eSIM

Cavli P32 Series Modules Evaluation Kit

P32 Series EVK Hardware Manual Release Version 2.0

www.cavliwireless.com

Copyright © 2019 by Cavli Wireless Inc.

All rights reserved. All document materials, including, without limitation, the logos, design, text, graphics, other files and the selection and arrangement are copyrighted as otherwise noted on the linked page. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law. For permission requests, are allowed only for company with registered customer ID within Cavli Wireless. Write to sales@cavliwireless.com for registering the company.

Cavli Wireless Inc.
177 Park Avenue
San Jose, California, USA 95113
www.cavliwireless.com

Designed in USA

Table of Contents

Table of Figures

Chapter 1. Introduction

This document is the wireless solution product Cavli P32 EVK hardware interface manual. It is intended to describe the hardware composition and functional characteristics of the module solution product, application interface definition and usage, electrical and mechanical characteristics. Combined with this document and other application documents, users can quickly use the module to design wireless products.

Chapter 2. EVK Overview

The P32 evaluation kit provides users to test and develop their own applications on P32 series modules. There are three modules which P32EVK supports.

- ✓ P32C1RS
- ✓ P32C31QM
- ✓ P32C1RM

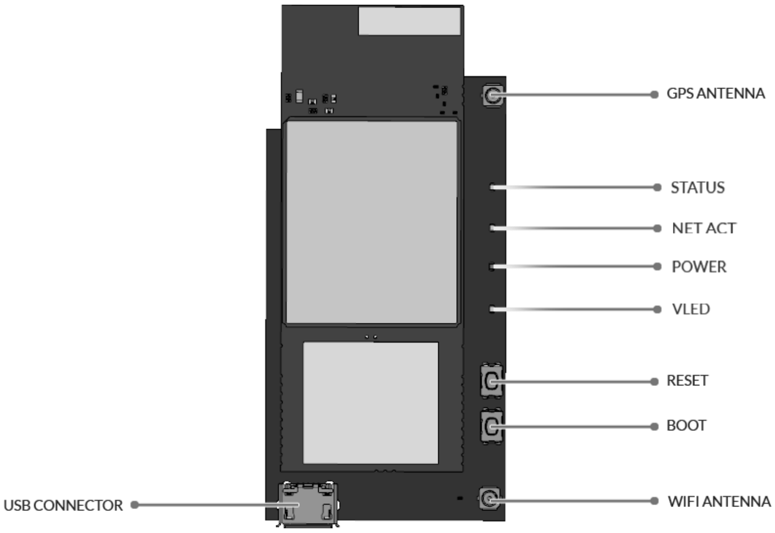


Figure 1 P32 EVK

2.1 Technical Specification

The P32 Series EVK technical specification is given below

Characteristic	Describe
Physical characteristics	55.93mm*30.24mm*5.53mm
On Board LEDS	Status, Netact, Power and Uled
Working voltage	5V
Buttons	Reset and Boot
General Purpose IO pins	12
Status pins	AP_READY and W_DISABLE
SMA Connector	One SMA connector with 50 ohm impedence
USB Connector	UDP/TCP/CoAP/LWM2M PPP/SSL/DTLS/FTP HTTP/MQTT/HTTPS
Antenna interface	50 Ω interface of the main antenna
	50 Ω interface of Bluetooth antenna
Firmware update	Serial port upgrade
Temperature range	Normal working temperature - 20℃ to +70℃ Limit working temperature - 40℃ to +85℃ Storage temperature: -45℃ to +90℃

Chapter 3. P32 EVK

3.1 EVK Pin Layout

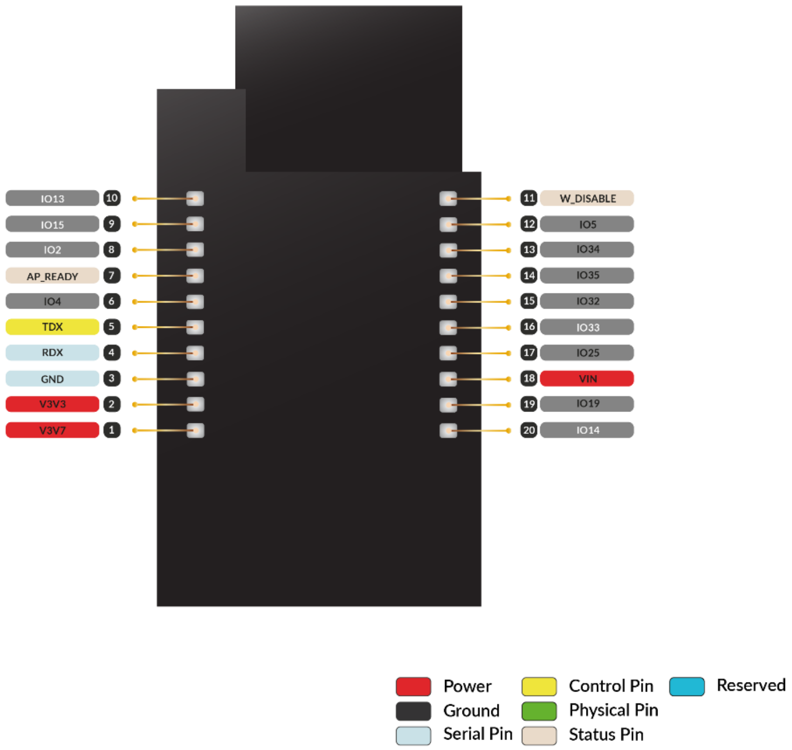


Figure 2 P32 pin layout

3.2 EVK Pin Description

Pins	Pins name	IO	Functional description	Remarks
1	V3V7	PO	Modem Voltage	Input Voltage
2	V3V3	PO	MCU Voltage	Input Voltage
3	GND		Ground Signal	GND
4	RXD	DI	Master Data Reception	3.3 Voltage domain
5	TXD	DO	Master Data Transmission	3.3 Voltage domain
6	IO4	IO	General Purpose IO	3.3 Voltage domain
7	AP_READY	DO	AP Sleep state detection	1.8 Voltage domain
8	IO2	IO	General Purpose IO	3.3 Voltage domain
9	IO15	IO	General Purpose IO	3.3 Voltage domain
10	IO13	IO	General Purpose IO	3.3 Voltage domain
11	W_DISABLE	DO	Airplane Mode Control	1.8 Voltage domain
12	IO5	IO	General Purpose IO	3.3 Voltage domain
13	IO34	IO	General Purpose IO	3.3 Voltage domain
14	IO35	IO	General Purpose IO	3.3 Voltage domain

15	IO32	IO	General Purpose IO	3.3 Voltage domain
16	IO33	IO	General Purpose IO	3.3 Voltage domain
17	IO25	IO	General Purpose IO	3.3 Voltage domain
18	VIN	PI	Main Power Supply	5V
19	IO19	IO	General Purpose IO	3.3 Voltage domain
20	IO14	IO	General Purpose IO	3.3 Voltage domain

Chapter 4. EVK Interface

3.1 Power Interface

The voltage input range of the power supply of P32EVK Board is 5V. There are two methods to power the board.

- ✓ USB Mode: User can power the board using USB connector
- ✓ PIN Mode: User can power the board using VIN and GND pin respectively.

P32EVK has two output voltage ie., V3V3 and V3V7 pins has regulated 3.3 and 3.7 voltages respectively.

Notes:

- ✓ User should not power the board via USB port and pins simultaneously, it will damage the device permanently.
- ✓ Make sure, the power supply is stable and can deliver enough current.

Pins	Pins name	IO	Functional description	Min	Typ	Max
1	V3V7	PO	Modem Voltage	3.6	3.7	4.2
2	V3V3	PO	MCU Voltage	2.7	3.3	3.7
3	GND		Ground Signal			

3.2 Reset

The P32EVK can be reset using pressing the RESET button on the board.

3.3 Boot

Boot button is used for programming purpose. User need to press the boot button and then reset button to enter in the boot mode of the microcontroller.

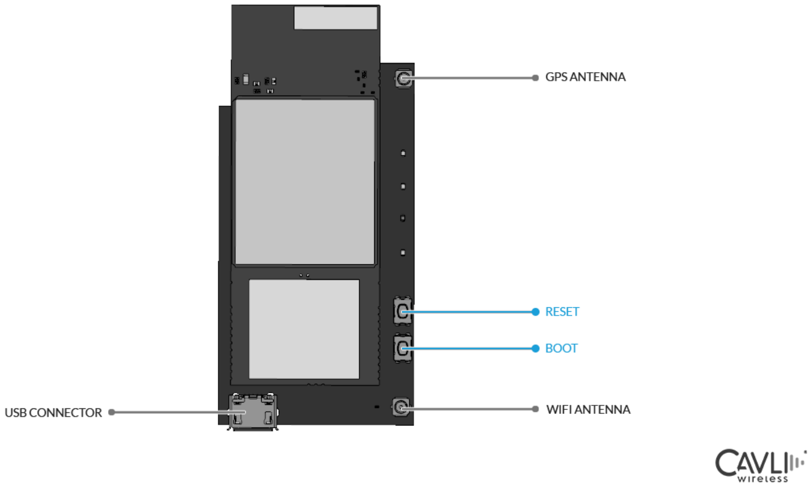


Figure 3 Reset and Boot

3.4 UART Interface

P32EVK serial port baud rate can be set to 115200.When users want to use the full function serial port, you can refer to the following connection mode

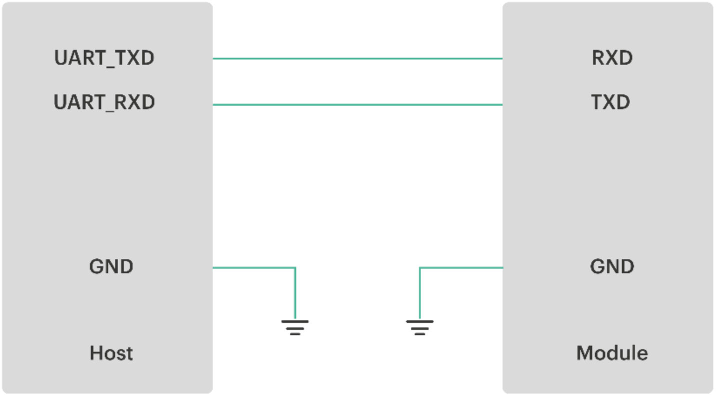


Figure 4 P32 EVK UART Connection

Pin	Pin Name	IO	Functional Description	Remarks
4	RXD	DI	Master Data Reception	3.3 Voltage domain
5	TXD	DO	Master Data Transmission	3.3 Voltage domain

Note:

- ✓ While programming, configure the upload speed to 921600. The detailed information is given in the guidelines.

3.5 GPIO Interface

P32EVK has 12 GPIO pins which can be assigned various functions by programming the appropriate registers.There are several kinds of GPIOs: digital-only, analog-enabled, capacitive-touch-enabled, etc. Analog-enabledGPIOs and Capacitive-touch-enabled GPIOs can be configured as digital GPIOs.

Pin	Pin Name	IO	Functional Description	Remarks
6	IO4	IO	General Purpose IO	3.3Voltage domain
8	IO2	IO	General Purpose IO	3.3Voltage domain
9	IO15	IO	General Purpose IO	3.3Voltage domain
10	IO13	IO	General Purpose IO	3.3Voltage domain
12	IO5	IO	General Purpose IO	3.3Voltage domain
13	IO34	IO	General Purpose IO	3.3Voltage domain
14	IO35	IO	General Purpose IO	3.3Voltage domain
15	IO32	IO	General Purpose IO	3.3Voltage domain

16	IO33	IO	General Purpose IO	3.3Voltage domain
17	IO25	IO	General Purpose IO	3.3Voltage domain
19	IO19	IO	General Purpose IO	3.3Voltage domain
20	IO14	IO	General Purpose IO	3.3Voltage domain

3.6 LED Interface

There are three LED indicators and general purpose led on the board.

Pin	Pin Name	IO	Functional Description	Remarks
1	STATUS		Network status Indication	
2	NETLIGHT		Network Indicator	
3	POWER		Power Indicator	
4	VLED		General Purpose LED	

Chapter 4. EVK Guidelines

4.1 Programming Guide (EVK)

- ✓ Power the EVK through either USB port or VIN(5V) and GND pin.
- ✓ Connect COM PORT via U0TXD and U0RXD pins or through on board USB port
- ✓ Press BOOT button and then press RESET button to enable ESP32 BOOT mode, when it entered in BOOT mode, a BOOT message will be displayed in the COM port.
- ✓ Flash the firmware to P32 Series module at upload speed of 256000
- ✓ After successful firmware flash, reset the module by pressing RESET button

  - ✓ Programing Consoles : Arduino, ESP IDF
  - ✓ Upload Speed – 926000/256000
  - ✓ CPU Frequency – 40MHz (Mandatory)
  - ✓ Flash size – 2MB (Mandatory)
  - ✓ The library packages of P32 Series can be found in the https://github.com/cavli-wireless
  - ✓ To disable the output log, following steps should be considered:
    - ▪ GPIO15 - Pull down to ground: Disable ROM Bootloader output
    - ▪ Make menuconfig – Bootloader config – Bootloader log verbosity – No output
    - ▪ Make menuconfig – Component config – Log Output – Default Log Verbosity – No output

Notes:

- ✓ User should not power the board via USB port and pins simultaneously, it will damage the device permanently.
- ✓ Make sure, the power supply is stable and can deliver enough current.

4.2 Setting up Arduino Platform (Windows)

STEP 1: Now, let’s get started. The first step would be to download and install the Arduino IDE. This can be done easily by following the link https://www.arduino.cc/en/Main/Software and downloading the IDE for free. If you already have one make sure it is of the latest version.

STEP 2: Go to File->Preferences.

Enter https://dl.espressif.com/dl/package_esp32_index.json into the “Additional Boards Manager URLs” field as shown in the figure below.

Then, click the “OK” button:
Note: if you already have the ESP32 boards URL, you can separate the URLs with a comma.

STEP 3: Open Arduino and go to Tools->Board->Board Manager. Select Type as “All” and search for esp32 by Espressif Systems and install the same. Close the window after installation.

STEP 4: Connect your ESP32 board to your computer through the micro-USB cable. Make sure the red LED goes high ON the module to ensure power supply.

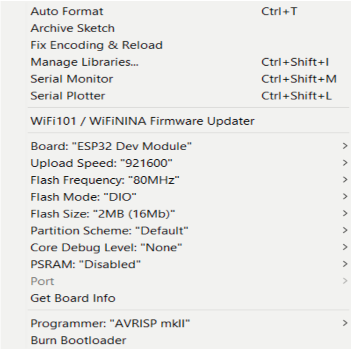
STEP5: Start the Arduino IDE and navigate to Tools -> Boards and select ESP32Dev board.

STEP6: Go back to Arduino IDE and under Tools -> Port select the Port to which your ESP is connected to.

STEP7: Let’s upload the Blink Program, to check if we are able to program our ESP32 module. This program should blink the LED at an interval of 1 second.

STEP 6: To upload the code, just click on upload and you should see the Arduino console displaying the following if everything works as expected.

Flash Configuration:



```
Auto Format                         Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries...                 Ctrl+Shift+I
Serial Monitor                      Ctrl+Shift+M
Serial Plotter                      Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "ESP32 Dev Module"                     >
Upload Speed: "921600"                        >
Flash Frequency: "80MHz"                      >
Flash Mode: "DIO"                             >
Flash Size: "2MB (16Mb)"                      >
Partition Scheme: "Default"                   >
Core Debug Level: "None"                      >
PSRAM: "Disabled"                             >
Port                                          >
Get Board Info
Programmer: "AVRISP mkII"                     >
Burn Bootloader
```

Figure 5 P32 EVK Flash configuration

4.3 Setting up Arduino Platform (Ubuntu)

STEP 1: Now, let’s get started. The first step would be to download and install the Arduino IDE. This can be done easily by following the link *https://www.arduino.cc/en/Main/Software* and downloading the IDE for free. If you already have one make sure it is of the latest version.

STEP 2: Unzip the folder and open the terminal. Run the following command.

```
./install.sh
```

STEP 3: Allow non root user to use tty0 (USB to Serial converter) serial communication with ESP32

```
sudo usermod -a -G dialout $USER
```

STEP 4: Go to **File->Preferences**.
Enter https://dl.espressif.com/dl/package_esp32_index.json into the "Additional Boards Manager URLs" field as shown in the figure below.

Then, click the "OK" button:
Note: if you already have the ESP32 boards URL, you can separate the URLs with a comma.

STEP 5: Open Arduino and go to **Tools->Board->Board Manager**. Select Type as "All" and search for **esp32** by **Espressif Systems** and install the same. Close the window after installation.

STEP 6: Connect your ESP32 board to your computer through the micro-USB cable. Make sure the red LED goes high ON the module to ensure power supply.

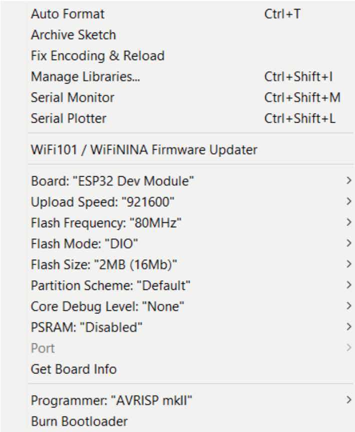
STEP 7: Start the Arduino IDE and navigate to Tools -> Boards and select ESP32Dev board.

STEP 8: Go back to Arduino IDE and under Tools -> Port select the Port to which your ESP is connected to.

STEP 9: Let's upload the Blink Program, to check if we are able to program our ESP32 module. This program should blink the LED at an interval of 1 second.

STEP 10: To upload the code, just click on upload and you should see the Arduino console displaying the following if everything works as expected.

Flash Configuration:



Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "ESP32 Dev Module"
Upload Speed: "921600"
Flash Frequency: "80MHz"
Flash Mode: "DIO"
Flash Size: "2MB (16Mb)"
Partition Scheme: "Default"
Core Debug Level: "None"
PSRAM: "Disabled"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader

Figure 6 P32 EVK Flash configuration - 2

4.4 Setting up ESP-IDF Platform (Ubuntu)

STEP 1: Installing Toolchain: Open Terminal and run the following command
sudo apt-get install git wget libncurses-dev flex bison gperf python python-pip python-setuptools python-serial python-click python-cryptography python-future python-pyparsing python-pyelftools cmake ninja-build ccache

Step2: With some Linux distributions you may get the Failed to open port /dev/ttyUSB0 error message when flashing the ESP32. This can be solved by running following two commands

sudo usermod -a -G dialout $USER

Step3: Get ESP-IDF Open Terminal, and run the following commands:

cd ~/esp
git clone --recursive https://github.com/espressif/esp-idf.git

Step4: Set up the tools: Aside from the ESP-IDF, you also need to install the tools used by ESP-IDF, such as the compiler, debugger, Python packages, etc.

cd ~/esp/esp-idf
./install.sh

Step5: Set up the environment variables: In the terminal where you are going to use ESP-IDF, run:
. $HOME/esp/esp-idf/export.sh

Step6: Start a Project: Now you are ready to prepare your application for ESP32. You can start with get-started/hello_world project from examples directory in IDF.

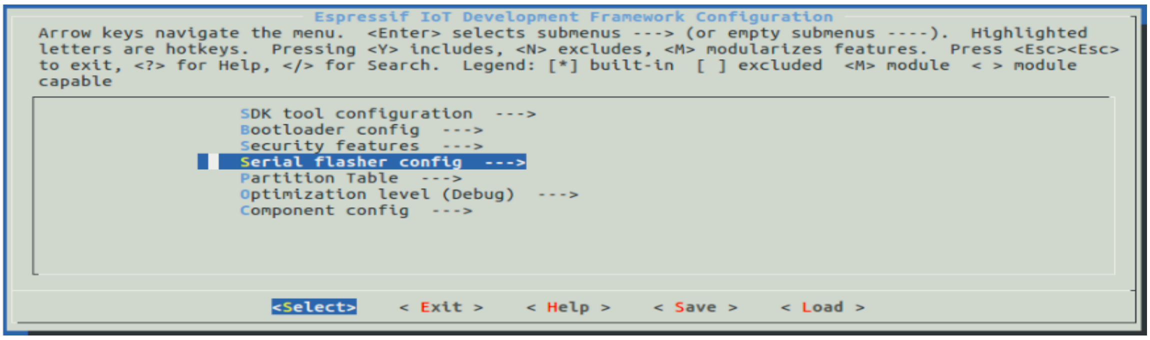
cd ~/esp
cp -r $IDF_PATH/examples/get-started/hello_world .

Step7: Connect your device: Now connect your ESP32 board to the computer and check under what serial port the board is visible. Serial ports have the patterns in their name /dev/tty

Step8: Configure: Navigate to your hello_world directory from Step 5. Start a Project and run the project configuration utility menuconfig.

cd ~/esp/hello_world
make menuconfig

If the previous steps have been done correctly, the following menu appears:



```
                 Espressif IoT Development Framework Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted
  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc>
  to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module
  capable

                   SDK tool configuration  --->
                   Bootloader config  --->
                   Security features  --->
                   Serial flasher config  --->
                   Partition Table  --->
                   Optimization level (Debug)  --->
                   Component config  --->

             <Select>    < Exit >    < Help >    < Save >    < Load >
```

Figure 7 P32 EVK ESP IDF Framework configuration

Step9: Compile the application by running following command

Make

Step10: Flash the program into the chip

Make flash

4.5 Setting up ESP-IDF Platform (Windows)

STEP 1: The easiest way to install ESP-IDF's prerequisites is to download the ESP-IDF Tools installer from this URL:

https://dl.espressif.com/dl/esp-idf-tools-setup-2.0.exe

The installer includes the cross-compilers, OpenOCD, cmake and Ninja build tool, and a configuration tool called mconf-idf. The installer can also download and run installers for Python3.7 and Git For Windows if they are not already installed on the computer.

STEP 2: Navigate to the folder C:\msys32\ and open the application mingw32.exe

STEP 3: Open mingw32.exe and run the following commands

```
git clone --recursive https://github.com/espressif/esp-idf.git
```

STEP 4: Open vim to add IDF_PATH environment variable to your MYSYS system

```
cd esp-idf
vim /etc/ profile.d/export_idf_path.sh
```

Type “i” to enter write/read mode and type the following command

```
export IDF_PATH=”C:/msys32/home/esp-idf”
```

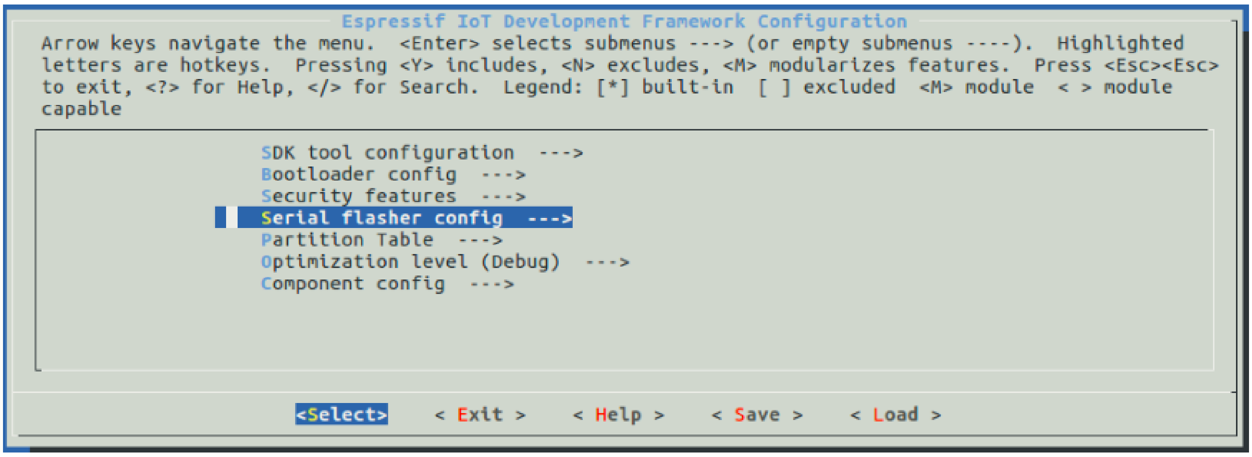
Please enter your esp-idf folder path after C:/msys32 and press escape and type “:wq” to save and exit vim.

Close and reopen MSYS terminal and type “printenv IDF_PATH” to ensure the ESP-IDF is known by the system

Step5: Configure: Navigate to your hello_world directory from Step 5. Start a Project and run the project configuration utility menuconfig.

```
cd ~/esp/hello_world
make menuconfig
```

If the previous steps have been done correctly, the following menu appears:



```
                Espressif IoT Development Framework Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted
 letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc>
 to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module
 capable

                    SDK tool configuration  --->
                    Bootloader config  --->
                    Security features  --->
                    Serial flasher config  --->
                    Partition Table  --->
                    Optimization level (Debug)  --->
                    Component config  --->

         <Select>    < Exit >    < Help >    < Save >    < Load >
```

Figure 8 P32 EVK ESP IDF Framework configuration – 2

Step6: Compile the application by running following command

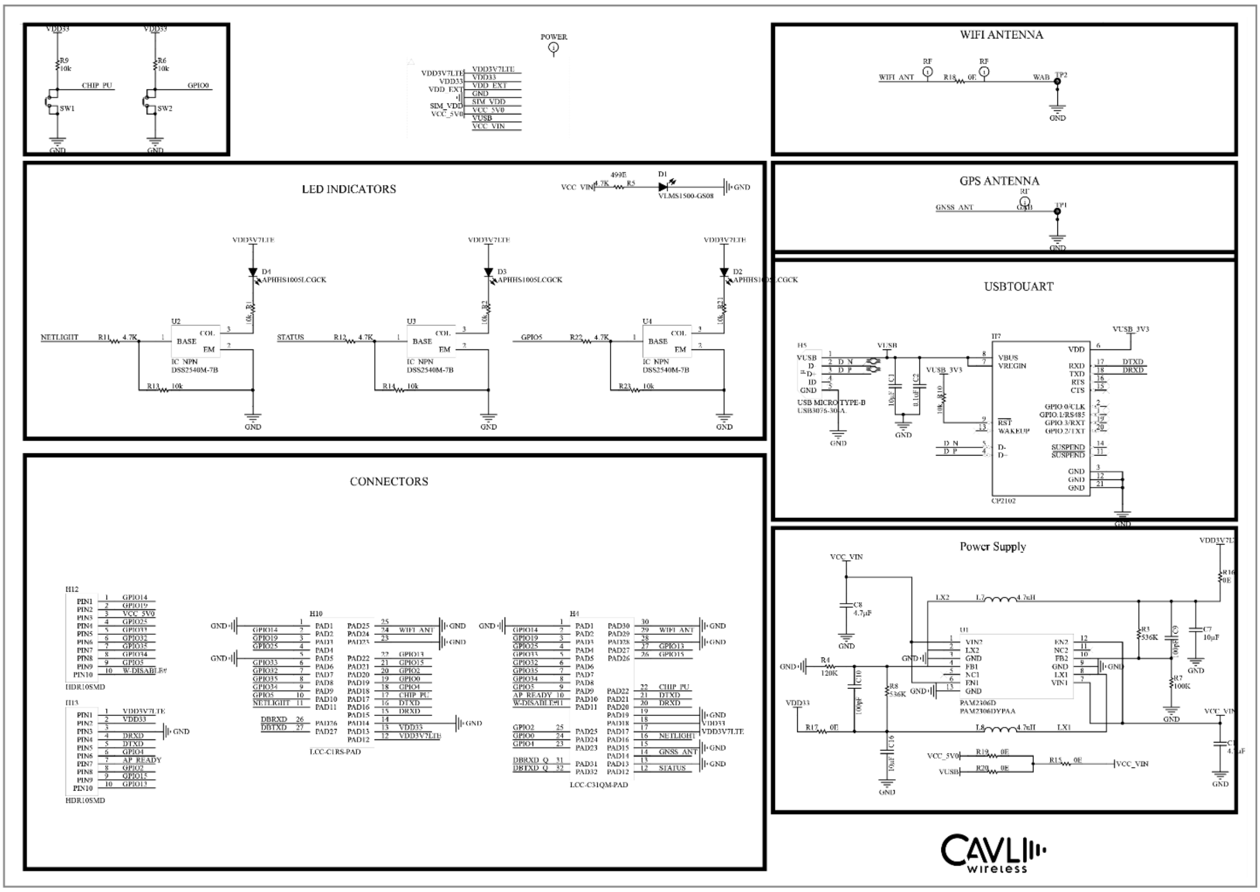
Make

4.6 Sample code

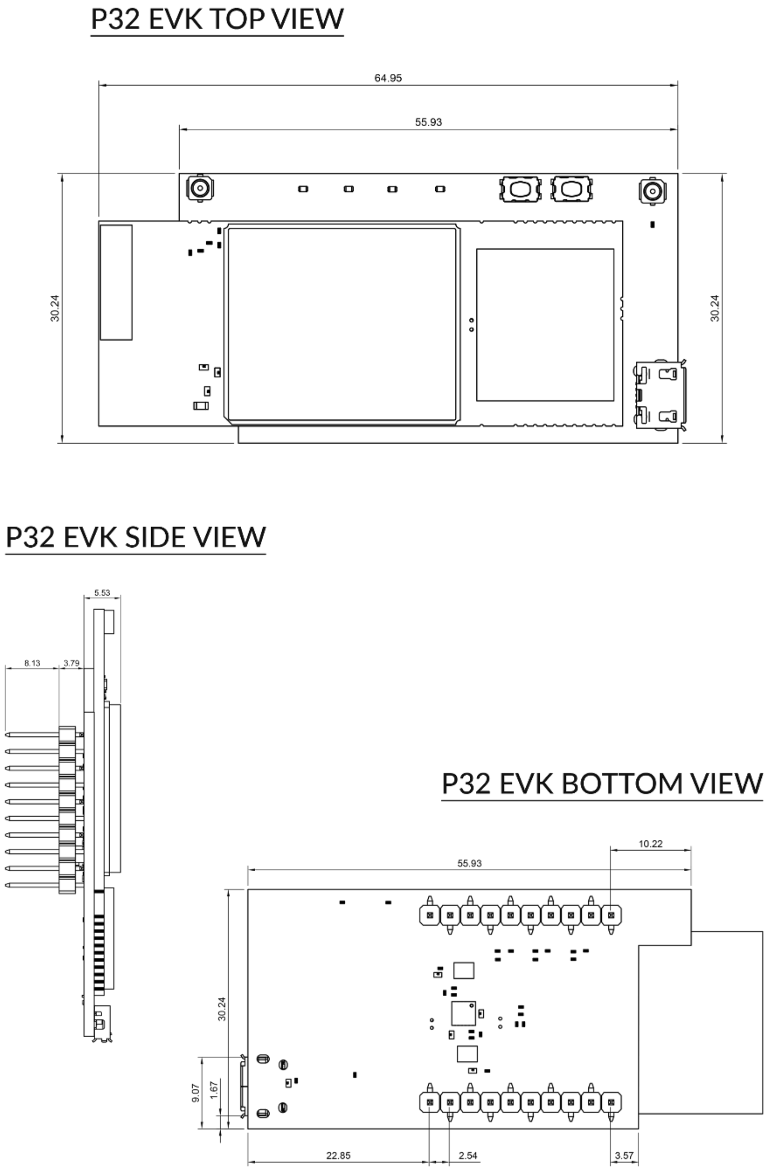
The sample code for P32 Series modules can be found at https://github.com/cavli-wireless

4.6 Reference Circuit

The reference circuit of P32 series EVK is given below,



4.7 Dimensions



P32 EVK TOP VIEW

P32 EVK SIDE VIEW

P32 EVK BOTTOM VIEW

Figure 9 P32 EVK Dimensions