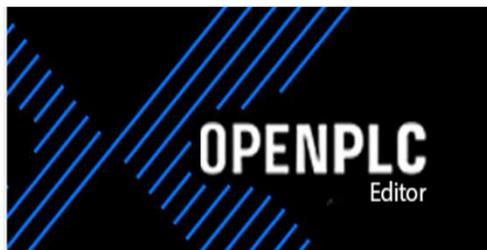
MicroPython (Raspberry Pi Pico) • /dev/ttyACM0

OpenPLC Overview

- OpenPLC is an open-source Programmable Logic Controller (PLC) that is based on easy-to-use software.
- It is the first fully functional standardized open-source PLC in software and hardware.
- The OpenPLC project was created in accordance with the IEC 61131-3 standard. The IEC (International Electrotechnical Commission) standard for programming PLCs
 - a) defines basic software architecture
 - b) defines programming languages

OpenPLC Overview

- OpenPLC is a versatile and powerful tool that can be used for various industrial automation applications. It is a good choice for both hobbyists and professional engineers.
- Besides allowing PLCs to be programmed easily, OpenPLC can be used to program an ESP32 microcontroller. (Innovation)



Question 3

OpenPLC is a legacy Programmable Logic Controller that is based on traditional software.

- a) True
- b) False



IEC61131-3 Functional Programming Specification

- IEC 61131-3 is the third part of the IEC 61131 standard for programmable logic controllers (PLCs).
- It defines the basic software architecture and programming languages for PLCs.
- The standard specifies two textual and three graphical programming languages:
 - a) Instruction List (IL)
 - b) Structured Text (ST)
 - c) Ladder Diagram (LD)
 - d) Function Block Diagram (FBD)
 - e) Sequential Function Charts (SFC)

IEC61131-3 Functional Programming Specification. . .

- Instruction List (IL) - A low-level, assembly-like language that is easy to learn but can be difficult to read and maintain.
- Structured Text (ST) - A high-level, Pascal-like language that is more readable and maintainable than IL.
- Ladder Diagram (LD) - A graphical language that is based on electrical ladder diagrams.
- Function Block Diagram (FBD) - A graphical language that is based on function blocks.
- Sequential Function Charts (SFC) - A graphical language that is based on state machines. SFC is used to represent complex control sequences.

IEC61131-3 Functional Programming Specification...

An Example of an IL: Use Case, A Traffic Light

```
LD      Input_1
ANDN    Input_2
ST      Red_Output

LD      Input_2
ANDN    Input_3
ST      Green_Output

LD      Input_3
ANDN    Input_1
ST      Yellow_Output
```

IEC61131-3 Functional Programming Specification...

An Example of SFC: Use Case, A Traffic Light

```
[Start] --> [Red]  
[Red] --> [Green]  
[Green] --> [Yellow]  
[Yellow] --> [Red]
```

IEC61131-3 Functional Programming Specification...

An Example of ST: Hello World

```
PROGRAM ESP32_Hello_World
VAR
    Hello_PB AT %IX0.1 : BOOL;
    Hello_LED AT %QX0.1 : BOOL;
END_VAR

    Hello_LED := Hello_PB;
END_PROGRAM

CONFIGURATION Config0

    RESOURCE Res0 ON PLC
        TASK task0 (INTERVAL := T#20ms, PRIORITY := 0);
        PROGRAM instance0 WITH task0 : ESP32_Hello_World;
    END_RESOURCE
END_CONFIGURATION
```

IEC61131-3 Functional Programming Specification...

An Example of ST: Hello World

With Tags and I/O Addresses

```
PROGRAM ESP32_Hello_World
VAR
    Hello_PB AT %IX0.1 : BOOL;
    Hello_LED AT %QX0.1 : BOOL;
END_VAR

Hello_LED := Hello_PB;
END_PROGRAM
```

```
PROGRAM ESP32_Hello_World
VAR
    Hello_PB : BOOL;
    Hello_LED : BOOL;
END_VAR

Hello_LED := Hello_PB;
END_PROGRAM
```

Basic BIT Instructions

What are BIT Instructions?

- BIT instructions in a PLC (Programmable Logic Controller) are used to manipulate individual bits or groups of bits within memory locations or registers.
- These instructions are essential for controlling and monitoring:
 - a) discrete input and output devices.
 - b) performing logical operations.
 - c) and implementing complex control sequences.

Basic BIT Instructions. . .

What are BIT Instructions?

- **Examine If Set (XIC):** This instruction examines the state of a specified bit. If the bit is set to 1, the instruction evaluates to true, otherwise, it evaluates to false.
- **Examine If Reset (XIO):** This instruction examines the state of a specified bit. If the bit is reset to 0, the instruction evaluates to true, otherwise, it evaluates to false.
- **Output Energize (OTE):** It controls the state of an output device, such as a relay, solenoid, or motor, by setting or resetting a corresponding bit in the PLC's memory.

Basic BIT Instructions. . .

What are BIT Instructions?

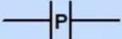
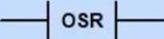
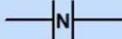
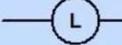
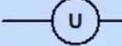
S.No	Description	Instruction Symbol
1	Contact	
2	Inverted Contact	
3	Positive transition (or) One shot Rising edge	 or 
4	Negative transition (or) One shot Falling edge	 or 
5	Coil	
6	Latched Coil (or) Set Coil	 or 
7	Unlatched Coil (or) Reset Coil	 or 

Image courtesy of

<https://seeithow.com/2019/07/08/ladder-logic-bit-instructions/>

Question 4

Which BIT instruction examines the state of a specified bit?

- a) Examine If Set**
- b) Examine If Reset**
- c) Output Energize**
- d) None of the above**

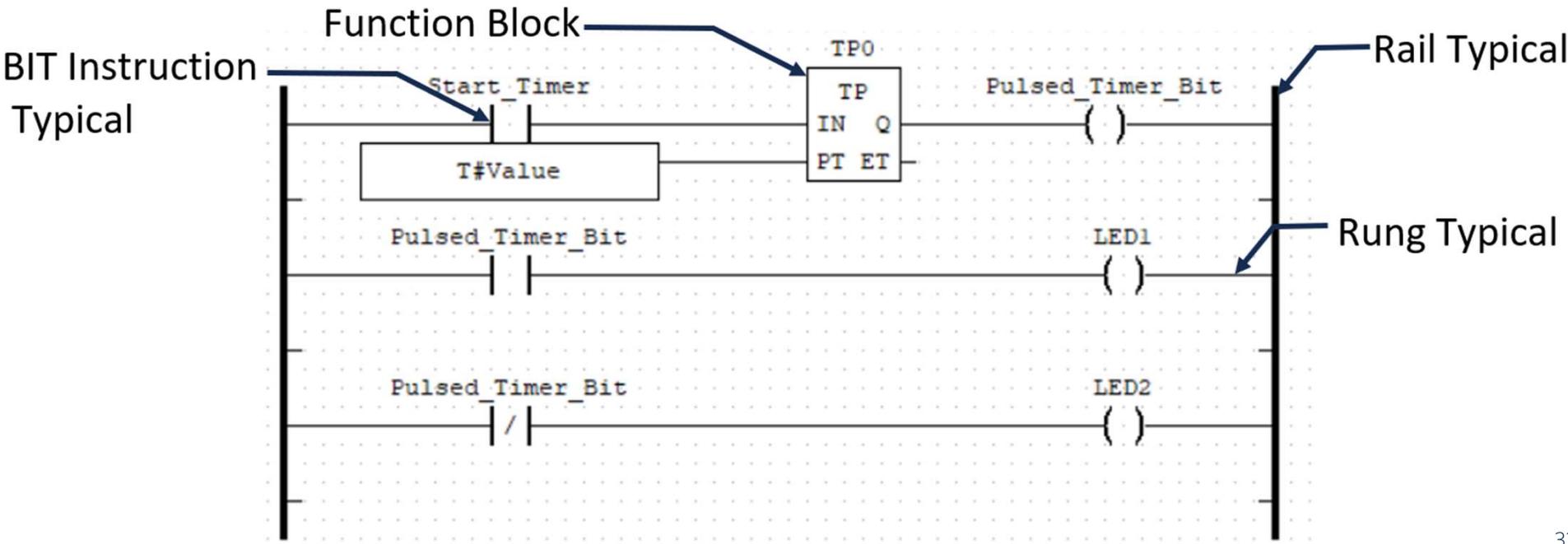


Ladder Logic Basics

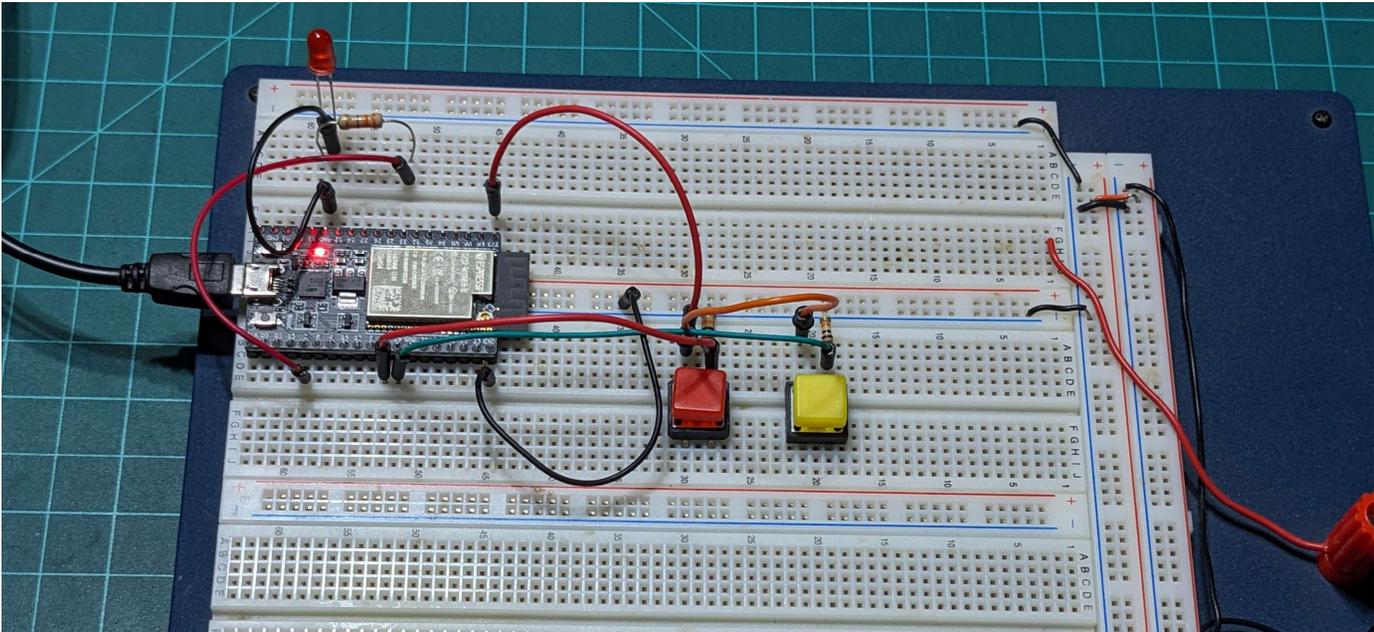
- A ladder diagram (LD) is a graphical representation of a program for a programmable logic controller (PLC).
- LD is based on the following elements.
 - a) Electrical ladder diagrams (ELD).
 - b) An ELD is used to design relay circuits.
- LDs are easy to understand and use.
- Popular language choice for programming PLCs.

Ladder Logic Basics

Anatomy of a Ladder Diagram



Lab: Hello World Demo



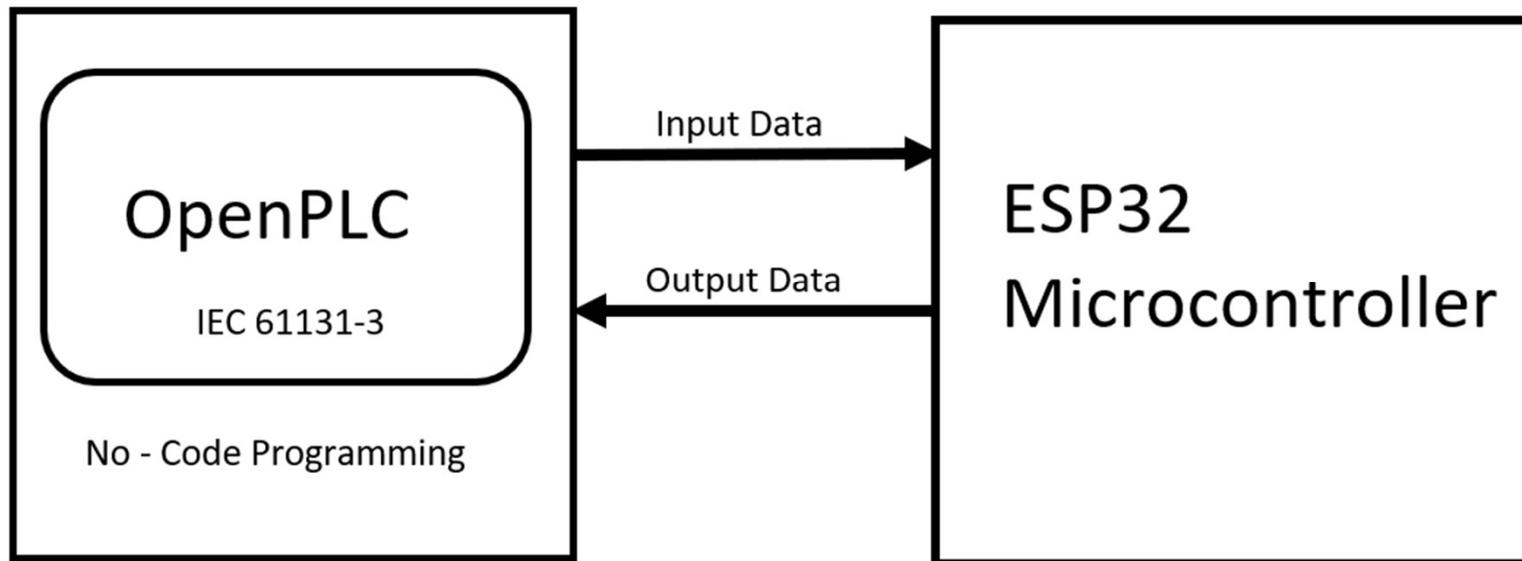
Lab: Hello World Demo...

Lab Objectives:

- Participants will learn to wire the ESP32 microcontroller Hello World demo circuit.
- Participants will learn to install OpenPLC Editor.
- Participants will learn to program the ESP32 microcontroller
- Participants will learn how to run the Hello World Demo

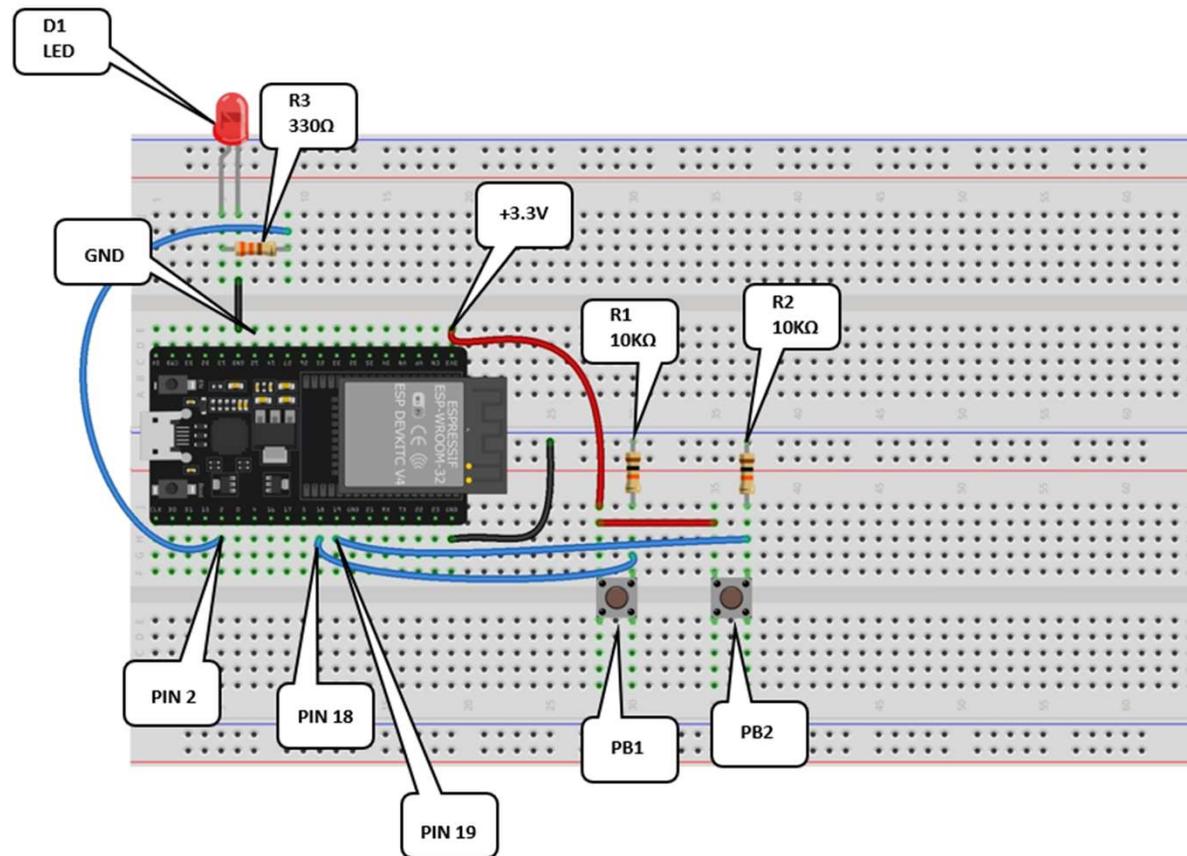
Lab: Hello World Demo...

Concept System Block Diagram

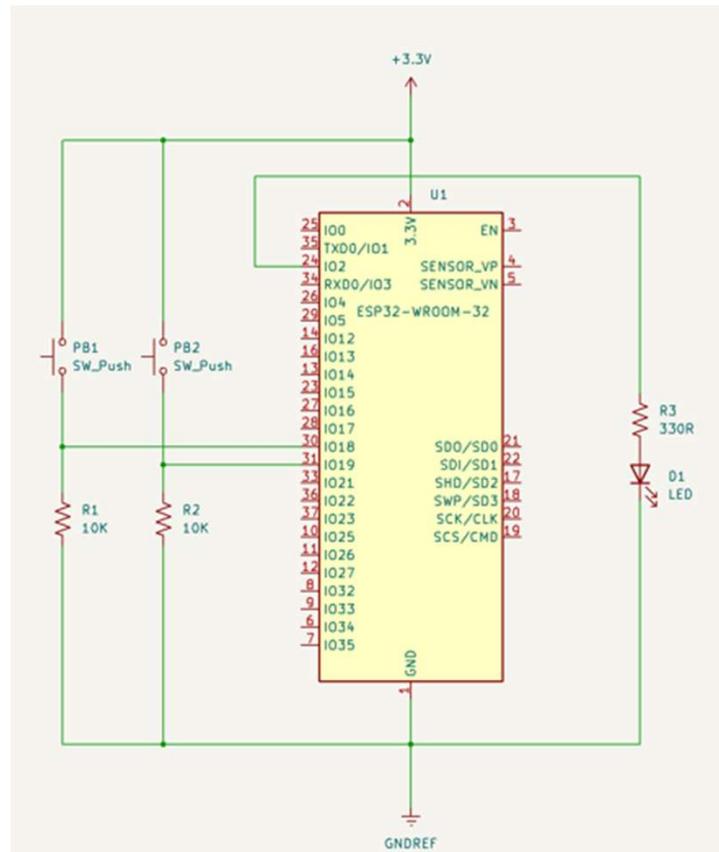


Lab: Hello World Demo...

Electrical Wiring
Diagram:
Solderless
Breadboard view

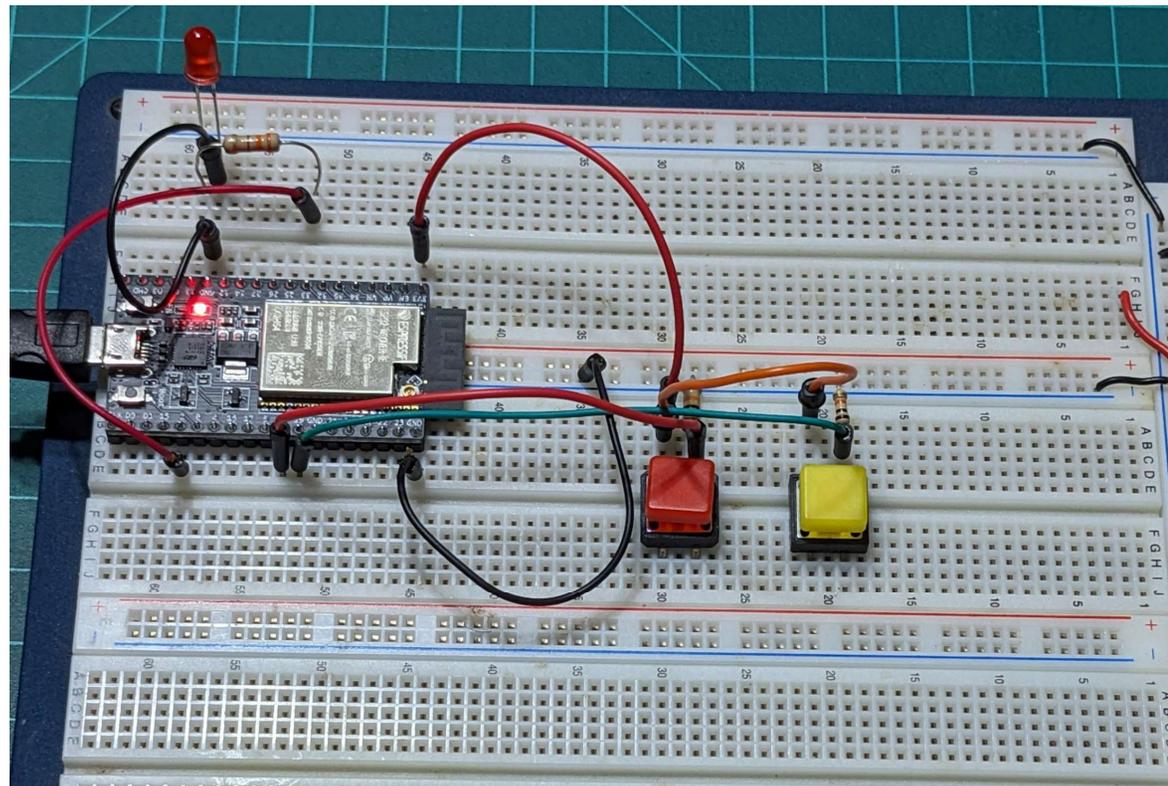


Lab: Hello World Demo...

Electronic Circuit
Schematic
Diagram

Lab: Hello World Demo...

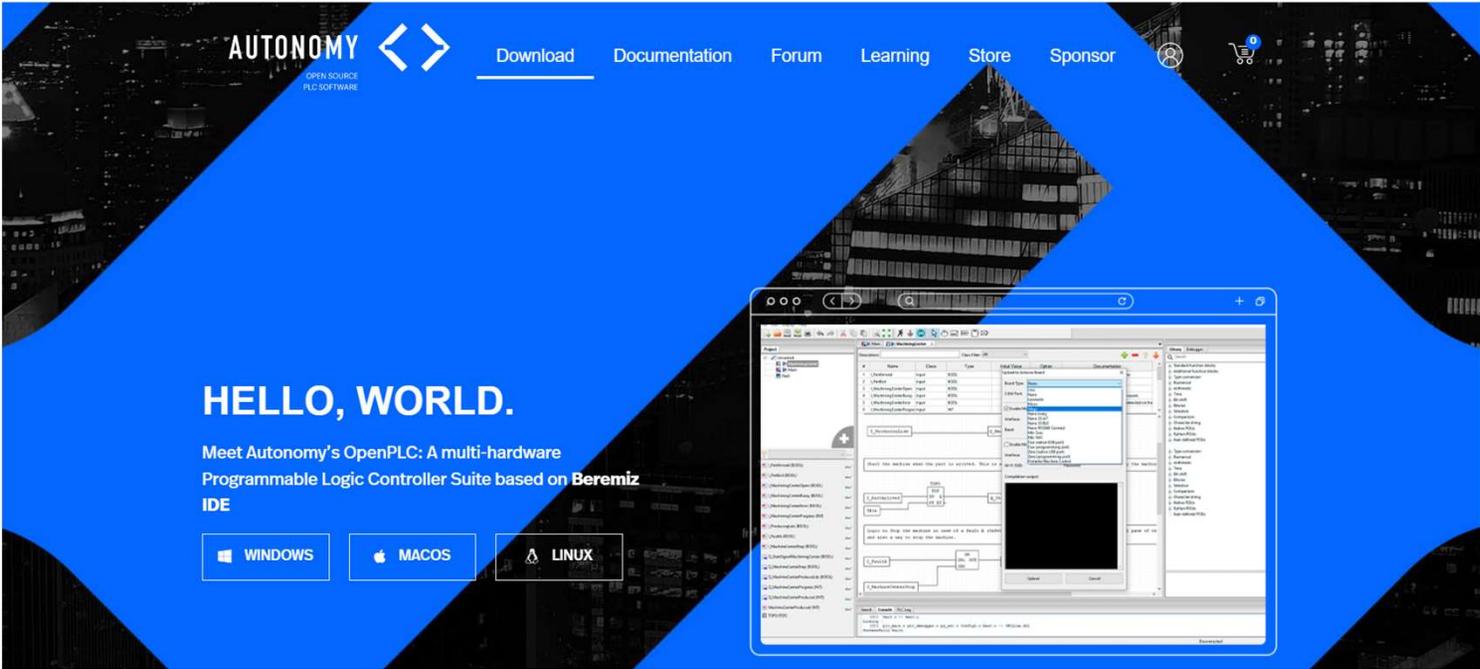
Solderless Breadboard Build



Lab: Hello World Demo...



Download OpenPLC from here

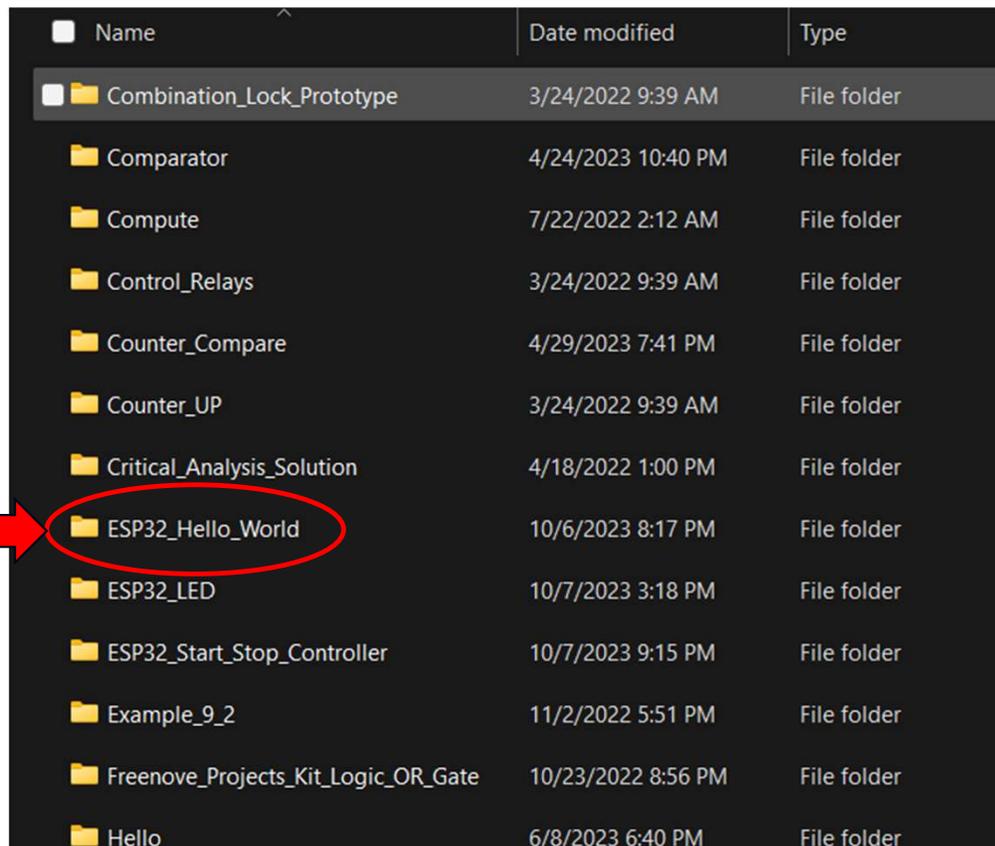


Source: <https://autonomylogic.com/>

Lab: Hello World Demo...

Create a Main Projects Folder

Name of Project Folder



Name	Date modified	Type
Combination_Lock_Prototype	3/24/2022 9:39 AM	File folder
Comparator	4/24/2023 10:40 PM	File folder
Compute	7/22/2022 2:12 AM	File folder
Control_Relays	3/24/2022 9:39 AM	File folder
Counter_Compare	4/29/2023 7:41 PM	File folder
Counter_UP	3/24/2022 9:39 AM	File folder
Critical_Analysis_Solution	4/18/2022 1:00 PM	File folder
ESP32_Hello_World	10/6/2023 8:17 PM	File folder
ESP32_LED	10/7/2023 3:18 PM	File folder
ESP32_Start_Stop_Controller	10/7/2023 9:15 PM	File folder
Example_9_2	11/2/2022 5:51 PM	File folder
Freenove_Projects_Kit_Logic_OR_Gate	10/23/2022 8:56 PM	File folder
Hello	6/8/2023 6:40 PM	File folder

Lab: Hello World Demo...

Click the OpenPLC Icon, the Editor will appear on the screen.

LD demo program
can be found in the
Github
Embedded_Controls_
Development_Code
folder here:

https://github.com/DWilcher/HCI_Electronics

The screenshot displays the OpenPLC Editor interface for an ESP32 microcontroller. The main workspace shows a Ladder Diagram (LD) with two rungs. The first rung contains a normally open contact labeled 'Hello_PB' connected to a coil labeled 'Hello_LED'. Below the diagram, the text 'Hello World LD demo program' is visible. The interface includes a Project Explorer on the left, a Variable Declaration Table at the top, and a Library on the right.

#	Name	Class	Type	Location	Initial Value	Option	Documentation
1	Hello_PB	Local	BOOL	%IX0.1			Pin 18 on ESP32 microcontroller
2	Hello_LED	Local	BOOL	%QX0.1			Pin 19 on ESP32 microcontroller

Lab: Hello World Demo...

Click the Arduino Icon, the Upload to Arduino Board window will appear.



Click on the Arduino Icon

Upload to Arduino Board

Board Type: ESP32

Mode: Compile Only

COM Port:

Enable Modbus RTU (Serial)

Interface: Serial Baud: 115200

Slave ID: 0 Tx Pin: -1

Enable Modbus TCP

Interface: Ethernet

MAC: 0xDE, 0xAD, 0xBE, 0xEF, 0xDE, 0xAD

IP: DNS:

Gateway: Subnet: 255.255.255.0

Wi-Fi SSID: Password:

Compilation output:

Upload Cancel

Establishing Communication with the ESP32 DevKit board

Lab: Hello World Demo...

Establishing Communication with the ESP32 DevKit board

Upload to Arduino Board

Board Type: ESP32

Mode: Compile Only

COM Port: [Dropdown]

Select COM port

Click here to upload Hello_World LD to ESP32 DevKit Board

Compilation output:

[Text Area]

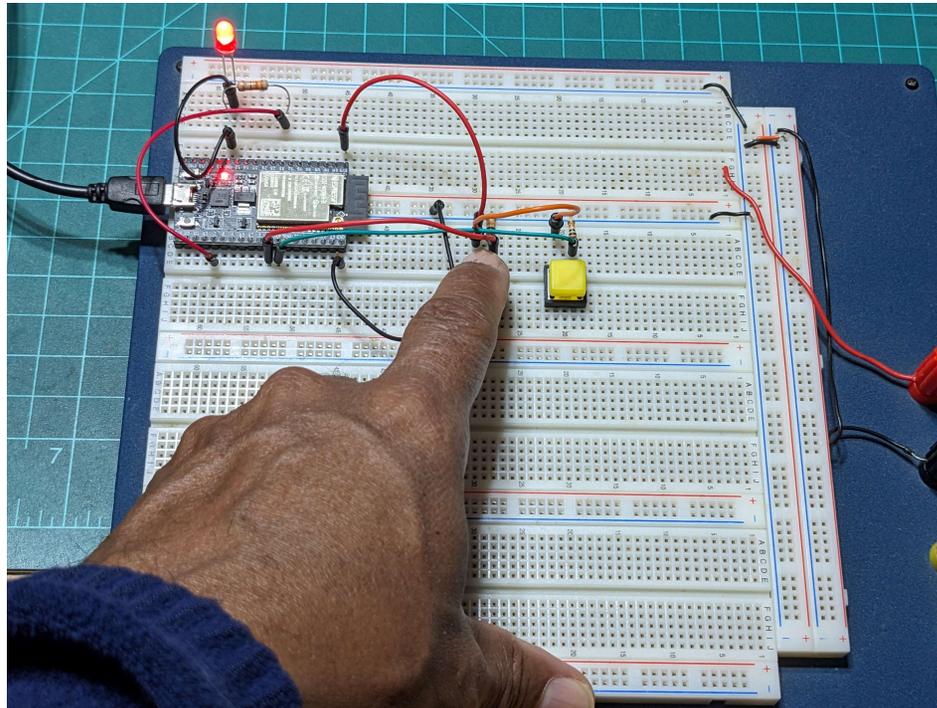
Upload Cancel

Lab: Hello World Demo...

Running Hello World Demo on Solderless Breadboard

Click on the link to watch
the Hello World Demo

<https://youtu.be/QgXomLAnhs8>



Question 5

When the Arduino Icon is clicked with a mouse, _____ will appear on the monitor.

- a) The Arduino Window**
- b) The Arduino IDE**
- c) The Arduino Board**
- d) None of the above**



Thank you for attending

Please consider the resources below:

Zemmouri, A., Barodt, A., Dahou, H., Alarequi, M., Eigouri, R., Htou, L., & Benbrahim, M. (2023). A microsystem design for controlling a dc motor by pulse width modulation using microblaze soft-core. *International Journal of Electrical and Computer Engineering*, 13(2), 1337-1448.

https://www.researchgate.net/publication/365994306_A_microsystem_design_for_controlling_a_DC_motor_by_pulse_width_modulation_using_MicroBlaze_soft-core

OpenPLC.(2023). *Openplc overview*. <https://autonomylogic.com/docs/openplc-overview/>



DesignNews

Thank You

Sponsored by

DigiKey

